

Projet de département: Detection automatique de nodules.

Marie Rozand, Raphael Sivera, Yohann Salaun

25 juin 2013

Introduction

Les nodules sont des anomalies du corps de forme ellipsoïdale. Certains d'entre eux peuvent être des tumeurs et leur détection/surveillance permet d'éviter de graves cancers. Cependant, leur faible taille les rend difficilement détectables. Ce projet a ainsi pour but de détecter automatiquement des nodules pulmonaires à partir d'images scanner. Pour cela, l'utilisateur sélectionne grossièrement une partie de l'image puis l'algorithme indique où est situé le nodule.

Cette détection se décompose en plusieurs parties qui affinent tour à tour la segmentation de la zone étudiée et rendent un résultat de plus en plus précis. Pour cela, les méthodes suivantes sont utilisées tour à tour :

- Filtres de Gabor
- Analyse en Composantes Principales
- Transformée de Hough
- Coupe de graphes

1 Préliminaires

Nous reconstruisons donc tout d'abord une image 3D en niveaux de gris à partir des données DICOM. L'utilisateur peut alors sélectionner la zone qui l'intéresse. La segmentation s'effectue donc sur un petit échantillon d'image. Les nodules n'ont pas tous la même taille mais sont en général dans une image d'une cinquantaine de voxels de côté.

Pour réaliser une première segmentation de l'image on va classer les voxels en fonction de leurs réponses à différents filtres appliqués à l'image. En pratique on applique n filtres de Gabor à l'image, pour chaque filtre on obtient une nouvelle image. On en déduit donc un vecteur de n valeurs associé à chaque voxel. On effectue ensuite une ACP pour noter ces différents vecteurs. Un seuillage peut nous donner une approximation de la position du nodule. On peut aussi garder les valeurs telles quelles pour estimer une probabilité d'appartenance au nodule.

1.1 Filtres de Gabor

Principe Les filtres de Gabor est un filtre linéaire qui est le produit d'une sinusoïde par une enveloppe gaussienne. Son expression spatiale est donc de la forme :

$$G(x) = \cos(2\pi(u_0 \cdot x) + \phi) \exp\left(-\frac{x - x_0^2}{\sigma^2}\right)$$

On applique donc cette fonction à un masque de convolution pour avoir un filtre de Gabor. Les paramètres qui nous intéressent ici sont u_0 qui permet de définir l'orientation et la fréquence du filtre et σ qui permet de spécifier l'étendue du noyau gaussien.

Il existe des variantes où l'écart type de la gaussienne dépend de la direction sélectionnée. La direction peut être donnée par une matrice orthonormale U et si l'on note $\tilde{x} = Ux$ on peut écrire la fonction de

Gabor sous la forme.

$$G(x) = \cos(2\pi(\lambda \cdot \tilde{x})) \exp(-\Sigma \tilde{x}^2)$$

L'utilisation d'un écart-type anisotrope est souvent utilisé en détection des contours car de tels filtres de Gabor détectent les discontinuités dans une certaine direction. En effet, localement, la convolée d'une fonction régulière par une sinusoïde est nulle.

Application à la détection de textures sur l'image originale Les filtres de Gabor peuvent aussi caractériser les différentes textures d'une image. En appliquant un banc de filtre à une image on peut extraire les fréquences et les orientations caractéristiques des différentes régions de l'image. C'est cet aspect qui nous intéresse plus ici. Nous nous sommes donc limités à un sigme isotrope. Il est déjà assez difficile de choisir les autres paramètres. Nous avons au total un nombre assez élevé de paramètres et il n'est pas évident de pouvoir fournir un critère objectif sur la qualité du résultat fourni par le banc de filtre. Cependant il est difficile ici de différencier certaines textures avec la résolution des images scanner. Le choix de la fréquence est limité par la discrétisation de l'espace. On doit donc se contenter de moyennes et basses fréquences. On ne travaille que sur un portion de l'image, il n'y a donc pas de problèmes de bord en revanche pour des questions d'efficacité, on doit limiter la taille des masques de convolution, et donc l'écart type, à des valeurs raisonnables. Nous devons également effectuer de nombreuses convolutions sans pouvoir a priori moyenner ou utiliser une structure particulière. La convolution d'un noyau de taille s sur une image cubique de côté l est en $O(s^3 l^3)$. Pour un masque de taille 7 sur une image 50x50x50 on doit effectuer environ 43 000 000 d'opérations.

