

# Évaluation quantitative des descripteurs qualificatifs proposés

Dans le chapitre précédent, nous avons présenté un ensemble de caractéristiques de textures permettant d’avoir une description qualitative et robuste des textures volumétriques. Dans ce chapitre, nous proposons de tester les caractéristiques de notre méthode HUF (*Human Understandable Feature*) au travers de plusieurs expérimentations visant à produire des résultats quantitatifs. Il sera ainsi possible de comparer objectivement différentes méthodes dans des conditions identiques de travail. Pour cela, nous avons choisi des méthodes d’analyse de textures reconnues dans la littérature : les matrices de cooccurrence d’Haralick qui sont une référence en analyse d’images, ainsi que la méthode des motifs binaires (LBP) reconnue comme très performante et efficace en 2D et que nous avons étendue à la 3D. Cette extension constitue une autre contribution de notre travail décrite dans [Paulhac *et al.*, 2008].

Nous effectuons dans un premier temps des tests de classification, puis dans un deuxième temps, des tests de segmentation sur un ensemble de textures solides de synthèse. Dans les deux cas, les résultats présentés sont des évaluations quantitatives, obtenus à partir de critères d’évaluations reconnus. Afin de réaliser ces expérimentations, et faute de base existante, il nous a d’abord été nécessaire de produire notre propre base de textures solides de synthèse (textures volumétriques). Cette contribution est décrite dans la première partie de ce chapitre.

## 3.1 Base de textures solides pour l’évaluation de méthodes de classification et de segmentation

Dans la littérature, il existe plusieurs types de bases de données concernant le domaine des textures. Parmi les plus connues, nous trouvons les bases de données de textures 2D. La base de données de Brodatz [Brodatz, 1966] mais également la base de Meastex (<http://www.texturesynthesis.com/meastex/meastex.html>) sont souvent utilisées pour leur richesse. Les textures 3D et les textures volumiques, à ne pas confondre avec les textures faisant l’objet de notre étude, sont très différentes des textures so-

### 3.1. BASE DE TEXTURES SOLIDES POUR L'ÉVALUATION DE MÉTHODES DE CLASSIFICATION ET DE SEGMENTATION

---

lides : une texture 3D [Cuba et Dana, 2004] désigne la structure externe observée en 2D, d'un objet 3D vu d'un angle particulier avec différentes conditions d'éclairage ; une texture volumique [Neyret, 1995] correspond au rendu de reflets et de géométries répétitifs à l'intérieur d'une image 2D. Parmi les bases de textures 3D, PMTex (<http://www.taurusstudio.net/research/pmtexdb/compare.htm>), CUREt (<http://www1.cs.columbia.edu/CAVE/software/curet/index.php>) et OUTex ([http://www.outex.oulu.fi/index.php?page=outex\\_home](http://www.outex.oulu.fi/index.php?page=outex_home)) constituent des références. Concernant les textures volumiques, il est possible d'en trouver un grand nombre sur le site personnel de Neyret (<http://www-evasion.imag.fr/Membres/Fabrice.Neyret/>). Comme nous l'indiquons dans la partie état de l'art, les textures solides (textures volumétriques) sont des textures représentées dans le domaine 3D et sont assimilables à une série d'images texturées 2D ou comme des textures situées dans un espace volumétrique. Dans [Kopf *et al.*, 2007], les auteurs proposent de construire des textures solides à partir d'images de textures 2D. Plusieurs exemples sont disponibles à cette adresse <http://johanneskopf.de/publications/solid/textures/index.html> mais actuellement trop peu d'images sont disponibles pour des expérimentations de classification ou de segmentation.

Dans la partie état de l'art, nous décrivons trois différents types de textures : les textures déterministes, caractérisées par la répétition de motifs identiques ; les textures stochastiques, identifiables par leur irrégularité ; et les textures observables qui sont un mélange des deux précédentes [Gool *et al.*, 1985].

Afin d'avoir une base de données suffisamment représentative et complète, nous avons choisi de construire des images synthétiques représentatives de ces 3 classes de textures. Pour limiter l'espace disque ainsi que les temps de calcul lors des expérimentations, nous avons construit des textures volumétriques de taille  $64^3$ . Au regard de l'ensemble des tests de classification produits dans la littérature, cette taille semble suffisante et adaptée pour un problème de classification de textures. L'ensemble des images de notre base est construit à partir de quatre méthodes simples que nous détaillons dans les prochaines sous-sections. Pour finir, il est important de préciser que le but de cette base d'images n'est pas de fournir des textures solides qui soient le plus réaliste possible mais plutôt de mettre à disposition un ensemble de classes de textures volumétriques important, afin de permettre l'évaluation d'algorithmes d'analyse de textures solides à travers des tests de classification ou de segmentation.

#### 3.1.1 Méthodes de synthèse utilisées

La première méthode utilisée permet de construire une texture volumétrique en utilisant des images texturées 2D identiques à celles disponibles dans l'album de Brodatz. Deux textures 2D ou plus sont interpolées afin d'obtenir une image 3D (Figure 3.1). Les textures 2D utilisées dans la construction d'une image 3D font partie d'une même classe de textures mais ne sont pas exactement similaires. A partir de cette méthode d'interpolation, les textures volumétriques générées possèdent une direction particulière et évoluent de façon régulière. L'intérieur d'un tronc d'arbre constitue un bon exemple pour illustrer ce type de textures. Formellement, une texture interpolée peut être définie comme un ensemble de

### 3.1. BASE DE TEXTURES SOLIDES POUR L'ÉVALUATION DE MÉTHODES DE CLASSIFICATION ET DE SEGMENTATION

---

blocs texturés  $B_T$  construits à partir de textures 2D  $E_T$  :

$$B_{T,i} = E_T + i \frac{(E_T - E_{T+1})}{M} \quad (3.1)$$

avec  $M = N/(R - 1)$  le nombre de composants 2D dans un bloc,  $N$  la profondeur de la texture solide,  $R$  le nombre d'images texturées 2D utilisées pour l'interpolation,  $T = \{0, \dots, R - 1\}$  et  $i = \{0, 1, \dots, M - 1\}$ .

Les propriétés de ce genre de textures sont dépendantes des textures 2D choisies mais également du nombre de plans d'interpolation.

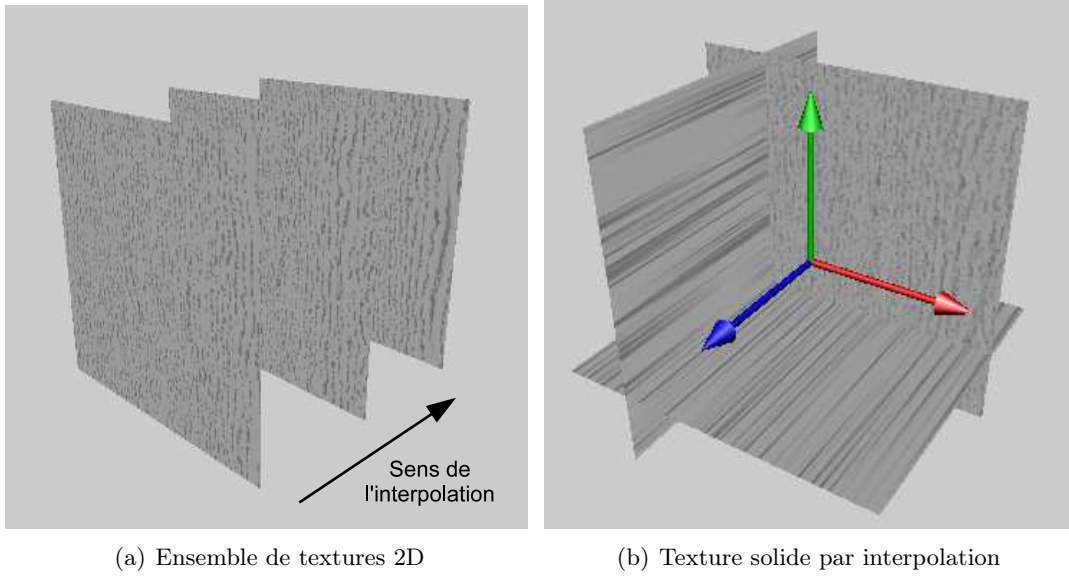


FIG. 3.1 – Exemple de construction d'une texture solide à partir de la méthode d'interpolation (méthode 1)

La deuxième méthode propose d'utiliser des formes géométriques de base (sphères, cubes, ellipses) incorporées au sein d'une image 3D. Cela permet notamment de construire des textures de type déterministe. Un motif est représenté par une forme géométrique positionnée dans une direction donnée. De plus, il est possible de construire des textures de type observable, par exemple en utilisant des sphères dont la taille et le positionnement sont aléatoires. Il est alors facile de construire une texture de type gruyère en insérant aléatoirement des sphères ou des ellipses dont la taille varie. Les propriétés de ce type de textures sont dépendantes du nombre de motifs utilisés, de la forme des motifs, de la taille et de la couleur des motifs, qui peuvent être fixes ou variables. La Figure 3.2 présente deux textures volumétriques générées en utilisant cette méthode.

La troisième méthode mise en place génère des textures volumétriques à partir de la transformée de Fourier [Lewis, 1984]. Pour cela, il est nécessaire de construire un spectre de puissance, qui permet de spécifier les fréquences et les amplitudes présentes dans la texture à synthétiser. Ce spectre peut être dessiné en respectant la symétrie des quadrants mais il est également possible d'utiliser le spectre de puissance d'une texture existante.

### 3.1. BASE DE TEXTURES SOLIDES POUR L'ÉVALUATION DE MÉTHODES DE CLASSIFICATION ET DE SEGMENTATION

---

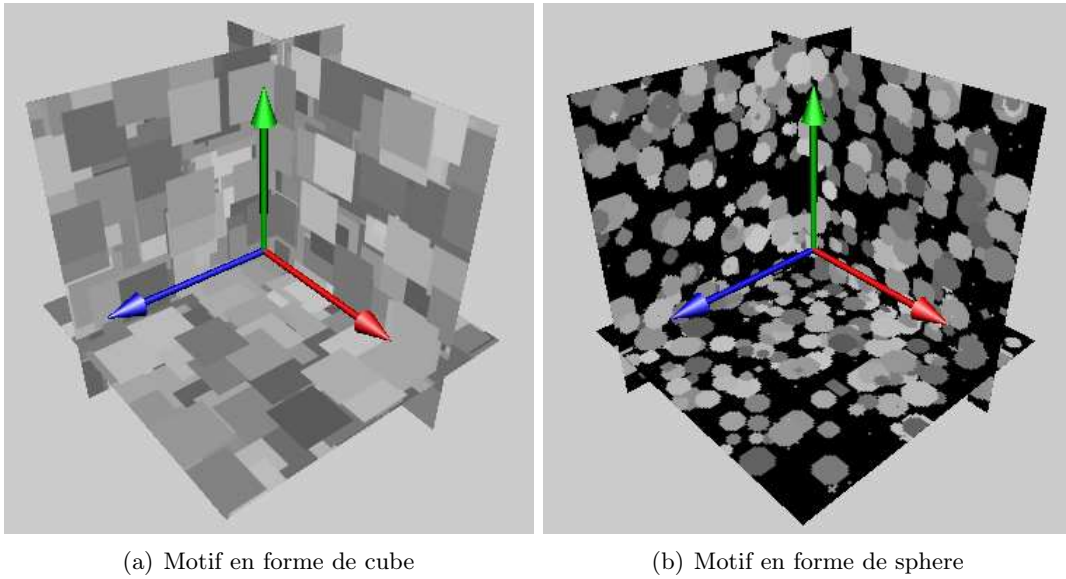


FIG. 3.2 – Textures solides construites à partir de la 2ème méthode

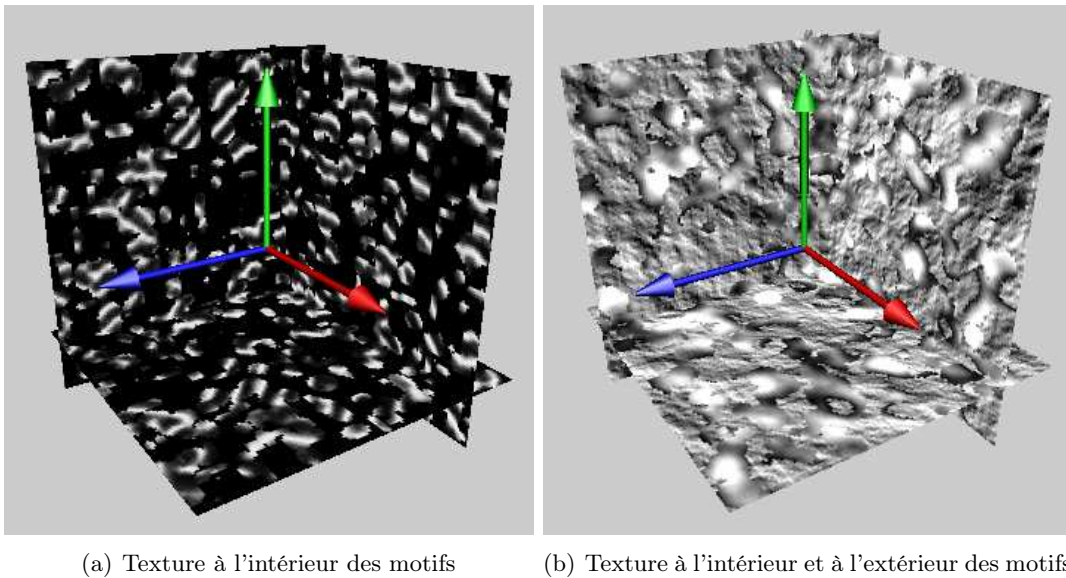


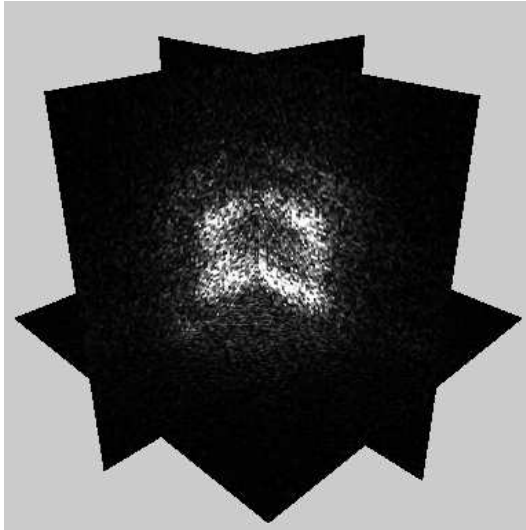
FIG. 3.3 – Exemple de textures solides avec des textures à l'intérieur et à l'extérieur des motifs géométriques (méthode 4)

Concernant le spectre de phase, des valeurs aléatoires peuvent être utilisées ou récupérées d'une texture existante. De cette façon, il est possible de synthétiser une texture solide à partir de la transformée inverse de Fourier. Dans [Lewis, 1984], les auteurs décrivent cette méthode de synthèse de textures comme peu intuitive. Cependant, elle permet de construire des textures qui sont riches et difficiles à obtenir dans le domaine spatial. Enfin, les textures appartenant à une même classe de textures sont générées à partir d'un unique spectre de

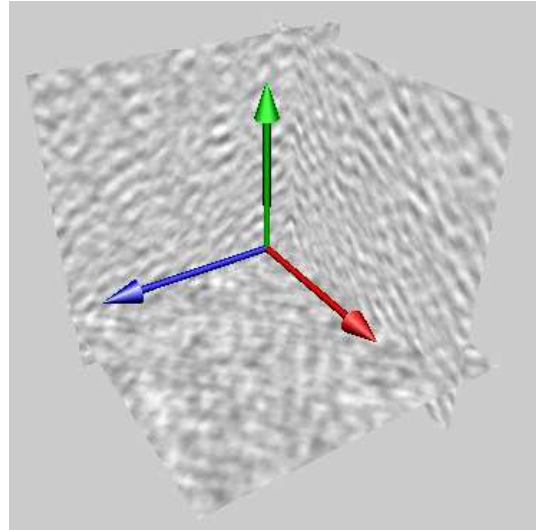
### 3.1. BASE DE TEXTURES SOLIDES POUR L'ÉVALUATION DE MÉTHODES DE CLASSIFICATION ET DE SEGMENTATION

---

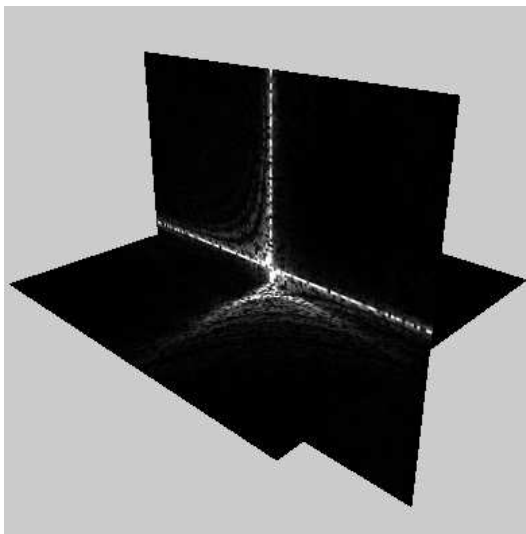
puissance et d'un spectre de phase variable. De cette façon, les textures à l'intérieur d'une classe ne sont pas exactement identiques.



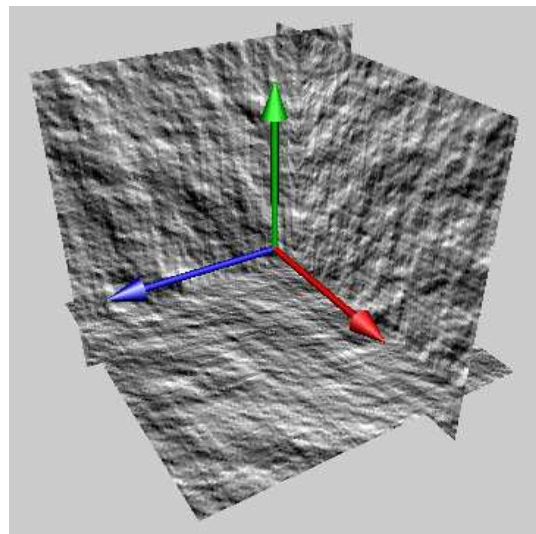
(a) Spectre de puissance 1



(b) Texture construite avec le premier spectre de puissance



(c) Spectre de puissance 2



(d) Texture construite à partir du deuxième spectre de puissance

FIG. 3.4 – Exemple de textures solides construites à partir de la transformée de Fourier (méthode 3)

Enfin, il est possible d'obtenir une quatrième catégorie de textures volumétriques en utilisant les propriétés des trois méthodes précédentes. Au lieu de générer un motif avec une couleur (méthode 2), nous pouvons créer des motifs contenant une texture donnée en utilisant la première méthode de synthèse ou la transformée de Fourier inverse (Figure 3.3).

### 3.1. BASE DE TEXTURES SOLIDES POUR L'ÉVALUATION DE MÉTHODES DE CLASSIFICATION ET DE SEGMENTATION

---

Les textures ainsi générées contiennent des formes géométriques composées de textures particulières. De plus, il est possible d'insérer une texture à l'extérieur des motifs afin d'obtenir des objets qui soient texturés. Les propriétés de ce genre de textures dépendent du nombre de motifs, de leur forme et leur taille, et des textures utilisées à l'intérieur et à l'extérieur des motifs.

Afin d'augmenter les variabilités à l'intérieur de notre base, nous appliquons un certain nombre de transformations de base pour enrichir l'ensemble des classes de textures proposées. Pour chaque classe, nous disposons ainsi de textures n'ayant subi aucune transformation, de textures bruitées, de textures floutées, de textures sous-échantillonnées, et de textures avec des rotations aléatoires selon les axes  $x$ ,  $y$  et  $z$  (Figure 3.5). De cette façon, il devient possible de tester si une méthode d'analyse de textures est invariante à la rotation, de tester sa robustesse au bruit ou au flou, et de pouvoir augmenter facilement la difficulté de comparaison des images 3D de la base de données lors d'un problème d'analyse de textures. Afin d'appliquer un sous-échantillonnage, nous construisons des images 3D texturées de taille  $128^3$  qui sont ensuite sous-échantillonnées afin d'obtenir une texture solide de taille  $64^3$  contenant des motifs deux fois plus petits. Concernant les distorsions de bruit et de flou, la distribution Gaussienne est centrée en 0 et l'écart type  $\sigma$  prend la valeur 5 dans le cas du bruit Gaussien et la valeur de 1.5 dans le cas du flou Gaussien. Ces transformations sont appliquées par convolution d'une image 3D avec un masque de taille  $3^3$ .

#### 3.1.2 Organisation de la base d'images

La base de données proposée est organisée de la façon suivante : les images texturées 3D sont séparées en fonction de leur méthode de synthèse et sont partitionnées en quatre dossiers. Dans chacun d'eux, les images sont regroupées selon le type de transformations appliquées : aucune, flou Gaussien, bruit Gaussien, rotation aléatoire, sous-échantillonnage. Les sous-dossiers "transformations" contiennent l'ensemble des classes de textures générées par une méthode de synthèse.

Actuellement, notre base de données contient 95 classes différentes : 30 classes sont construites avec la méthode d'interpolation, 25 avec la méthode géométrique, 15 avec la transformée de Fourier inverse et 25 avec la méthode des motifs texturés. Chacune des classes se compose de 50 exemples de textures solides : 10 textures floues, 10 textures bruitées, 10 textures sous-échantillonnées, 10 textures avec un ensemble de rotations aléatoires et 10 textures sans aucune transformation. Une image 3D est représentée par un ensemble de 64 images BMP 2D de dimension  $64^2$  pixels. Nous avons choisi le format BMP de par l'intérêt de sa large diffusion. Il est par ailleurs utilisé et proposé dans la plupart des bases de données en image et texture. En effet, la mise en place d'un logiciel capable d'utiliser ce format est simple et rapide. Un logiciel de visualisation capable de lire les images 3D proposées dans la base est disponible sur le site web <sup>2</sup>. Afin de proposer une taille suffisamment importante pour des tests de classification, les textures volumétriques générées sont de dimension  $64^3$  pixels. Ce choix permet également un bon compromis concernant l'espace disque. Enfin, les images utilisées pour les tests de segmentation (images qui contiennent

---

<sup>2</sup>[http://www.rfai.li.univ-tours.fr/fr/ressources/3Dsynthetic\\_images\\_database.html](http://www.rfai.li.univ-tours.fr/fr/ressources/3Dsynthetic_images_database.html)