Par ailleurs l'ACP divise toujours l'espace en 2 parties alors que le nombre de zones visibles sur l'image varie. Une grande partie de l'image est occupé par le poumon mais on peut aussi voir des muscles intercostaux, des côtes, des tronçons d'arbre pulmonaires et bien sûr un ou des nodules. L'arbre pulmonaire apparaît de façon très similaire aux nodules mais le lissage en fait disparaître une grande partie. En revanche, les côtes et les muscles peuvent occuper d'importants volumes. On pourrait essayer d'améliorer l'algorithme de segmentation en estimant le nombre de classes présentes ou alors superviser l'apprentissage ou la segmentation. Dans ce cas il faudrait surement augmenter le nombre de filtres utilisés afin d'augmenter l'information disponible lors de la classification.

On a préféré utiliser ici la géométrie générale de l'objet recherché. On utilise donc un banc de filtre relativement simple constitué des 5 filtres suivants :

σ	u_x	u_y	u_z
1.	0	0	0
2.	0	0	0
2.	2.	0	0
2.	0	2.	0
2.	0	0	2.

1.2 Pré-traitement géométrique

On a vu que les résultats données par la partie précédente était difficilement utilisable à cause de la présence de la paroi thoracique. Afin de localiser le nodule, on remarque que, selon l'axe z (la verticale), le nodule doit correspondre à un maximum local de probabilité. On effectue donc sur chaque colonne de voxel une régression par un polynôme de degré 2 et on vérifie que l'on a un maximum significatif. On définit donc ainsi un cylindre à l'emplacement du nodule. La motivation principale est que, contrairement au nodule, la paroi thoracique occupe souvent toute la verticale même si le corps humain n'est pas parfaitement cylindrique. On effectue ensuite une étape d'ouverture (érosion puis dilatation) afin d'éliminer les éléments indésirable. Cette étape est relativement efficace pour situer grossièrement le nodule ce qui est nécessaire pour réaliser les étapes suivantes.

2 Délimitation par un ellipsoïde

Dans cette partie nous utilisons les propriétés géométriques du nodule pour obtenir une délimitation approchée. Nous savons qu'un module est globalement de forme ellipsoïdale, donc nous allons chercher quel ellipsoïde représente le plus fidèlement possible le nodule recherché. Pour plus d'efficacité en terme de temps calcul, nous allons passer par deux étapes. Une première étape qui donne une première approximation de l'ellipsoïde utilisant en particulier l'analyse en composante principale, puis une deuxième étape qui permet de raffiner les paramètres de l'ellipsoïde et qui utilise la transformée de Hough. Pour cela on utilise la matrice renvoyée par la première partie qui renvoie à chaque voxel la probabilité qu'il soit dans le nodule.

2.1 Approximation des paramètres

Le problème consiste à déterminer les 12 paramètres de l'ellipsoïde : les 3 coordonnées du centre et celles des 3 axes de l'ellipsoïde. Pour cela il faut déterminer quels voxels seront considérés, en première approximation, dans le nodule. On commence donc par faire un seuillage à l'aide de la fonction `Is_in` qui stocke dans un tableau les coordonnées des voxels dont la probabilité d'être dans le module est supérieure au seuil. Toute cette étape va se baser sur cet ensemble. Il faut donc bien choisir le seuil pour optimiser cette approximation.

2.1.1 Analyse en composante principale

Théorie L'analyse en composante principale [6] permet d'analyser un ensemble de données dans sa globalité et le synthétiser en donnant les n axes principaux. Ces données sont souvent complexes, représentées suivant de nombreuses variables. Cette méthode permet de transformer ces variables corrélées en nouvelles variables décorrélées et donc de réduire leur nombre. En effet, elle rend l'information moins redondante et on peut alors choisir les k variables les plus pertinentes du point de vue de l'inertie ou de la variance pour former ce qu'on appelle un dictionnaire.

Par exemple, dans une image, l'ACP va déterminer les deux axes qui expliquent le mieux la dispersion de l'objet interprété comme un nuage de points. Elle va aussi les ordonner par inertie expliquée, le second axe étant perpendiculaire au premier.

Soit f_1, \dots, f_n un ensemble de donnée avec $f_i \in \mathbb{R}^d$, on cherche un sous-espace tel que la projection soit la plus proche de la donnée originale, c'est à dire que la projection des données sur ce sous-espace soit de norme maximale. On peut exprimer ce problème sous la forme :

$$\max_V \sum_{i=1}^n \|P_V(f_i - \bar{f}_n)\|^2$$

où V est un sous-espace vectoriel et P_V est la projection sur le sous-espace vectoriel V et \bar{f}_n la moyenne des données f_i .

On peut aisément démontrer que ce sous-espace optimal est $V = Vect(u_1, \dots, u_k)$ où u_1, \dots, u_k sont les vecteurs propres associés aux k plus grandes valeurs propres de la matrice de covariance des données. Les valeurs propres représentent l'importance de la composante dans la représentation.

Application Pour déterminer les paramètres approchés de l'ellipsoïde, on commence donc par calculer le centre en moyennant les voxels contenus par le tableau de données renvoyé par la fonction `Is_in`.

On applique ensuite la méthode de l'Analyse en Composantes Principales sur ce même tableau où les données sont les n voxels du tableau et les variables sont les 3 coordonnées. On obtient ainsi les axes de l'ellipsoïde (vecteurs propres de la matrice de covariance).

Pour déterminer les rayons R_i de l'ellipsoïde on utilise les deux informations suivantes :

- Volume = $\frac{4}{3}\pi R_1 R_2 R_3$ = taille du tableau
- le rapport entre les rayons est proche de la racine carré du rapport entre les valeurs propres de la covariance des données (calculées lors de l'AP)

2.2 Raffinement par la transformée de Hough

La transformée de Hough [3,5] va donc nous permettre de corriger l'erreur commise par l'ACP lors de la détermination de l'ellipsoïde.

2.2.1 Application de la transformée de Hough

L'objectif est toujours d'identifier les formes élémentaires modélisant notre ensemble de données. Cette méthode tente d'accumuler des évidences sur l'existence d'une forme particulière telle qu'une droite, un cercle ou une ellipse. Pour cela il faut tout d'abord déterminer les paramètres qui définissent entièrement notre forme. Ensuite on teste pour chaque valeur des paramètres la précision de l'ellipsoïde et on choisit le meilleur. Mais **comment évaluer la précision ?**

La transformée de Hough suggère de calculer le nombre de données dans notre ensemble se situant sur notre surface. Le meilleur est alors celui avec le plus grand score.

Cependant, pour choisir un volume, cette méthode ne fonctionne pas car il y a trop peu de points sur le bord, et si on compte le nombre de points à l'intérieur, l'algorithme nous pousserait à prendre un volume trop grand qui contiendrait tous les "outliers". On a finalement appliqué une variante de la méthode qui pénalise cet excès de volume.

Dans notre application, nous avons accès à la probabilité d'être dans le nodule grâce à la première partie. Nous avons alors utilisé ces probabilités pour "voter" pour ou contre un ellipsoïde. C'est ce que [?] appelle la transformée de Hough pondérée car le poids de chaque vote dépend de la fiabilité du pixel.

La **précision** se calcule alors ainsi : pour chaque voxel contenu dans l'ellipsoïde testé, on ajoute la probabilité du voxel d'être dans le nodule et on soustrait la valeur du seuil. Ainsi les voxels qui ne sont pas dans le nodule (probabilité inférieure au seuil) pénalisent l'efficacité de l'ellipsoïde.