### 3.1. BASE DE TEXTURES SOLIDES POUR L'ÉVALUATION DE MÉTHODES DE CLASSIFICATION ET DE SEGMENTATION

---

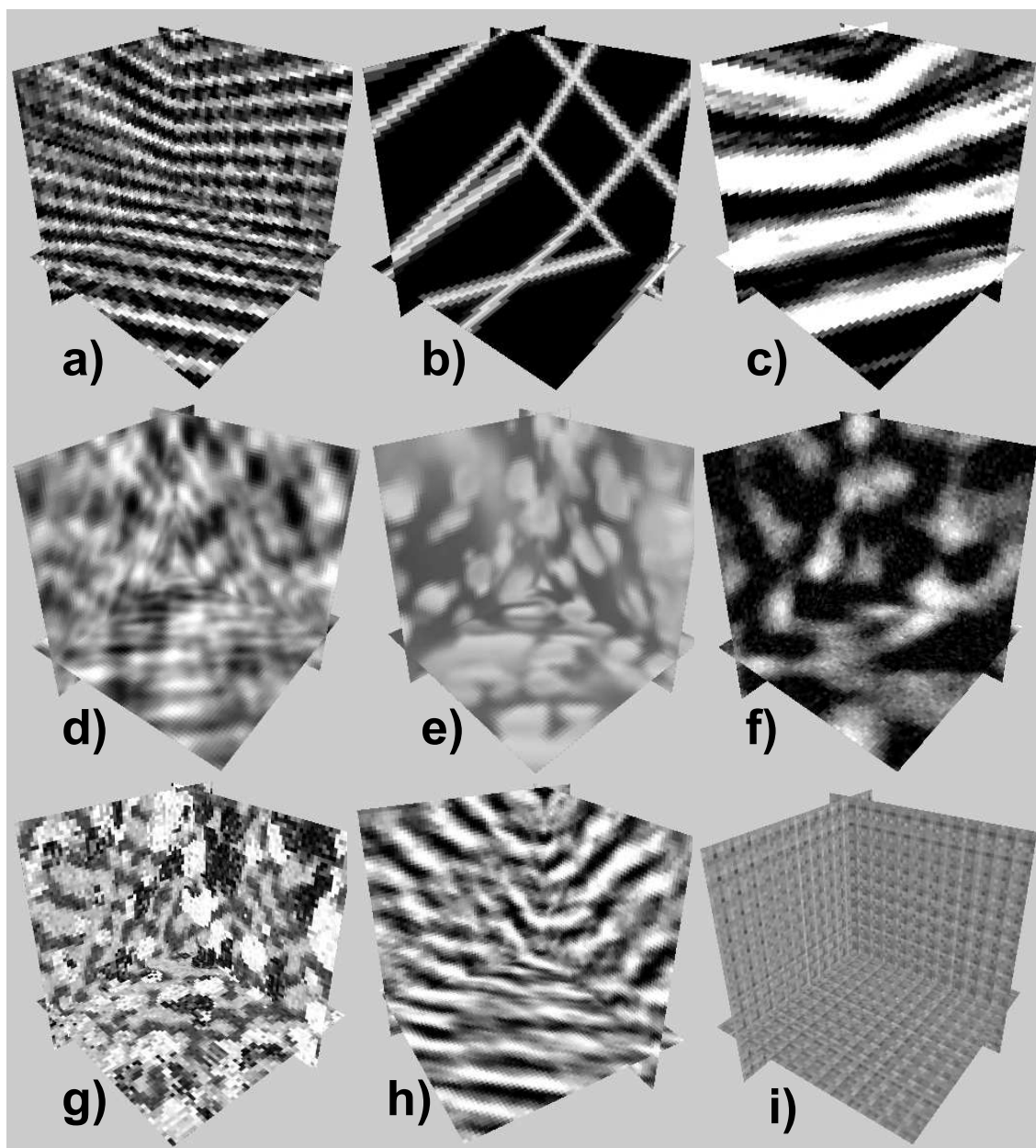


FIG. 3.5 – [a-c] Textures solides avec rotation aléatoire, [d-e] Deux textures floutées, [f-g] Deux textures bruitées, [h-i] Deux textures solides sous-échantillonnées

plus d'une seule texture) ont une taille de  $128^3$  offrant un degré de liberté plus important pour l'emplacement des différentes textures solides dans l'image 3D. Ces images sont actuellement disponibles au nombre de 15 : 5 textures contenant 2 classes, 5 textures contenant 3 classes, 5 textures contenant 4 classes.

Pour chacune des images 3D générées, un fichier XML associé résume l'ensemble des caractéristiques. La racine de l'arbre XML désigne une image 3D pouvant contenir une ou

plusieurs textures solides. Pour un problème de classification les images 3D utilisées correspondent à une unique texture solide alors que pour une problématique de segmentation, les images 3D contiennent plusieurs textures volumétriques différentes. Une texture solide est définie par un conteneur (emballage), par un nom (correspondant au nom d'une classe de textures), par le type de méthode de synthèse utilisée, par l'ensemble des propriétés et transformations qui lui sont appliquées. Un conteneur est utilisé puisque dans le cas d'un problème de segmentation, une texture volumétrique n'est pas forcément définie dans un cube. Actuellement, une texture solide peut être créée à l'intérieur de trois formes différentes (cube, sphère ou ellipse), celles-ci étant caractérisées par une dimension, une localisation ainsi qu'une orientation particulière. Les propriétés sont dépendantes du type de méthode de synthèse utilisé pour construire une texture solide. Par exemple, une texture volumétrique générée à partir de la transformée de Fourier inverse dépend du spectre de puissance utilisé. Pour pouvoir diffuser et exploiter ces fichiers XML, nous avons créé un fichier DTD disponible en partie en annexe C.

Cette base de données est actuellement disponible en accès libre <sup>2</sup> sur le web. De ce fait, un chercheur peut l'utiliser pour évaluer des méthodes d'analyse de textures solides à l'aide d'expérimentations de classification ou de segmentations. Il est bien sûr possible de compléter cette base de données avec d'autres méthodes de synthèse le but étant d'accroître le nombre de classes et d'images. Actuellement cette base de données contient 95 classes différentes ce qui est suffisant si nous comparons avec les bases de textures 2D. Par exemple, la base de Brodatz [Brodatz, 1966], qui constitue un standard pour l'évaluation d'algorithmes, contient 112 classes différentes.

## 3.2 Extension 3D de la méthode des motifs binaires (LBP)

Dans la littérature, nous n'avons pas trouvé d'extension 3D de la méthode des motifs binaires (*local binary pattern*). C'est pourquoi dans [Paulhac *et al.*, 2008], nous avons proposé une extension de cette méthode au domaine 3D. Le but de ce travail était de démontrer l'intérêt d'une approche vraiment 3D (LBP 3D) en comparaison de son équivalent 2D. Pour cela nous avons mis en place un protocole d'évaluation en classification utilisant notre base de textures solides. Ces méthodes sont testées selon plusieurs cas de figure avec une base de test contenant des textures sans aucune transformation, une base de test bruitée, et une base de test de textures avec rotations. Sur l'ensemble des tests réalisés, l'approche 3D se révèle être plus robuste avec des différences considérables concernant la troisième base de test. L'intérêt majeur de la méthode LBP 3D par rapport à la méthode LBP 2D est qu'elle conserve la propriété d'invariance à la rotation dans le domaine 3D.

L'algorithme classique LBP 2D [Ojala *et al.*, 1996] permet de calculer jusqu'à  $2^8$  unités de textures (avec  $R = 1$ ), seulement dans le domaine 3D, le nombre de motifs augmente de façon considérable avec  $2^{26} = 6.7108864 \times 10^7$  unités de textures possibles. C'est la raison pour laquelle nous avons choisi de baser notre extension 3D sur la méthode LBP proposée dans [Ojala et Pietikäinen, 2002] où le nombre de motifs possibles est de  $P + 2$  avec  $P$  le

---

<sup>2</sup>[http://www.rfai.li.univ-tours.fr/fr/ressources/3Dsynthetic\\_images\\_database.html](http://www.rfai.li.univ-tours.fr/fr/ressources/3Dsynthetic_images_database.html)

### 3.2. EXTENSION 3D DE LA MÉTHODE DES MOTIFS BINAIRES (LBP)

nombre de voisins pour une unité de textures. De plus, cette méthode possède la propriété d'être invariante à la rotation.

En 2D, nous avons vu dans la partie état de l'art que le voisinage autour d'un pixel central, est réparti de façon circulaire. En respectant ce raisonnement, nous avons choisi de définir le voisinage d'un voxel central sur une sphère afin de proposer une méthode LBP 3D. Cette sphère est construite de la façon suivante : pour un voxel central  $g_c$  ayant comme coordonnée  $(0, 0, 0)$ , les coordonnées de  $g_{pp'}$  sont données par  $(R \cos(\pi p'/(S-1)) \cos(2\pi p/P), R \cos(\pi p'/(S-1)) \sin(2\pi p/P), R \sin(p'\pi)/(S-1))$  avec  $R$  le rayon de la sphère,  $S$  le nombre de cercles utilisés pour construire la sphère, et  $P$  le nombre de vertex dans chaque cercle (Figure 3.6). Le voisinage ainsi construit, l'opérateur de textures LBP 3D peut être défini de la façon suivante :

$$LBP_{P',R}^{riu2} = \begin{cases} \sum_{p=0}^{P'-1} s(g_p - g_c) & \text{if } U(LBP_{P',R}) \leq V \\ P' + 1 & \text{sinon} \end{cases} \quad (3.2)$$

avec  $P' = (S-2) \times P + 2$ ,

$$s(x) = \begin{cases} 1, x \geq 0 \\ 0, x < 0 \end{cases} \quad (3.3)$$

De la même façon qu'en 2D,  $U$  est une fonction de mesure d'uniformité qui compte le nombre de régions uniformes noires et blanches présentes sur un motif LBP 3D. Afin de permettre le bon fonctionnement de cette fonction, nous construisons un graphe à partir de tous les vertex de la sphère. Chacun des vertex de la sphère est connecté à ses plus proches voisins de façon à obtenir un graphe qui soit connexe. En utilisant ce graphe, nous appliquons un algorithme de croissance de région pour identifier les différentes régions présentes sur un motif 3D. En 2D, un motif est défini comme uniforme si le nombre de régions est inférieur ou égal à 2. En 3D, nous avons relaxé cette condition avec  $V \in \{2, 3\}$ . Si  $V = 3$ , sont considérés comme uniforme les motifs dont le nombre de régions est inférieur ou égal à 3. Nous proposons d'alléger la condition d'uniformité car lorsque le rayon de la sphère augmente, la proportion des motifs uniformes diminue de façon importante (Table 3.1). Avec un rayon  $R = 3$  et avec  $V = 2$ , le pourcentage de motifs uniformes est inférieur à 50%. En conséquence, il est possible de perdre un grand nombre d'informations sur une texture.

|            | V=2   | V=3   |
|------------|-------|-------|
| R=1,P'=26  | 90.01 | 96.22 |
| R=2,P'=98  | 59.92 | 80.36 |
| R=3,P'=218 | 46.01 | 61.28 |

TAB. 3.1 – Pourcentage des motifs uniformes

Avec cette méthode, nous sommes capables de construire un code LBP sur des images texturées 3D. Cependant, en 2D le code  $LBP_{P,R}^{riu2}$  correspond à un unique motif LBP alors qu'avec notre méthode 3D plusieurs motifs LBP sont identifiés par un même code (Figure 3.7). Il est donc possible de trouver deux textures différentes qui génèrent le même histogramme LBP. Idéalement, il faudrait calculer un code unique pour chaque motif uniforme

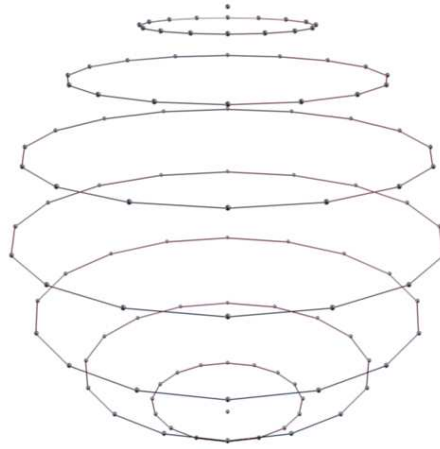


FIG. 3.6 – Représentation d'un motif local binaire 3D ( $S=9$ ,  $R=2$ ,  $P=16$ )

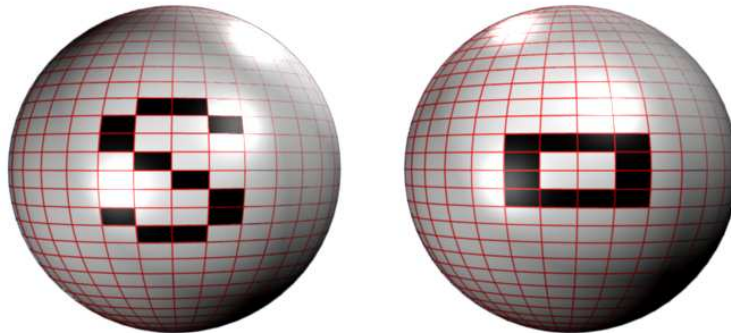


FIG. 3.7 – Exemple de deux motifs LBP 3D ayant le même code LBP ( $LBP_{P',R}^{riu2} = 12$ ). Ici, la valeur des vertex en noir est égale à 1 alors que la valeur des autres vertex est de 0.

de la même façon qu'en 2D. Cependant, le nombre de codes LBP possibles augmenterait de façon considérable.

### 3.3 Expérimentations en classification

Les images 3D ne contenant qu'une seule texture volumétrique peuvent être utilisées dans le cadre d'une problématique de classification. Il s'agit alors de déterminer la classe d'appartenance d'une texture à partir de descripteurs et d'algorithmes de classification. Les images contenant plusieurs textures solides peuvent aussi être utilisées en classification. L'objectif est alors d'attribuer un label à un voxel pour aboutir à une segmentation. Un grand nombre de méthodes ont été proposées pour l'évaluation des algorithmes de segmentation et de classification [Zhang, 1996]. Dans le prochain paragraphe, nous faisons une revue rapide de quelques-unes de ces méthodes d'évaluation.