2.2.2 Implémentation

Cette méthode permet d'obtenir la forme optimale, par contre elle est très coûteuse en temps calcul. En effet, pour chaque 12-uplet de paramètres il faut parcourir l'image entière. C'est pour cela que nous avons commencé par l'ACP, la valeur approchée nous permet de tester les paramètres sur un petit espace, typiquement plus ou moins 3 par rapport à la première approximation. Mais cela ne suffit pas, car 7^{12} ellipsoïdes reste beaucoup trop long à calculer. Il faut appliquer de nombreuses "ruses" pour accélérer la méthode.

Diminuer le nombre de paramètres à faire varier Nous savons que les axes de l'ellipsoïde sont orthogonaux. Donc après avoir fixé le premier axes, ils nous restent plus que deux coordonnées pour le deuxième axe, puis une pour le troisième. Ainsi nous nous ramenons à 9 paramètres à faire varier. Sur une image de 60x60 pixels on obtient un temps calcul d'environ 72h, c'est à dire complètement inutilisable.

Utilisation des matrices Avec le logiciel de programmation, le calcul matriciel est très optimisé. Pour chaque paramètre sur les axes, nous avons calculé une fois pour toute la matrice de 1 et 0 représentant l'ellipsoïde. En effet la variation du centre peut se faire directement sur la matrice et le calcul d'efficacité revient à une somme et produit sur les matrices. Ainsi nous avons réussi à passer à un calcul d'environ 2h30.

Diminuer la taille de l'image traitée Grâce l'approximation de l'ACP, on peut sélectionner une partie de l'image très serrée autour de l'ellipsoïde présumé et ainsi diminuer le temps calcul pour chaque jeu de paramètre. On arrive à environ 1h.

Finalement, cette méthode peut être extrêmement précise mais très coûteuse .

3 Coupes de graphes

Les méthodes de coupes de graphe permettent de résoudre des problèmes de classification binaire. Généralement coûteux en temps et en mémoire, ce type de méthode nous servira en fin d'algorithme pour affiner et régulariser la détection des nodules dans les images. Cette section présente brièvement la théorie des coupes de graphes et son application à notre projet. De nombreuses informations complémentaires concernant les coupes de graphes peuvent être trouvées dans [1, 2].

3.1 Théorie

On considère ici une image que l'on veut segmenter en deux parties. Le problème consiste donc à attribuer à chacun des points de l'image un label correspondant à l'une ou l'autre des classes de segmentation.

3.1.1 Formulation énergétique

Ce problème de classification peut s'écrire sous la forme d'une minimisation d'énergie dont la valeur dépend de l'affectation de chacun des points. Les points ont tendances à appartenir à l'une des classes lorsque d'une part leurs paramètres intrinsèques les y amènent (intensité, gradient local, ...) mais également lorsque leurs voisins y appartiennent (connexité des parties à segmenter). Cette énergie est alors constituée de 2 termes :

- **Terme d'attache aux données** : fonction d'un point et de ses paramètres pour favoriser son attache à l'un ou l'autre des labels de classification.
- **Terme de régularisation** : fonction de plusieurs points jugés *voisins* (c'est à dire que leurs affectations respectives dépendent l'une de l'autre).

L'énergie s'écrit alors sous la forme :

$$E_{TOTAL} = \sum_{p \in \mathcal{G}} E_{DATA}(\text{label}(p), p) + \lambda \sum_{p, q \in \mathcal{G} \text{ voisins}} E_{SMOOTH}(p, q)$$

où λ est appelé terme de régularité et permet de modifier l'importance du rapport attache-aux-données/régularité.

3.1.2 Forme des graphes

Il est également possible de modéliser le problème sous la forme d'un graphe \mathcal{G} . Ce graphe contient l'ensemble des points à classifier représentés sous la forme de ses sommets auxquels on ajoute 2 sommets particuliers généralement notés s (pour *source*) et t (pour *target*). Les points sont tous reliés par des arêtes à s et t qui représentent les 2 *labels* de classification du problème. Par ailleurs, on relie les points entre eux lorsque l'affectation de l'un est fonction de l'affectation de l'autre.

On obtient alors un graphe de la forme illustrée en figure 1.

3.1.3 Coupes minimales

On dit que \mathcal{C} est une **coupe** du graphe \mathcal{G} si elle sépare \mathcal{G} en deux ensembles connexes disjoints \mathcal{S}, \mathcal{T} tels que $s \in \mathcal{S}$ et $t \in \mathcal{T}$.

On peut alors définir la **capacité** de la coupe \mathcal{C} par :

$$c(\mathcal{C}) = \sum_{(u,v) \in \mathcal{S} \times \mathcal{T}} c_{uv}$$

où c_{uv} est le poids assigné à l'arête $[u, v]$.

Une coupe de \mathcal{G} est alors minimale lorsqu'il n'existe pas d'autres coupes dont la capacité est inférieure strictement. Cependant, il n'y a en général pas unicité de la coupe minimale. De nombreux algorithmes permettent de calculer la coupe minimale d'un graphe via le calcul de son équivalent, le **flot maximum**. En effet, le flot maximum est facilement calculable avec des algorithmes gloutons par exemple.

On cherche donc à se ramener au problème de calcul de coupe minimale en affectant aux arêtes les poids suivants :

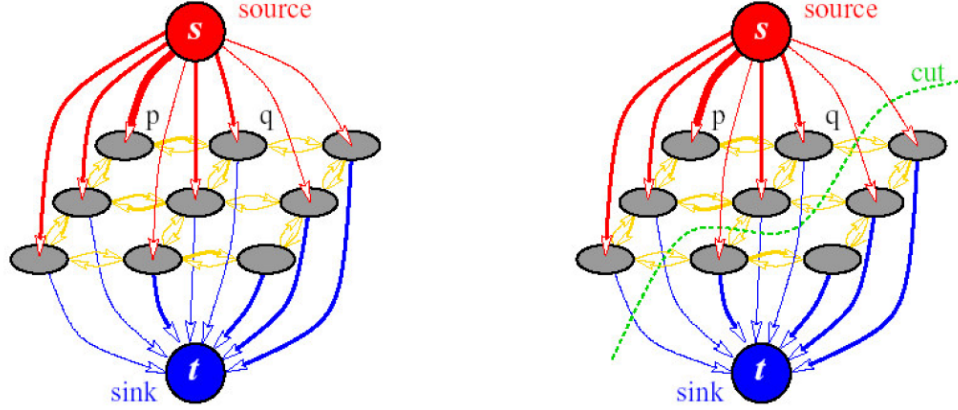


FIGURE 1 – Forme classique d’un graphe utilisable dans les méthodes de coupes de graphe. A gauche, le graphe avec les 2 noeuds principaux s et t . A droite, un exemple de coupe de graphe. Les arêtes en gras relient les sommets aux labels définis par la coupe.

- Un terme d’attache aux données, lorsque l’arête relie un sommet quelconque à t ou s : $E_{DATA}(\text{label}, p)$
- Un terme de régularité, lorsque l’arête relie deux sommets voisins p et q : $E_{SMOOTH}(p, q)$

On rend alors équivalents la minimisation de l’énergie E_{TOTAL} et la recherche de la coupe minimale de \mathcal{G} . Le calcul de la coupe minimale du graphe \mathcal{G} ainsi construit permet de minimiser l’énergie et donc de résoudre le problème de classification.

3.2 Application dans le cadre de la détection des nodules

Pour appliquer des méthodes de coupes de graph au sujet qui nous intéresse, il s’agit alors de définir les fonctions d’attaches aux données et de régularisation d’une part et de construire un graphe \mathcal{G} représentatif du problème d’autre part.

3.2.1 Construction du graphe

A chaque voxel de l’image on associe un sommet p du graphe. On ajoute également s_{IN} pour le label *interieur* et s_{OUT} pour le label *exterieur*. Les sommets étant définis, il s’agit alors de les relier entre eux avec des arêtes correspondants à la réalité du problème.

On relie donc chaque sommets p à s_{IN} et s_{OUT} . En remarquant par ailleurs que la complexité de la minimalisation de la coupe dépend du nombre d’arêtes dans le graphe, on se contente d’appliquer un modèle de 6-connexité pour les arêtes reliant 2 sommets p voisins. C’est à dire que l’affectation chaque voxel ne dépend que de l’affectation de ses 6 voisins les plus proches (2 selon chacun des 3 axes).