#### 3.3.1 Évaluation d'algorithmes de classification d'images

Le but d'un système de classification est de décider la classe d'appartenance d'une image donnée. De façon générale, un système de classification est constitué de trois parties principales. La première consiste à extraire de l'image traitée un ensemble de caractéristiques. Dans le cadre d'un problème de reconnaissance de textures, nous avons vu dans l'état de l'art les descripteurs les plus souvent utilisés [Haralick *et al.*, 1973, Chellappa et Jain, 1993, Mallat, 1989, Ojala *et al.*, 1996]. La deuxième étape consiste à sélectionner les caractéristiques ce qui permet de réduire l'espace de description tout en gardant les caractéristiques les plus significatives. Durant la dernière étape, les vecteurs de caractéristiques sont soumis à des algorithmes de classification comme les réseaux de neurones, les machines à vecteurs de support, les  $k$  plus proches voisins, etc., afin de produire une décision. Une phase d'apprentissage permet au classificateur d'apprendre les caractéristiques de chaque classe. Durant une seconde phase, l'algorithme de classification étiquette les images de la base de test à partir de ce qu'il a appris.

Afin d'évaluer des systèmes de classification et de comparer leur robustesse pour une application donnée, une approche classique consiste à exploiter la matrice de confusion qui représente le nombre d'éléments  $c_{i,j}$  de la classe  $i$  classifiés à l'intérieur de la classe  $j$ . La matrice de confusion normalisée  $NCM$  peut être calculée de la façon suivante :

$$NCM_{i,j} = \frac{c_{i,j}}{\sum_{k=1}^T c_{i,k}} \quad (3.4)$$

avec  $T$  le nombre de classes considérées.

En utilisant cette matrice, il est alors possible de calculer plusieurs mesures :

le taux de vrais positifs,

$$TPrate_i = NCM_{i,i} \quad (3.5)$$

le taux de faux positifs,

$$FPrate_i = \sum_{j=1, j \neq i}^T NCM_{i,j} \quad (3.6)$$

l'exactitude,

$$Accuracy = \frac{\sum_{i=1}^T NCM_{i,i}}{T} \quad (3.7)$$

### 3.3. EXPÉRIMENTATIONS EN CLASSIFICATION

---

le taux d'erreur de classification  $ECR$  avec par exemple :

$$ECR = \sum_{i=1}^T \frac{0.5(\sum_{j=1, j \neq i}^T NCM_{i,j} + \sum_{j=1, j \neq i}^T \frac{NCM_{j,i}}{T-1})}{T} \quad (3.8)$$

Le taux ECR considère deux types d'erreurs : les éléments d'une classe  $i$  faussement classifiés comme éléments d'une autre classe et les éléments classifiés dans une classe donnée  $j$  mais appartenant à une autre classe  $i$ .

Dans [Martin *et al.*, 2006], les auteurs proposent une approche intéressante qui consiste en la construction d'une matrice de confusion prenant en considération plusieurs vérités terrains traduisant les contradictions entre experts. Cette méthode est intéressante dans le cadre d'analyses d'images naturelles nécessitant habituellement plusieurs classifications expertes.

Une autre façon d'examiner la performance d'un ensemble de classificateurs est l'utilisation des graphes ROC (*receiver operating characteristics graph*) qui permettent de visualiser et de sélectionner un ensemble de classificateur en se basant sur leur performance. Les graphes ROC sont des graphes 2D dans lequel l'axe des ordonnées  $TPrate$  (formule 3.5) représente le taux de vrai positifs et l'axe des abscisses  $FPrate$  (formule 3.6) représente le taux de faux positifs d'un classificateur. Si  $T$  classes sont considérées, alors il est possible de générer  $T$  graphiques. Pour chaque classificateur, nous pouvons calculer une paire  $(FPrate, TPrate)$ , ce qui permet la comparaison de leur performance (Figure 3.8). Le point  $(0, 1)$ , désigné par le classificateur C, représente une classification parfaite. Pour plus d'information sur les méthodes d'analyse associées aux graphes ROC, Fawcett présente un guide sur leur utilisation en recherche afin de promouvoir des pratiques de l'évaluation qui soient meilleures [Fawcett, 2006].

Il existe un grand nombre d'autres méthodes pour l'évaluation de systèmes de classification passé en revue dans [Ferri *et al.*, 2008]. Les auteurs proposent une description et une étude des relations entre les mesures de performance les plus couramment utilisées en classification. Ils relèvent une similarité importante entre les différentes mesures mais également des différences significatives entre certaines.

#### 3.3.2 Évaluation de nos descripteurs en classification

Nous comparons, dans une problématique de classification, notre méthode HUF (*Human Understandable Features*) avec les matrices de cooccurrence d'Haralick [Haralick, 1979, Haralick *et al.*, 1973], la méthode LBP 2D présentée dans [Ojala et Pietikäinen, 2002] et notre extension 3D de la méthode LBP [Paulhac *et al.*, 2008]. Pour cela nous utilisons notre base de données de textures solides [Paulhac *et al.*, 2009] contenant 95 classes de textures différentes. Pour ces expérimentations de classification, nous utilisons l'algorithme machine à vecteurs de support (SVM) avec un noyau RBF (*radial basic function*)  $K(x, x') = \exp(-\gamma \|x - x'\|^2)$  avec  $x$  et  $x'$  deux vecteurs de dimensions  $p$  et  $\gamma > 0$ . Ce type de noyau permet d'envisager le cas où la relation entre les différentes classes n'est pas linéaire. De plus, le nombre de paramètres est faible si nous comparons avec d'autres noyaux comme le noyau polynomial. Dans notre implémentation, nous utilisons une librairie gratuite disponible sur le web et appelée "LIBSVM" [Chang et Lin, 2001]. Durant nos tests de classification, nous utilisons quatre

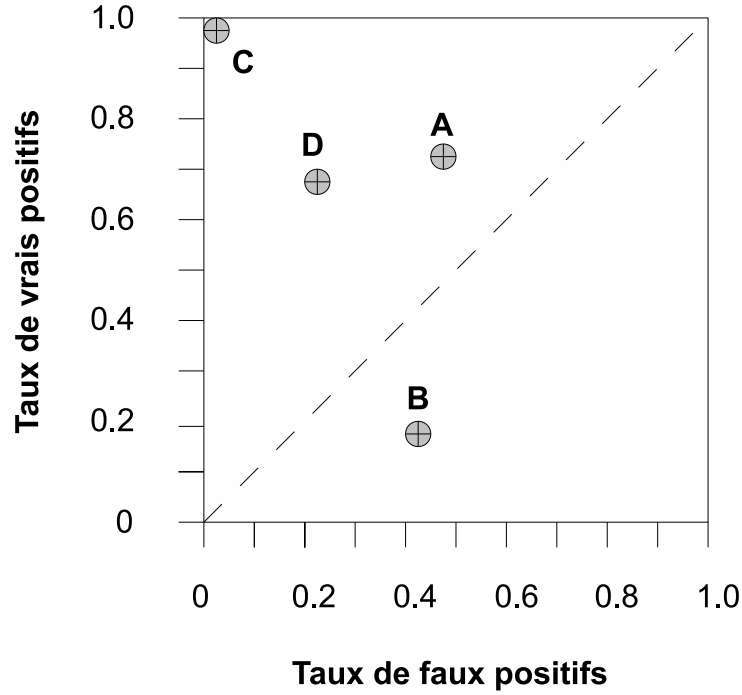


FIG. 3.8 – Un exemple de graphe ROC

images 3D par classe pour la phase d'apprentissage et six images 3D pour la phase de test. Pour calculer les caractéristiques d'Haralick, nous calculons une matrice de cooccurrence selon 13 directions ce qui permet d'obtenir une description plus riche [Haralick, 1979, Haralick *et al.*, 1973]. Dans ces tests nous utilisons les 14 caractéristiques d'Haralick présentées en annexe A. Cette méthode comprend deux paramètres : il s'agit de la distance entre voxels  $d$  et du paramètre de quantification  $q$  des niveaux de gris. Pour la méthode LBP 3D nous proposons, de façon identique à [Ojala et Pietikäinen, 2002], de tester l'opérateur  $LBP_{P,R}^{riu2}$  selon trois résolutions spatiales différentes et trois résolutions angulaires. Pour une valeur de rayon donnée, si le nombre de vertex est trop peu important, alors la probabilité d'obtenir des motifs uniformes est restreinte. En 3D, nous calculons les trois opérateurs suivants avec  $LBP_{26,1}^{riu2}$ ,  $LBP_{98,2}^{riu2}$  et  $LBP_{218,3}^{riu2}$  en considérant  $V \in \{2, 3\}$ . En 2D, les opérateurs  $LBP_{8,1}^{riu2}$ ,  $LBP_{16,2}^{riu2}$  et  $LBP_{32,3}^{riu2}$  sont calculés pour chaque image 2D dans le plan  $(x, y)$  de chaque texture volumétrique. Concernant la méthode HUF nous utilisons plusieurs ondelettes avec les ondelettes de Haar et Daubechie [Unser, 1995] ainsi que plusieurs résolutions. De plus, l'ensemble des caractéristiques proposées dans le chapitre 2 sont utilisées : la granularité, le volume, la compacité, la régularité, le contraste, la rugosité, la directionnalité. Afin d'évaluer la performance des différentes méthodes utilisées, nous présentons les pourcentages de reconnaissance, calculés à l'aide d'une matrice de confusion (formule 3.7) et sur l'ensemble des bases de test proposées.

Dans la section précédente, nous avons vu que notre base de données contenait 95 classes de textures solides différentes et que pour chacune d'elles plusieurs exemples de textures étaient disponibles, avec des textures normales sans aucune transformation et des

### 3.3. EXPÉRIMENTATIONS EN CLASSIFICATION

---

|     | q=8   | q=16  | q=32  | q=64  | q=128 |
|-----|-------|-------|-------|-------|-------|
| d=1 | 69.81 | 58.34 | 31.63 | 19.85 | 8.43  |
| d=2 | 63.09 | 38.84 | 21.44 | 8.78  | 2.63  |
| d=3 | 52.02 | 26.36 | 14.76 | 5.44  | 2.10  |

TAB. 3.2 – Pourcentage de reconnaissance avec les 14 caractéristiques d’Haralick sur la base de test sans bruit (95 classes, 4 images d’apprentissage et 6 images de test par classe).

|     | q=8   | q=16  | q=32  | q=64  | q=128 |
|-----|-------|-------|-------|-------|-------|
| d=1 | 50.96 | 35.85 | 17.39 | 10.54 | 4.50  |
| d=2 | 44.11 | 22.84 | 11.95 | 5.62  | 1.93  |
| d=3 | 35.67 | 14.41 | 8.78  | 3.33  | 1.58  |

TAB. 3.3 – Pourcentage de reconnaissance avec les 14 caractéristiques d’Haralick sur la base de test contenant à la fois des images normales et bruitées (95 classes, 4 images d’apprentissage et 6 images de test par classe).

|            | V=2   | V=3   |
|------------|-------|-------|
| R=1,P'=26  | 85.87 | 86.04 |
| R=2,P'=98  | 89.91 | 86.75 |
| R=3,P'=218 | 88.68 | 83.83 |

TAB. 3.4 – Pourcentage de reconnaissance avec la méthode LBP 3D sur la base de test sans bruit ( $P'$  caractéristiques, 95 classes, 4 images d’apprentissage et 6 images de test par classe).

|            | V=2   | V=3   |
|------------|-------|-------|
| R=1,P'=26  | 47.62 | 46.74 |
| R=2,P'=98  | 52.37 | 48.11 |
| R=3,P'=218 | 50.26 | 46.39 |

TAB. 3.5 – Pourcentage de reconnaissance avec la méthode LBP 3D sur la base de test contenant à la fois des images normales et bruitées ( $P'$  caractéristiques, 95 classes, 4 images d’apprentissage et 6 images de test par classe).

|          | Base normale | Base normale + bruit |
|----------|--------------|----------------------|
| R=1,P=8  | 85.21        | 47.14                |
| R=2,P=16 | 89.14        | 52.02                |
| R=3,P=32 | 88.68        | 48.94                |

TAB. 3.6 – Pourcentage de reconnaissance avec la méthode LBP 2D sur les deux bases de test ( $P$  caractéristiques, 95 classes, 4 images d’apprentissage et 6 images de test par classe).

textures avec des distorsions comprenant des textures bruitées, des textures floutées, des textures sous-échantillonnées ainsi que des textures ayant subi des rotations aléatoires selon les trois axes  $x$ ,  $y$  et  $z$ . Durant nos tests, nous utilisons une base d’apprentissage et deux