3.2.2 Energies

L’énergie d’attache aux données se calcule à partir des probabilités définies dans la première partie. Celles-ci permettent d’avoir une information grossière sur la forme de l’objet à détecter. On suppose que cette probabilité est relativement proche de la réalité et on définit ainsi l’énergie d’attache aux données par :

- $E_{DATA}(s_{IN}, p) = f(\mathbb{P}(p \in s_{IN}))$
- $E_{DATA}(s_{OUT}, p) = f(\mathbb{P}(p \in s_{OUT}))$

f peut être la fonction *valeur absolue* ou *carrée* par exemple.

L’énergie de régularisation est nulle lorsque les labels des 2 voxels voisins sont les mêmes. Dans le cas contraire, elle dépend de deux paramètres :

- la différence d'intensité entre les 2 voxels doit être suffisamment importante. En effet, 2 voxels voisins d'intensité similaire appartiennent certainement au même objet. La réciproque n'est par contre pas vérifiée.
- la localisation de leur barycentre. En effet, celui-ci est sensé appartenir à la frontière séparant les 2 objets.

En suivant les conseils de [4], nous avons utilisé les énergies suivantes :

- pour la différence d'intensité : $E_{SMOOTH1} \propto e^{-\frac{\Delta I^2}{2\sigma^2}}$ où ΔI est l'écart d'intensité et σ est le paramètre de pénalisation.
- pour la localisation du barycentre : $E_{SMOOTH2} \propto \frac{d^2}{\sigma^2}$ où d est la distance du barycentre à la frontière.

Les fonctions utilisées peuvent être quelconques à partir du moment où elles vérifient la condition de **régularité** :

$$\forall i, j \in \{VOXEL\}, \forall l_1, l_2 \in \{LABEL\} \quad E(i \rightarrow l_1, j \rightarrow l_1) + E(i \rightarrow l_2, j \rightarrow l_2) \leq E(i \rightarrow l_1, j \rightarrow l_2) + E(i \rightarrow l_2, j \rightarrow l_1)$$

Cette condition est évidemment vérifiée dans le cas précédent puisque l'égalité des labels entraîne la nullité du terme de régularité et que les autres termes sont positifs ou nuls.

On obtient finalement une énergie de la forme :

$$E_{TOTAL} = \sum_{p \in \mathcal{G}} (1 - \lambda) \mathbb{P}(p \in \text{label}(p))^2 + \lambda \sum_{p, q \in \mathcal{G} \text{ voisins}} \delta_{\text{label}(p)=\text{label}(q)} \left[\lambda \frac{d\left(\frac{p+q}{2}, \epsilon\right)^2}{\sigma_d} + (1 - \lambda) e^{-\frac{\Delta I^2}{2\sigma_c^2}} \right]$$

où δ est le symbole de Kronecker, $d\left(\frac{p+q}{2}, \epsilon\right)$ est la distance du barycentre de p et q à la frontière de l'ellipsoïde ϵ et ΔI est la différence d'intensité entre p et q . Les valeurs de σ_d et σ_c ont été fixées à 5 pixels et 3 d'intensité. Elles sont cependant changeables et une étude plus approfondies, réalisée à l'aide de données sur les variations d'intensité des images médicales et la taille moyenne des nodules permettrait de leur affecter une valeur plus raisonnable.

4 Tests de segmentation

4.1 Test sur la détermination de l'ellipsoïde

La méthode est robuste au bruit diffus, on a pu la tester sur des images d'ellipsoïdes bruités générés par la fonction `generate_ellipsoid`. Même avec un bruit important on obtient de bons résultats pour le centre et les directions des axes. Cependant, les rayons sont assez sensibles au bruit puisque ils dépendent directement du nombre de "faux positifs" et que leur détermination dépend d'une égalité approximative des rapports. Il serait intéressant de chercher une manière de diminuer l'influence de ces "faux positifs" sur le calcul des rayons.