### 3.3. EXPÉRIMENTATIONS EN CLASSIFICATION

---

|             | haar  | db2   | db4   | db6   | db8   |
|-------------|-------|-------|-------|-------|-------|
| $\beta = 1$ | 60.36 | 62.45 | 63.33 | 61.57 | 60.87 |
| $\beta = 2$ | 68.78 | 68.08 | 70.71 | 69.48 | 69.31 |

TAB. 3.7 – Pourcentage de reconnaissance avec notre méthode HUF sur la base de test contenant des images normales ( $7\beta$  caractéristiques, 95 classes, 4 images d'apprentissage et 6 images de test par classe).

|             | haar  | db2   | db4   | db6   | db8   |
|-------------|-------|-------|-------|-------|-------|
| $\beta = 1$ | 44.46 | 46.22 | 46.92 | 43.40 | 41.82 |
| $\beta = 2$ | 53.77 | 53.42 | 53.77 | 54.13 | 53.07 |

TAB. 3.8 – Pourcentage de reconnaissance avec notre méthode HUF sur la base de test contenant à la fois des images normales et bruitées ( $7\beta$  caractéristiques, 95 classes, 4 images d'apprentissage et 6 images de test par classe).

bases de tests différentes. Notre base d'apprentissage contient, pour chaque classe de textures, quatre exemples différents de textures n'ayant subi aucune transformation (texture normale). Concernant les deux bases de tests utilisées, la première contient, pour chaque classe de textures solides représentées, six textures volumétriques normales (sans transformations) alors que la deuxième contient trois textures normales mais également trois textures bruitées. Les résultats de classification obtenus sur la première base de test par l'ensemble des méthodes sont présentés dans la table 3.2 (résultats avec les caractéristiques d'Haralick), la table 3.4 (résultats avec LBP 3D), la table 3.6 (résultats avec LBP 2D) et la table 3.7 (résultats avec notre méthode HUF). Celles-ci représentent le pourcentage de reconnaissance (formule 3.7) des textures solides sur l'ensemble de la base de test.

Comme nous pouvons le constater, la méthode des motifs binaires 3D donne les meilleurs résultats de classification (environ 90%) suivie de près par la méthode LBP 2D. La méthode HUF permet d'obtenir des résultats de classification légèrement supérieurs (70%) à ceux obtenus à partir des caractéristiques d'Haralick. Avec la méthode LBP 3D, les meilleurs résultats sont obtenus avec l'opérateur  $LBP_{98,2}^{riu2}$  et le paramètre d'uniformité  $V = 2$ . Si nous comparons avec la méthode LBP 2D, les résultats de classifications sont relativement proches. Le gros intérêt de la méthode LBP 3D par rapport à la méthode LBP 2D est qu'elle permet de conserver la propriété d'invariance à la rotation dans le domaine 3D. Concernant la méthode des matrices de cooccurrence d'Haralick, le meilleur résultat de classification est obtenu avec une distance  $d = 1$  et avec le paramètre de quantification  $q = 8$ . Nous pouvons également constater que lorsque les paramètres de distance et notamment de quantification augmentent, les résultats de classification s'effondrent de façon importante. Enfin notre méthode présente des résultats de classification relativement stable quelle que soit l'ondelette choisie. Nous pouvons également souligner l'importance de la multirésolution qui permet une forte amélioration des résultats de classification.

Sur la base de test contenant des images bruitées, les meilleurs résultats sont obtenus avec la méthode HUF (54%). Les méthodes LBP 2D et 3D présentent de fortes difficultés à reconnaître les textures volumétriques ayant subi une distorsion (52%). En effet, les histogrammes générés à partir des images bruitées présentent d'importants changements dans la répartition des motifs binaires. Les caractéristiques d'Haralick fournissent les moins

bons résultats et se comportent de façon similaire à l'expérimentation sur la base de test précédente en présentant des résultats de classification qui chutent au fur et à mesure de l'augmentation des paramètres de distance  $d$  et de quantification  $q$ . Si nous comparons avec la base de test contenant des textures normales, la chute la plus importante des taux de classification concerne les méthodes LBP 2D et 3D car celles-ci se révèlent peu robustes sur la plupart des textures bruitées. Enfin, la multirésolution permet également d'améliorer les résultats de classification de notre méthode HUF de façon significative.

## 3.4 Expérimentations en segmentation

### 3.4.1 Évaluation d'algorithmes de segmentation d'images

Les méthodes comme les matrices de confusion permettent uniquement d'évaluer les approches de classification mais ne donnent pas d'évaluation de la segmentation produite [Martin *et al.*, 2006]. L'évaluation d'une segmentation ne peut se satisfaire uniquement d'une comparaison visuelle et nécessite l'utilisation de métriques. Plusieurs méthodes ont été proposées et celles-ci sont réparties selon deux familles : les méthodes d'évaluation supervisées qui nécessitent l'accès à une vérité terrain et les méthodes d'évaluation non-supervisées qui ne nécessitent aucune connaissance *a priori*.

Les méthodes d'évaluation supervisées mesurent le degré de similarité entre une ou plusieurs segmentations expertes et la segmentation générée par un algorithme. Le principal avantage de ce genre de méthodes est qu'elles permettent d'obtenir une évaluation très fine. Cependant, la génération d'une vérité terrain peut, dans certains cas, se révéler très difficile à obtenir. Son temps de construction est également une contrainte non négligeable surtout dans le domaine des images 3D. Ce problème ne se pose pas pour notre base de données de textures solides puisqu'il s'agit d'images de synthèse pour lesquelles il est facile de construire les vérités terrain à partir des informations contenues dans les fichiers xml (figure 3.9). Par ailleurs, durant l'évaluation d'une segmentation, plusieurs types d'erreurs peuvent être considérées : la sous-segmentation (des éléments sont manquants), la sur-segmentation (supplément de pixels/voxels), et enfin l'erreur de localisation. Dans [Chabrier *et al.*, 2008], une étude comparative de 14 critères d'évaluations supervisés est réalisée. Ceux-ci sont étudiés dans différentes conditions de dégradation (sous-segmentation, sur-segmentation, etc.). En conclusion de cet article, le critère de *Pratt* se révèle être le plus efficace en moyenne.

Les critères présentés dans [Chabrier *et al.*, 2008] correspondent à des méthodes permettant d'évaluer une segmentation à partir de la frontière des régions présentes dans une image. D'autres mesures permettent de donner une évaluation en se basant sur une métrique qui compare les différentes régions et non directement les frontières. C'est le cas de la mesure générique d'anomalie (*generic discrepancy measure*) [Cardoso et Corte-Real, 2005] que nous utiliserons pour nos évaluations. Celle-ci est basée sur la distance de partition qui est définie dans [Gusfield, 2002] de plusieurs façons :

**Definition 3.4.1.** *Étant donné deux partitions  $P$  et  $Q$  de  $S$ , la distance de partition correspond au nombre minimum d'éléments devant être supprimés de  $S$  de manière à ce*

que  $P$  et  $Q$  (restreint aux éléments restants) soient identiques.

**Definition 3.4.2.** La distance de partition est égale au nombre minimum d'éléments qui doivent être déplacés entre cluster dans  $P$  de façon à ce que la partition résultante soit égale à  $Q$ .

La mesure générique d'anomalie correspond à la distance de partition normalisée. Si nous considérons deux partitions  $P$  et  $Q$ , alors la mesure générique d'anomalie  $d_{ga}$  est définie de la façon suivante :

$$d_{ga} = d_{sym}(P, Q)/(N - 1) \quad (3.9)$$

avec  $N$  le nombre de pixels/voxels de l'image et  $d_{sym}$  la distance de partition.

Si la mesure générique normalisée est égale à 0, alors la segmentation générée est parfaite. À l'inverse, si tous les voxels sont étiquetés comme appartenant à la mauvaise région alors la mesure générique normalisée prendra la valeur de 1. Ce dernier cas de figure n'est possible que si  $P$  correspond à la partition nulle et  $Q$  à la partition infinie ou inversement.

Les méthodes d'évaluations non-supervisées permettent une évaluation quantitative et objective sans pour autant nécessiter de références. Ce genre de méthode est très intéressant dans certains types d'applications où il est parfois difficile, voire impossible, de construire une vérité terrain. Par exemple, générer une vérité terrain pour les images échographiques 3D est très difficile de par leur structure complexe mais également de par leur grande dimension. Beaucoup de métriques ont été présentées et certaines d'entre elles proposent de mesurer l'uniformité intra-région, la disparité inter-région, la forme, etc. Dans [Zhang *et al.*, 2008], les auteurs proposent un état de l'art de ces méthodes non-supervisées dans lequel sont présentées des expérimentations afin de les évaluer dans différentes situations. Nous ne nous étendons pas ici sur ce type de méthodes puisque nous ne les avons pas utilisées. En effet, elles ne sont généralement pas adaptées aux images texturées.

#### 3.4.2 Évaluation de nos descripteurs

Afin d'évaluer notre méthode (HUF : *Human Understandable Features*) dans une problématique de segmentation, nous avons choisi de la comparer avec deux méthodes classiques et performantes de la littérature : les matrices de cooccurrence d'Haralik (GLCM) [Haralick, 1979, Haralick *et al.*, 1973] et la méthode LBP 3D [Paulhac *et al.*, 2008]. Les résultats de segmentation sont produits en utilisant 15 textures solides différentes (Figure 3.10) générées à partir des textures présentes dans notre base de données. Parmi elles, 5 images 3D contiennent 2 classes de textures, 5 images 3D contiennent 3 classes de textures et les 5 dernières images 3D contiennent 4 classes de textures.

Les matrices de cooccurrence d'Haralik sont calculées selon 13 directions dans l'espace 3D. Le fait de construire une matrice selon plusieurs directions permet d'obtenir une matrice plus générale et plus représentative des textures présentes dans une image [Haralick, 1979, Haralick *et al.*, 1973]. À partir de cette matrice, nous avons calculé 14 caractéristiques différentes, classiquement utilisées, et dont les méthodes de calculs sont présentées en annexe A. Comme dans la partie classification, nous faisons varier les deux

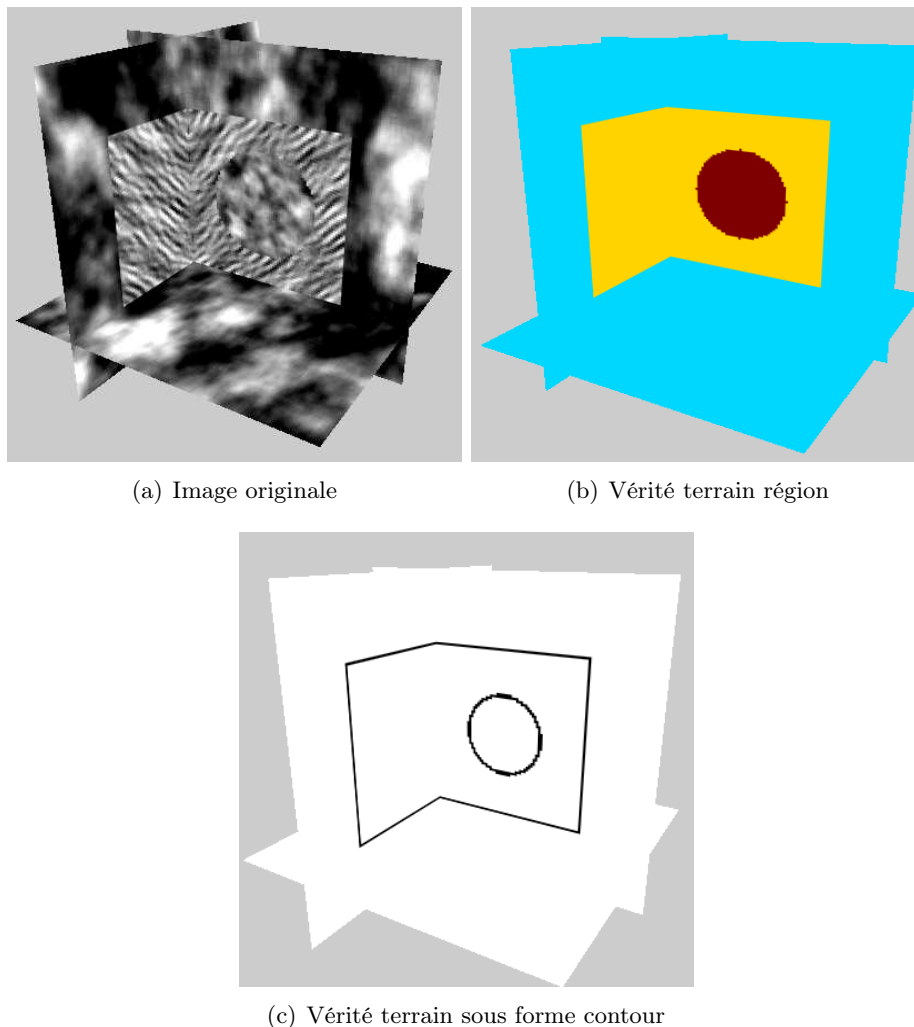
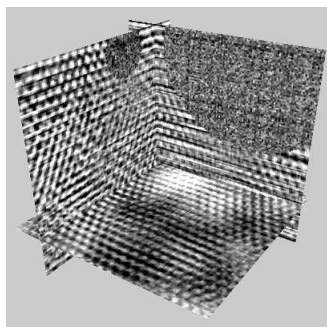


FIG. 3.9 – Exemple d’une image 3D contenant 3 textures solides avec ses vérités terrains associées

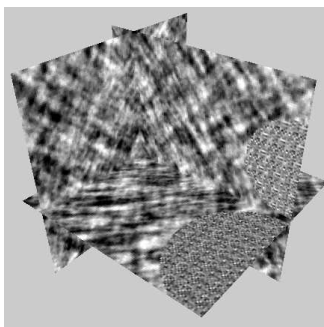
paramètres de cette méthode avec la distance  $d$  entre deux voxels et le paramètre de quantification  $q$ . Concernant notre méthode, l’ensemble des caractéristiques est utilisé comme dans la section précédente de classification. Des segmentations sont générées en faisant varier le nombre de résolutions  $\beta$  et en utilisant différentes ondelettes (haar, daubechie). Pour la méthode LBP 3D, nous proposons de calculer deux opérateurs différents avec  $LBP_{26,1}^{riu2}$ ,  $LBP_{98,2}^{riu2}$ . Enfin nous faisons varier le paramètre de régionalité  $N$  (voisinage autour d’un voxel), paramètre commun à chacune des trois méthodes. Dans la sous-section 3.4.2.1 suivante, nous expliquons comment les descripteurs HUF sont intégrés à l’intérieur d’un schéma de segmentation multirésolution.

### 3.4. EXPÉRIMENTATIONS EN SEGMENTATION

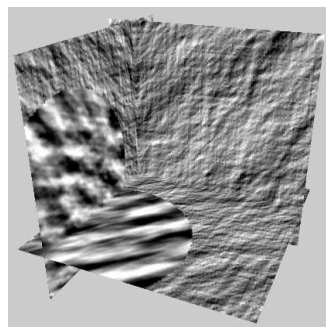
---



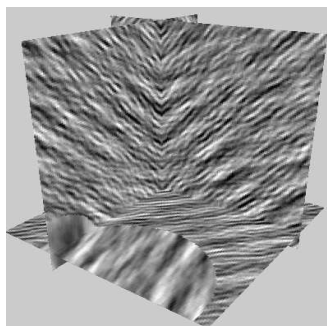
(a)



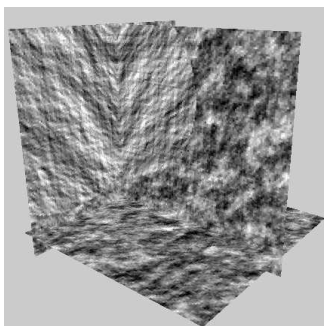
(b)



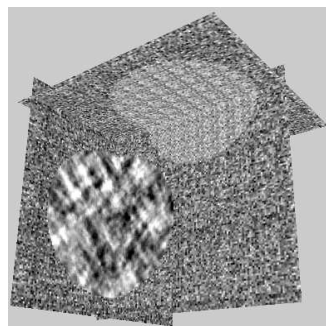
(c)



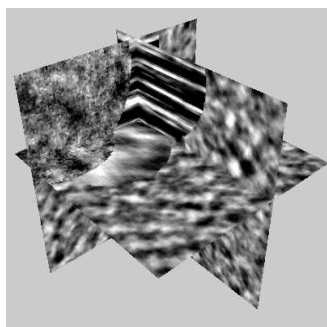
(d)



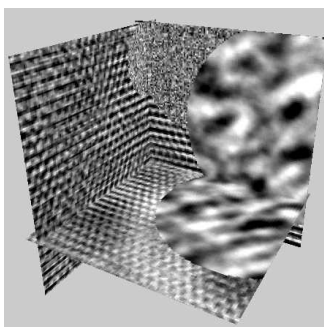
(e)



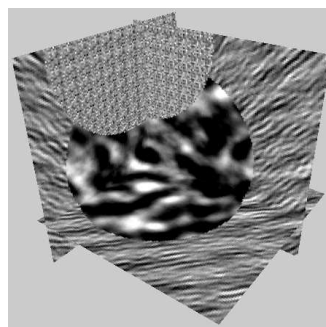
(f)



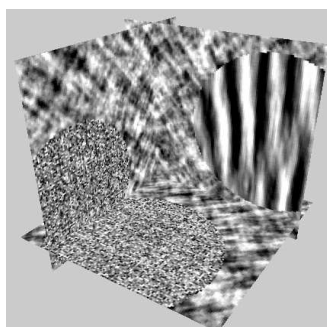
(g)



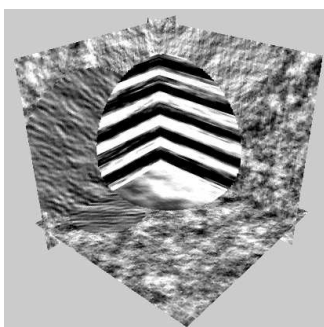
(h)



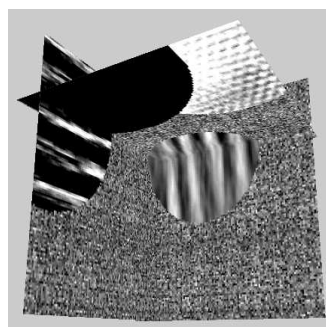
(i)



(j)



(k)



(l)

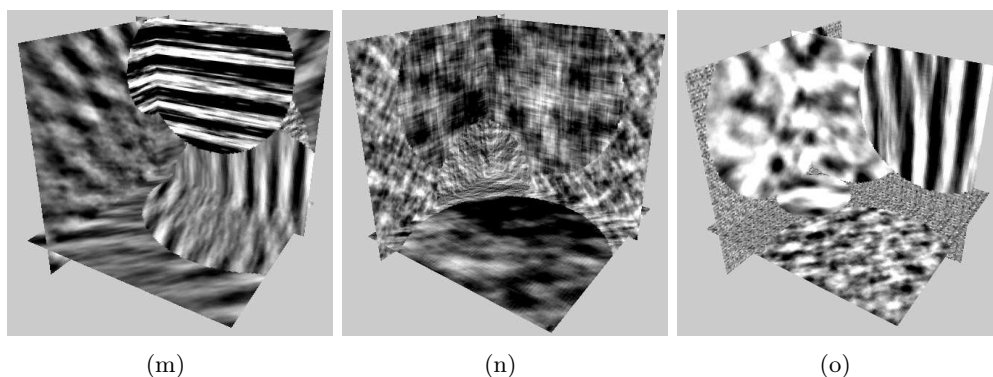


FIG. 3.10 – Images de textures volumétriques : [a-e] 2 classes de textures, [f-j] 3 classes de textures, [k-o] 4 classes de textures

### 3.4.2.1 Schéma de segmentation multirésolution avec les descripteurs HUF

L'approche proposée permet d'obtenir l'ensemble des caractéristiques de textures présentées dans le chapitre 2 et ceci sur plusieurs résolutions. Pour cela, nous avons vu qu'une décomposition en ondelette discrète est utilisée et que les caractéristiques de textures sont calculées à partir des sous-bandes coefficient de détail et d'approximation. Dans le cadre d'une segmentation, les caractéristiques de textures proposées sont calculées pour chacun des voxels de l'image à traiter. Afin d'avoir le même nombre d'attributs de textures, quelle que soit la résolution, il est nécessaire de sur-échantillonner l'ensemble des caractéristiques de textures, et ceci pour chaque résolution (Figure 3.11).

Un voxel est alors décrit par un vecteur contenant un ensemble d'attributs de textures. Un vecteur est composé par un maximum de  $7n$  caractéristiques de textures différentes avec  $n$  le nombre de résolutions choisi et 7 le nombre des caractéristiques de textures proposé : la granularité ( $f_{gran}$ ), les informations de forme sur les motifs d'une texture ( $f_{vol}$  et  $f_{comp}$ ), la régularité ( $f_{reg}$ ), le contraste ( $f_{con}$ ), la rugosité ( $f_{rug}$ ) et la directionnalité ( $f_{dir}$ ). Afin de produire une segmentation de l'ensemble des voxels, nous utilisons l'algorithme des K-means [Coleman et Andrews, 1979]. Celui-ci permet de classifier les voxels en sous-ensembles selon la valeur de leurs caractéristiques de textures. Les principaux avantages de cette méthode sont sa rapidité d'exécution et son faible coût en mémoire. Les images volumétriques sont souvent composées d'un grand nombre d'individus. Dans cette thèse, nous avons traité des images échographiques 3D de dimension  $300^3$  ce qui représente un grand nombre de voxels à classer. Il est donc nécessaire d'utiliser un algorithme de segmentation rapide et peu gourmand en mémoire. En terme de performance, l'algorithme des K-means ne garantit pas le retour de l'optimum global cependant il permet une classification efficace des voxels en très peu de temps. Cette méthode requiert donc le choix d'un nombre de classes attendues. Afin d'apporter une solution à ce problème, notre logiciel offre la possibilité de fusionner les différentes classes produites par l'algorithme des K-means *a posteriori*. Ainsi, l'utilisateur choisit, dans un premier temps, un grand nombre de classes pour produire une première segmentation (sur-segmentation). Par la suite, il peut choisir la segmentation la plus appropriée par visualisation et fusions de différentes classes. La fusion

### 3.4. EXPÉRIMENTATIONS EN SEGMENTATION

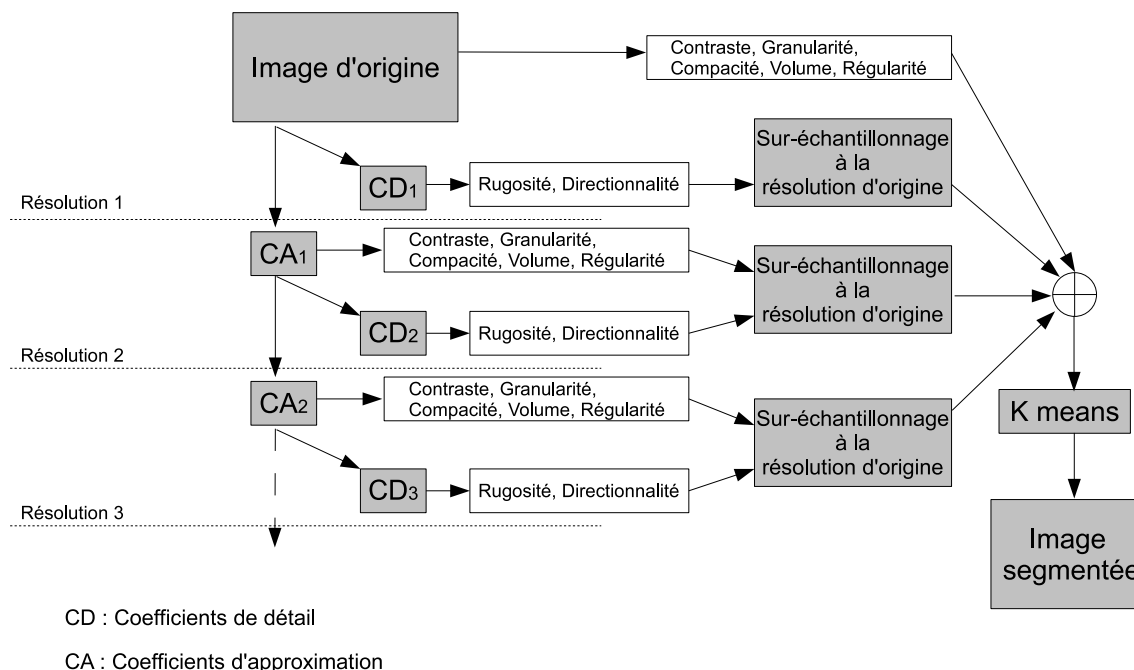


FIG. 3.11 – Schéma de segmentation multirésolution

des classes est réalisée en utilisant une classification ascendante hiérarchique à l'intérieur de laquelle les deux régions les plus proches sont fusionnées à chaque étape (Figure 3.12). La distance entre les régions est calculée en utilisant les caractéristiques de textures des centroïdes de chaque classe.