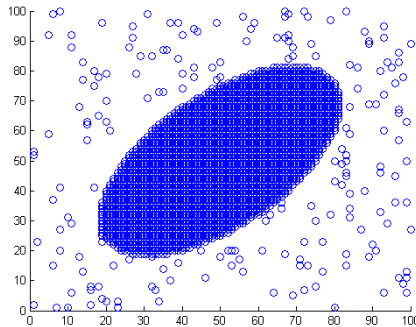


FIGURE 2 – Ellipsoïde bruité généré pour la phase de test

Cet ellipsoïde a été créé à partir des paramètres suivants :

$c = [50 \ 50 \ 30]$; Le centre de l'ellipsoïde
 $X1 = [1 \ 1 \ 0]$; Les axes
 $X2 = [-1 \ 1 \ 0]$;
 $X3 = [0 \ 0 \ 1]$;
 $R = [40 \ 20 \ 5]$; Les rayons

L'ACP renvoie les résultats suivants :

$c = [50.06 \ 50.02 \ 30.29]$;
 $X1 = [-0.7 \ -0.7 \ 0]$;
 $X2 = [-0.7 \ 0.7 \ 0]$;
 $X3 = [0 \ 0.01 \ 1]$;
 $R = [30.79 \ 16.12 \ 8.07]$;

On constate que le centre et les axes sont déjà très précis, mais ce n'est pas le cas des rayons. La transformation de hough va pouvoir l'améliorer :

$R = [35.79 \ 16.12 \ 8.07]$;

Sur le nodule, le bruit est aussi assez diffus mais le problème principal réside dans la prise en compte de la membrane. En effet, s'il reste une zone continue de membrane non filtrée, l'ACP donne des résultats très médiocres pour les 12 paramètres.

Finalement, tant que le bruit reste diffus, l'application de la transformée de Hough est suffisante pour corriger l'erreur de cette étape.

4.2 Test sur la coupe de graphe

Pour vérifier l'efficacité de la partie segmentation, nous avons utilisé des données théoriques :

- un ellipsoïde défini par avance par son centre, ses axes et ses rayons
- une image représentant un ellipsoïde d'une certaine couleur sur un fond d'une autre couleur
- la loi de probabilité correspondant à la segmentation effectuée dans la première partie

L'image et la loi de répartition ont été bruitées plus ou moins fortement pour tester la robustesse de la segmentation face à des données pas totalement fiables.

Les résultats suivants sont présentés sous forme de tableaux et indiquent le pourcentage de voxels mal segmentés en fonction du paramètre de régularité λ choisi.

λ	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
ϵ (%)	0	0	0	0	0	0	0	0	0	0.7	∞

TABLE 1 – Résultats en l'absence de bruit.

λ	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
ϵ (%)	0	0	0	0	0	0	0	0	0	0.7	∞

TABLE 2 – Résultats en bruitant la loi de répartition uniquement.

λ	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
ϵ (%)	0.8	0.8	0.8	0.8	0.8	0.7	1.1	2.7	10.3	18.5	∞

TABLE 3 – Résultats en bruitant l'image à segmenter uniquement.

Les résultats montrent ainsi une tendance à choisir un coefficient de régularité proche de 0.5 évitant ainsi les cas extrêmes où aucune régularité/trop de régularité sont appliquées.

λ	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
ϵ (%)	∞	13.6	9.6	8	7.1	6.5	6.6	7.5	12.8	19.6	∞

TABLE 4 – Résultats en bruitant l’image à segmenter et la loi de répartition.