#### 3.4.2.2 Présentation des résultats de segmentation

Précédemment, nous avons vu que l'algorithme des K-means est utilisé afin de générer une segmentation. Cette méthode permet d'obtenir une classification des voxels en leur affectant une classe en fonction de leurs caractéristiques de textures. Dans le cadre de l'évaluation proposée ici, le couplage segmentation par l'algorithme des K-means et classification ascendante hiérarchique est également utilisé pour les méthodes LBP et GLCM. Pour chaque segmentation générée, l'utilisateur choisit le nombre de fusions nécessaire afin d'être le plus proche possible de la vérité terrain associée à l'image 3D traitée. Afin d'obtenir une évaluation des segmentations produites, nous avons choisi d'utiliser une métrique basée sur la comparaison de régions proposée dans [Cardoso et Corte-Real, 2005] (voir section 3.4).

Les résultats de segmentation sont présentés à partir de trois tables : la table 3.9 qui présente des résultats d'évaluation, calculés en utilisant des images 3D contenant 2 classes ; la table 3.10 dont les résultats sont calculés à partir de segmentation sur des images 3D contenant 3 classes ; et la table 3.11 qui illustre la qualité de segmentation obtenue sur des images 3D contenant 4 classes. Afin d'avoir une meilleure lisibilité, les résultats de distance

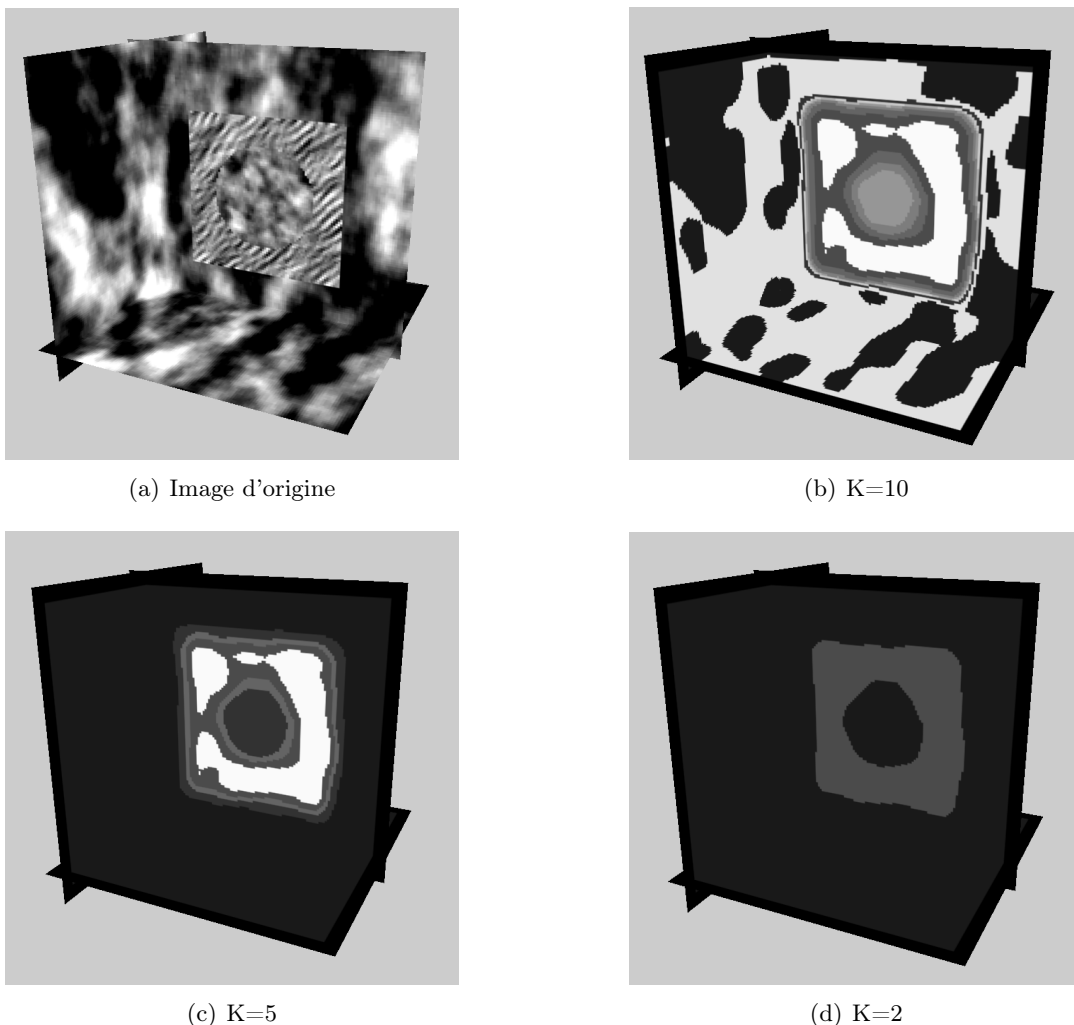


FIG. 3.12 – Exemple de segmentation avec  $K = 10$  et en fusionnant jusqu'à  $K = 2$

obtenus avec la mesure générique d'anomalie sont multipliés par 100 et sont compris entre 0 et 100.

Sur l'ensemble des segmentations présentées dans les tables 3.9, 3.10 et 3.11, les résultats de segmentation obtenus à partir des caractéristiques d'Haralick sont d'un point de vue quantitatif, en dessous des méthodes LBP 3D et HUF. Nous pouvons souligner le bon fonctionnement de cette approche sur des textures fines. La méthode GLCM se comporte de façon identique en classification et en segmentation avec une diminution importante des résultats lorsque la distance entre les paires de voxels augmente. Par exemple, sur la texture (b), la méthode GLCM obtient une valeur d'évaluation médiocre de 42,33 pour les paramètres  $d = 2$  et  $N = 9$  alors qu'elle génère une bonne segmentation avec une distance de 0,87 pour les paramètres  $d = 1$  et  $N = 9$ . Sur une grande partie des segmentations présentées, les moins bons résultats sont obtenus avec la méthode GLCM et pour une distance  $d = 2$ . Les méthodes LBP et HUF permettent d'obtenir les meilleurs résultats de

### 3.5. QUELQUES MOTS SUR LES TEMPS DE CALCUL ET LA NORMALISATION DES DONNÉES

---

segmentation avec un avantage pour la méthode HUF si nous considérons l'ensemble des textures utilisées dans ces tests de segmentation. Sur des textures relativement simples à segmenter, comme les textures (b), (c), (g) et (i), la méthode LBP 3D permet d'obtenir de très bons résultats qui restent proches de ceux obtenus avec la méthode HUF. En revanche, sur certaines textures plus complexes, comme les textures (m) et (n), la méthode LBP 3D est surpassée par la méthode HUF. En effet, la méthode HUF fournit de bons résultats sur l'ensemble des textures volumétriques, et dans le cas de textures très complexes, les résultats de segmentation sont très bons en comparaison aux méthodes GLCM et LBP 3D. Avec la méthode HUF, nous avons également effectué des segmentations en considérant deux niveaux de décomposition. Les résultats obtenus avec deux niveaux de résolution ne sont pas dans tous les cas les meilleurs, mais bien souvent, parmi ceux de très bonne qualité. Si pour un niveau de résolution nous obtenons une segmentation de très bonne qualité alors l'apport d'un deuxième niveau de décomposition n'est pas significatif. Cependant, pour certaines textures complexes, le deuxième niveau de résolution permet alors d'améliorer les résultats. Enfin, nous avons testé pour l'ensemble des méthodes plusieurs types de régionalités (voisinage) avec le paramètre  $N = \{5, 7, 9\}$ . Nous pouvons alors conclure qu'il n'existe pas de régionalité qui soit adaptée pour tous types de textures. Ce paramètre doit être réglé de façon à avoir une visibilité suffisante des motifs d'une texture avec un voisinage large pour segmenter des macro-textures et plus restreint pour segmenter des micro-textures. Il faut noter cependant que la multirésolution peut apporter une partie de réponse à ce problème puisqu'elle permet d'avoir une description à la fois globale et locale d'une texture.

### 3.5 Quelques mots sur les temps de calcul et la normalisation des données

La table 3.12 présente les temps d'exécution utilisés par les méthodes GLCM 3D, HUF et LBP 3D, pour segmenter une image 3D contenant  $128^3$  voxels. L'ordinateur utilisé est équipé d'un processeur Intel Core 2 Duo E8400 3.0 GHz et de 3.50 Go de RAM. La durée d'exécution de ces méthodes d'analyse de textures dépend énormément des paramètres choisis. Par exemple, la méthode  $LBP_{26,1}^{riu2}$  affiche des temps de calcul raisonnables alors que les temps d'exécution de la méthode  $LBP_{218,3}^{riu2}$  sont extrêmement longs. Plus le nombre de vertex de la sphère LBP est important, plus la méthode LBP 3D est lente. Le point sensible de cette méthode est la fonction de calcul de l'homogénéité des motifs ( $U(LBP_{P',R})$ ) qui effectue des croissances de région sur les différents vertex d'une sphère LBP. La méthode GLCM 3D se révèle très rapide avec un paramètre de quantification  $q = 8$ , cependant lorsque le nombre de niveaux de gris considéré est plus important alors les temps d'exécution peuvent devenir excessifs. En augmentant le paramètre de quantification, la taille de la matrice de cooccurrence devient plus importante et il est donc plus long d'en extraire les caractéristiques. Les temps d'exécution de la méthode HUF dépendent du nombre de caractéristiques choisies ainsi que du nombre de résolutions utilisées. Les temps de calcul sont relativement faibles même lorsque trois niveaux de décompositions sont considérés (compris entre 1 et 2 minutes).

### 3.6. CONCLUSION

| Méthodes                         | (a)          | (b)          | (c)          | (d)          | (e)           |
|----------------------------------|--------------|--------------|--------------|--------------|---------------|
| $LBP_{26,1}^{riu2}$ , N=5        | 0, 88        | 0, 60        | 2, 96        | 2, 36        | 30, 82        |
| $LBP_{26,1}^{riu2}$ , N=7        | 0, 74        | 0, 50        | <b>1, 18</b> | 0, 95        | 21, 49        |
| $LBP_{26,1}^{riu2}$ , N=9        | 2, 23        | 0, 72        | 1, 43        | 0, 97        | 20, 25        |
| $LBP_{98,2}^{riu2}$ , N=5        | 1, 23        | 0, 78        | 3, 74        | 3, 32        | 31, 88        |
| $LBP_{98,2}^{riu2}$ , N=7        | 1, 33        | 0, 51        | 2, 48        | 2, 35        | 20, 88        |
| $LBP_{98,2}^{riu2}$ , N=9        | 6, 87        | <b>0, 46</b> | 2, 06        | 0, 76        | 19, 68        |
| HUF, db2, $\beta = 1$ , N=5      | 0, 91        | 1, 73        | 8, 31        | 2, 07        | 27, 35        |
| HUF, db2, $\beta = 1$ , N=7      | <b>0, 71</b> | 0, 60        | 8, 42        | 2, 00        | 16, 70        |
| HUF, db2, $\beta = 1$ , N=9      | 1, 12        | 0, 50        | 2, 09        | 1, 04        | 16, 18        |
| HUF, db2, $\beta = 2$ , N={5,7}, | 1, 30        | 0, 87        | 1, 87        | <b>0, 72</b> | <b>12, 45</b> |
| HUF, haar, $\beta = 1$ , N=5     | 2, 52        | 1, 26        | 16, 78       | 2, 34        | 27, 82        |
| HUF, haar, $\beta = 1$ , N=7     | 2, 00        | 1, 30        | 8, 26        | 2, 76        | 17, 31        |
| HUF, haar, $\beta = 1$ , N=9     | 5, 96        | 2, 57        | 2, 60        | 1, 51        | 30, 32        |
| HUF, haar, $\beta = 2$ , N={5,7} | 4, 25        | 2, 39        | 9, 17        | 1, 98        | 14, 65        |
| GLCM, $q = 8$ , $d = 1$ , N=5    | 0, 87        | 9, 75        | 17, 50       | 3, 40        | 36, 66        |
| GLCM, $q = 8$ , $d = 1$ , N=7    | 1, 14        | 3, 95        | 3, 83        | 2, 60        | 31, 28        |
| GLCM, $q = 8$ , $d = 1$ , N=9    | 1, 00        | 0, 87        | 9, 29        | 2, 99        | 32, 23        |
| GLCM, $q = 8$ , $d = 2$ , N=5    | 1, 24        | 54, 16       | 21, 30       | 12, 33       | 29, 83        |
| GLCM, $q = 8$ , $d = 2$ , N=7    | 4, 10        | 33, 91       | 13, 82       | 43, 08       | 32, 46        |
| GLCM, $q = 8$ , $d = 2$ , N=9    | 18, 46       | 42, 33       | 22, 13       | 12, 34       | 33, 63        |

TAB. 3.9 – Evaluation de segmentations de textures solides contenant deux classes à partir de la mesure générique d'anomalie.