4.3 Tests pratiques

La comparaison entre la segmentation réalisée par l’expert et la segmentation réalisée par notre algorithme est très décevante. En effet, même sans avoir testé l’algorithme sur l’ensemble de la base de donnée, les résultats sont déjà catastrophiques, atteignant même jusqu’à 70%. Ces résultats peuvent s’expliquer entre autre par le fait que le nodule noté par l’expert diffère nettement d’un ellipsoïde : En conséquence, l’algo-

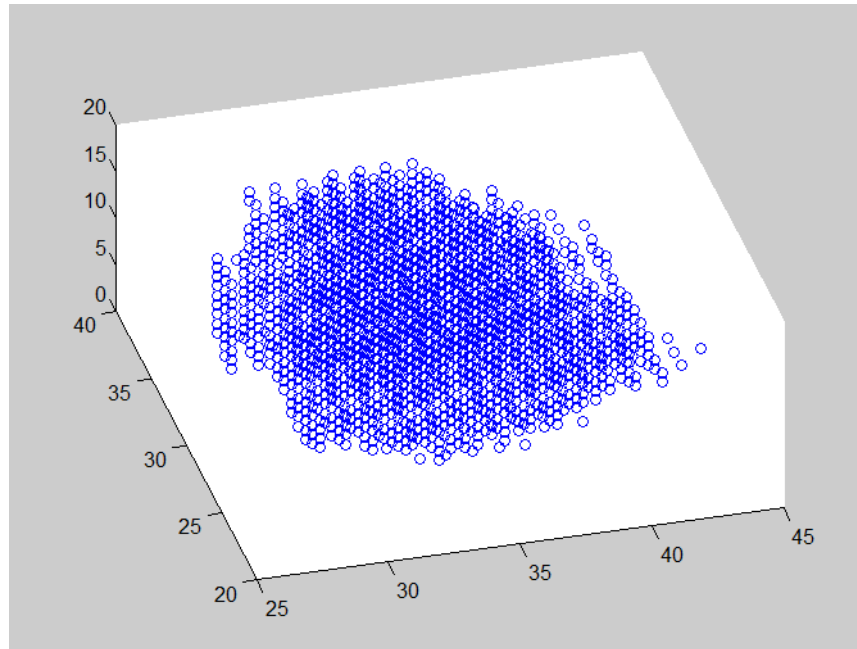


FIGURE 3 – Nodule délimité par l’expert

ritme de graph cut devrait autoriser une plus grande liberté à la forme de la segmentation (au risque d’enlever/ajouter de mauvais voxels). Pour cela, il faudrait alors changer les divers paramètres entrant en compte dans la partie graph cut. Cependant, leur nombre élevé rend l’optimisation de ces paramètres longue et laborieuse et n’a donc pas pu être réalisé pour notre projet.

Conclusion

Finalement, chaque étape nous permet d’affiner la segmentation. L’application des filtres permet de segmenter de manière probabiliste les voxels de l’image, la détermination de l’ellipsoïde utilise les caractéristiques de géométries du nodule pour faire une segmentation globale, et la coupe de graphe permet de s’adapter au caractère granuleux de la surface.

Cependant, cette détermination repose sur l’hypothèse que le nodule est de forme globalement ellipsoïdale ce qui n’est pas toujours le cas d’après les segmentations manuelles réalisées par le praticien. A ce moment l’algorithme n’est plus très efficace.

Cette détection automatique serait d’une très grande aide pour les praticiens. En effet, lors de la mise en place de ce programme, nous avons pu constater la difficulté de tirer des informations globales sur une image 3D et jauger de la difficulté et de la lenteur d’une détermination manuelle. Le programme a été testé

et adapté aux cancers du poumon mais pourrait être largement réutilisé pour d'autres parties du corps, tant que le nodule se situe à l'intérieur d'une partie molle, c'est à dire bien différentiable du nodule lui-même par scanner.

Références

- [1] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern.*
- [2] Y. Boykov and O. Veksler. Graph cuts in vision and graphics : Theories and applications.
- [3] R.O. Duda and P.E. Hart. Use of the hough transformation to detect lines and curves in pictures. Technical Report 36, AI Center, SRI International, 333 Ravenswood Ave, Menlo Park, CA 94025, Apr 1971. SRI Project 8259 Comm. ACM, Vol 15, No. 1.
- [4] D. Freedman and T. Zhang. Interactive graph cut based segmentation with shape priors. *CVPR*, 2005.
- [5] P. V. C. Hough. Machine Analysis of Bubble Chamber Pictures. In *International Conference on High Energy Accelerators and Instrumentation*, CERN, 1959.
- [6] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(6) :559–572, 1901.