Dans sa version actuelle, toutes les caractéristiques de la méthode HUF ne sont pas comprises entre 0 et 1. Pour réaliser une segmentation optimale à l'aide de l'algorithme des K-means, il est alors nécessaire d'effectuer une étape de normalisation des données ou d'utiliser une distance appropriée. Nous avons choisi d'utiliser la distance de Mahalanobis qui demande la construction d'une matrice de covariance et permet ainsi de prendre en compte l'hétérogénéité des différentes caractéristiques. L'hétérogénéité des valeurs prises par nos descripteurs est plus difficile à gérer lorsqu'il s'agit de fournir à l'utilisateur une aide à l'interprétation : "Une valeur de granularité de 0, 92 correspond-elle à une amplitude forte, faible ou moyenne?". Une solution envisageable serait d'effectuer un étalonnage sur des textures typiques pour déterminer les valeurs extrêmes (minimales et maximales) ce qui permettrait de borner l'ensemble des descripteurs HUF entre 0 et 1.

## 3.6 Conclusion

Dans le but de valider notre méthode HUF (*Human understandable features*), nous présentons, dans ce chapitre, des tests expérimentaux en classification et en segmentation sur des images de synthèse pour lesquelles nous disposons de vérités terrains. Pour cela, nous la comparons avec des méthodes très utilisées et connues dans la littérature : les matrices de cooccurrence d'Haralick (GLCM) [Haralick, 1979, Haralick *et al.*, 1973], la méthode des

### 3.6. CONCLUSION

| Méthodes                         | (f)          | (g)          | (h)          | (i)          | (j)          |
|----------------------------------|--------------|--------------|--------------|--------------|--------------|
| $LBP_{26,1}^{riu2}$ , N=5        | 9, 76        | 6, 19        | 6, 71        | 29, 69       | 11, 18       |
| $LBP_{26,1}^{riu2}$ , N=7        | 9, 72        | 5, 51        | 3, 01        | 15, 08       | 17, 31       |
| $LBP_{26,1}^{riu2}$ , N=9        | 14, 07       | <b>1, 57</b> | 3, 21        | 39, 55       | 13, 73       |
| $LBP_{98,2}^{riu2}$ , N=5        | 9, 26        | 13, 21       | 20, 62       | 21, 27       | 14, 14       |
| $LBP_{98,2}^{riu2}$ , N=7        | 9, 80        | 25, 50       | 21, 33       | <b>3, 45</b> | 8, 88        |
| $LBP_{98,2}^{riu2}$ , N=9        | 9, 51        | 1, 68        | 2, 61        | 14, 97       | <b>7, 89</b> |
| HUF, db2, $\beta = 1$ , N=5      | 8, 10        | 23, 50       | 21, 76       | 8, 16        | 15, 82       |
| HUF, db2, $\beta = 1$ , N=7      | 4, 84        | 7, 61        | 24, 29       | 8, 77        | 20, 80       |
| HUF, db2, $\beta = 1$ , N=9      | 8, 48        | 5, 91        | <b>2, 38</b> | 9, 52        | 20, 37       |
| HUF, db2, $\beta = 2$ , N={5,7}  | 6, 52        | 7, 64        | 6, 88        | 8, 60        | 15, 67       |
| HUF, haar, $\beta = 1$ , N=5     | 8, 19        | 10, 00       | 26, 23       | 13, 72       | 25, 52       |
| HUF, haar, $\beta = 1$ , N=7     | <b>3, 90</b> | 10, 02       | 23, 96       | 8, 48        | 24, 82       |
| HUF, haar, $\beta = 1$ , N=9     | 9, 04        | 5, 84        | 8, 94        | 12, 07       | 22, 59       |
| HUF, haar, $\beta = 2$ , N={5,7} | 6, 90        | 7, 01        | 10, 70       | 9, 03        | 15, 61       |
| GLCM, $q = 8$ , $d = 1$ , N=5    | 10, 27       | 10, 69       | 20, 13       | 47, 10       | 18, 51       |
| GLCM, $q = 8$ , $d = 1$ , N=7    | 9, 11        | 9, 18        | 22, 06       | 29, 99       | 19, 51       |
| GLCM, $q = 8$ , $d = 1$ , N=9    | 9, 75        | 7, 05        | 16, 85       | 21, 94       | 17, 01       |
| GLCM, $q = 8$ , $d = 2$ , N=5    | 11, 79       | 14, 35       | 35, 53       | 65, 26       | 24, 61       |
| GLCM, $q = 8$ , $d = 2$ , N=7    | 36, 31       | 17, 81       | 22, 04       | 25, 63       | 36, 27       |
| GLCM, $q = 8$ , $d = 2$ , N=9    | 39, 44       | 15, 93       | 26, 68       | 39, 58       | 26, 08       |

TAB. 3.10 – Evaluation de segmentations de textures solides contenant trois classes à partir de la mesure générique d’anomalie.

motifs binaires (LBP) 2D [Ojala et Pietikäinen, 2002] et une extension 3D que nous avons proposé [Paulhac *et al.*, 2008]. Pour pouvoir effectuer ces tests, nous avons construit une base de données de textures solides contenant des images 3D en nombre et diversité suffisant pour mettre en place des évaluations significatives de classification et de segmentation. Cette base de données est en libre accès sur le web et il est donc possible pour un chercheur de l’utiliser pour valider une méthode en analyse de textures ou pour l’enrichir.

Durant les tests de classification, deux bases de test ont été utilisées : une première ne contenant que des textures volumétriques sans aucune transformation (normales) et une deuxième comprenant à la fois des textures normales et des textures bruitées. Sur la première base, les méthodes LBP 2D et 3D permettent d’obtenir les meilleurs résultats. Vient ensuite la méthode HUF suivie des caractéristiques d’Haralick. Cependant, sur la deuxième base de test, la méthode HUF dépasse les méthodes LBP 2D et 3D dont les taux de classification baissent fortement en présence de bruit. Notre méthode HUF se révèle donc moins sensible à certaines déformations comme le bruit ce qui prouve une robustesse plus large face à ce type de caractéristiques. En segmentation, la méthode HUF permet d’obtenir les meilleurs résultats si nous considérons l’ensemble des textures volumétriques considérées dans nos tests. La méthode LBP 3D permet elle aussi d’obtenir de bons résultats mais celle-ci est distancée par la méthode HUF, notamment sur les textures les plus complexes. Nous pouvons également constater qu’il est difficile de prédire une valeur de régionalité qui

### 3.6. CONCLUSION

| Méthodes                         | (k)          | (l)         | (m)          | (n)          | (o)          |
|----------------------------------|--------------|-------------|--------------|--------------|--------------|
| $LBP_{26,1}^{riu2}$ , N=5        | 38,05        | 12,19       | 58,12        | 57,72        | 43,08        |
| $LBP_{26,1}^{riu2}$ , N=7        | 40,27        | 15,27       | 53,08        | 47,10        | 27,92        |
| $LBP_{26,1}^{riu2}$ , N=9        | 33,06        | 1,57        | 52,25        | 51,42        | 30,32        |
| $LBP_{98,2}^{riu2}$ , N=5        | 39,01        | 13,21       | 43,36        | 51,09        | 44,26        |
| $LBP_{98,2}^{riu2}$ , N=7        | 32,09        | 4,76        | 32,47        | 46,52        | 40,07        |
| $LBP_{98,2}^{riu2}$ , N=9        | 22,41        | 16,41       | 42,02        | 48,51        | 28,18        |
| HUF, db2, $\beta = 1$ , N=5      | 31,16        | 8,60        | 41,66        | 51,81        | 31,23        |
| HUF, db2, $\beta = 1$ , N=7      | 30,77        | 9,88        | 26,35        | 40,22        | 20,47        |
| HUF, db2, $\beta = 1$ , N=9      | <b>19,12</b> | <b>1,14</b> | 24,81        | 32,22        | <b>10,19</b> |
| HUF, db2, $\beta = 2$ , N={5,7}  | 22,03        | 2,41        | 13,43        | 40,17        | 20,10        |
| HUF, haar, $\beta = 1$ , N=5     | 40,23        | 10,56       | 34,50        | 59,80        | 34,35        |
| HUF, haar, $\beta = 1$ , N=7     | 33,30        | 8,54        | 28,79        | 50,30        | 16,16        |
| HUF, haar, $\beta = 1$ , N=9     | 30,54        | 4,43        | <b>10,92</b> | <b>29,90</b> | 26,34        |
| HUF, haar, $\beta = 2$ , N={5,7} | 20,02        | 3,36        | 15,44        | 40,34        | 17,75        |
| GLCM, $q = 8$ , $d = 1$ , N=5    | 34,63        | 13,39       | 39,92        | 54,53        | 41,50        |
| GLCM, $q = 8$ , $d = 1$ , N=7    | 38,80        | 12,59       | 46,18        | 49,98        | 34,46        |
| GLCM, $q = 8$ , $d = 1$ , N=9    | 36,76        | 14,76       | 40,39        | 45,80        | 33,88        |
| GLCM, $q = 8$ , $d = 2$ , N=5    | 43,30        | 19,25       | 32,25        | 61,31        | 46,40        |
| GLCM, $q = 8$ , $d = 2$ , N=7    | 40,38        | 17,04       | 49,05        | 55,69        | 48,89        |
| GLCM, $q = 8$ , $d = 2$ , N=9    | 41,77        | 15,75       | 43,00        | 57,89        | 46,27        |

TAB. 3.11 – Evaluation de segmentations de textures solides contenant quatre classes à partir de la mesure générique d’anomalie.

| Méthode                           | temps d’exécution (en seconde) |
|-----------------------------------|--------------------------------|
| HUF, db2, $\beta = 1$ , N=7       | 86,98                          |
| HUF, db2, $\beta = 2$ , N={5,7}   | 113,55                         |
| HUF, db2, $\beta = 3$ , N={3,5,7} | 144,43                         |
| GLCM, $q = 8$ , $d = 1$ , N=7     | 68,19                          |
| GLCM, $q = 16$ , $d = 1$ , N=7    | 147,53                         |
| GLCM, $q = 32$ , $d = 1$ , N=7    | 579,26                         |
| $LBP_{26,1}^{riu2}$ , N=7         | 93,70                          |
| $LBP_{98,2}^{riu2}$ , N=7         | 437,72                         |
| $LBP_{218,3}^{riu2}$ , N=7        | 1143,46                        |

TAB. 3.12 – Temps d’exécution des méthodes HUF, GLCM 3D et LBP 3D pour segmenter une image 3D de taille  $128^3$ .

soit efficace pour tous types de textures. Dans le cas d’une macro-texture, ce paramètre doit être assez large afin d’obtenir une visibilité suffisante concernant les motifs. Son réglage est moins problématique pour les micro-textures cependant une valeur trop importante réduirait la précision sur les frontières des régions à identifier. Enfin il est important de souligner l’importance de la multirésolution qui permet, comme nous pouvons le voir dans nos tests de classification, une amélioration significative des résultats.

### 3.6. CONCLUSION

---

Ce chapitre valide donc de manière quantitative les performances de notre méthode HUF tant en segmentation qu'en classification. Dans le chapitre suivant, nous intégrons nos caractéristiques dans une application médicale afin de segmenter des images échographiques 3D. Nous nous attachons alors à démontrer l'intérêt d'utiliser des caractéristiques qui soient compréhensibles par un utilisateur afin d'être à même de segmenter différents types de textures à l'aide d'un unique système.