



RAPPORT DE STAGE

Master « Sciences et Numérique pour la Santé »

Délivré par l'**Université de Montpellier**

Préparée au sein du **Laboratoire d'Informatique,
de Robotique et de Microélectronique de
Montpellier (LIRMM)**

Spécialité : **Ingénierie des Dispositifs pour la
Santé**

Présentée par **ABEZA David**

**AUTOMATISATION DE LA
SEGMENTATION DES OS DE LA
BASE DU CRANE**



Remerciements

Je remercie mon responsable de master, **Emmanuel LE CLEZIO**, ainsi que toute l'équipe pédagogique qui a intervenu lors de ma formation, pour m'avoir apporté le savoir théorique nécessaire au stage.

Je remercie l'**équipe ICAR** du LIRMM, pour son accueil ainsi que son ambiance, durant les deux mois du stage.

Je remercie la succession de personne, **Olivier STRAUSS** et **Florentin KUCHARCZAK**, m'ayant permis de rentrer en contact avec **Valentin FAVIER**.

Je remercie **Valentin FAVIER** et **Gérard SUBSOL** d'avoir accepté de me faire confiance et me prendre comme stagiaire. Je remercie pour leurs encadrement, conseils et écoute. Je les remercie aussi pour le professionnalisme et la rigueur qu'ils ont permis de m'améliorer.

Enfin, je les remercie pour les conseils d'organisation et la relecture qui fut faite de ce rapport.

Je vous remercie à vous lecteurs et évaluateurs pour votre temps accordé et vous souhaite une bonne lecture.

Table des matières

Page de garde	1
Remerciements	2
Table des matières	3
Table des figures	4
Liste des tableaux	5
Introduction	5
Outils & définitions	6
Outils	6
Définitions	6
Etude Bibliographique	8
Présentation du lieu du stage	8
Contexte, enjeux et problématique	10
Travail réalisé	11
Evaluation du travail déjà fait	11
Fonctionnement et principe du programme	11
Fonctions implantées pour évaluer le programme	13
Evaluations des résultats et mise en évidence des problèmes	15
Amélioration du programme	16
Idées de solutions	16
Comparaison nouvelle / ancienne structure	18
Nouveaux résultats	19
Conclusion et perspectives	22
Annexes	23
Annexe 1 : Tableaux	23
Annexe 2 : Les 1ers résultats	25
Bibliographie	31

Table des figures

Figure 1 : Localisation des divers types de sinus. Source : Wikipédia, sinus paranasaux.	7
Figure 2 : 3 slices (de gauche à droite : slice 26, 58 et 87) d'image de scanner.	7
Figure 3: Mise en évidence des zones d'intérêt principal et des zones posant un souci lors de calcul (encadrée en rouge).	10
Figure 4 : Représentation du Blocs de voxel voisin au voxel observé.	12
Figure 5 : Slice 87 des images pour visualiser le fonctionnement du programme. Conditions : méthode Moyenne, Bloc 3x3x3, $\sigma_{\min}=150$, Os relatif=100, Os Absolu=400, Air Absolu = -700, % os= 19%	13
Figure 6 : Terminal de commande du programme d'évaluation du bloc.	14
Figure 7 : Synthétisation du fonctionnement du programme d'Otsu.	17
Figure 8 : Comparaison entre l'ancien et le nouveau programme.	18
Figure 9 : Mise en évidence de la calcification des vaisseaux sanguins.	19
Figure 10 : Comparaison avec et sans écart-type. Conditions ; air absolu : -700, os absolu : 400, écart type : 150, méthode : moyenne, os relatif : 100, pourcentage 5%	20
Figure 11 : Résultats finaux	21
Annexe 2.1: Coupe 1 de l'image de test	25
Annexe 2.2: Coupe 11 de l'image de test	26
Annexe 2.3: Coupe 20 de l'image de test	27
Annexe 2.4: Coupe 28 de l'image de test	28
Annexe 2.5: Coupe 32 de l'image de test	29
Annexe 2.6: Coupe 42 de l'image de test	30

Liste des tableaux

Tableau 1 : Evaluation du temps de calcul pour un Bloc de 5x5x3 et 11x11x9.	23
Tableau 2 : Utilisations des compteurs de comparaisons	24

Introduction

Il est important de comprendre la morphologie 3D des sinus aériens de la face, ainsi que leurs évolutions pour les traitements médicaux et interventions chirurgicales. La connaissance précise de la zone dans laquelle doit intervenir un chirurgien est indispensable au bon déroulement d'une opération. Cela est d'autant plus vrai pour la chirurgie endoscopique.

L'imagerie médicale permet de visualiser, de manière non-invasive, les structures anatomiques du corps. L'imagerie par tomographie assistée par ordinateur est particulièrement utile pour la visualisations 2D et 3D des os; elle consiste à mesurer la densité d'absorptions des rayons X. Cependant, la structure des sinus, rend difficile l'évaluation et la segmentation de ceux-ci.

Dans la bibliographie différents travaux sont réalisés sur la segmentation des sinus. Certains utilisent le seuillage local. C'est partir de cette méthode que mon travail a démarré.

Un premier travail avait déjà été fait avant mon arrivé. Ma première tâche fut de l'évaluer.

J'ai ensuite développé de nouvelles fonctions et programmes.

Pour enfin proposer des améliorations pour résoudre les problèmes engendrés par le premier travail.

Outils & définitions

Outils

Langage C++

Durant mon stage j'ai été amené à utiliser le langage de programmation C++. Il permet de programmer des procédures, des orientations objets ou générique... Il a pour avantage d'être très performant, son assimilation avec le C et la grande variété de bibliothèque existantes, font de celui-ci un des plus utilisés.

Bibliothèque CImg

J'ai principalement utilisé la bibliothèque CImg durant le stage. Elle est en utilisation libre, ne demande que les bibliothèques standards du C. Le fait qu'elle soit contenue entièrement dans un fichier type .h (header) la rend simple à implanter et à transmettre. Cette bibliothèque fournit des "class" et fonction simplifiant le chargement, la sauvegarde, le traitement et l'affichage d'une image. Elle est capable d'accepter tout type d'image (de 1D à 4D).

Compilateur Xcode

Du fait de travailler sur un Mac j'ai utilisé l'environnement et le compilateur propre à celui-ci. Il contient une grande variété de logiciels (graphique, audio...). Cependant, je n'avais pas l'utilité de toutes ses fonctions, donc, j'ai juste utilisé comme compilateur C/ C++. Utiliser des fonctions propres à Mac rendrait aussi la transmission du programme à mes encadrants plus complexe que nécessaire.

Fiji

Pour pouvoir visualiser, comparer et analyser, j'ai aussi utilisé le logiciel Fiji. Il est possible de faire du traitement d'images avec, mais je ne l'ai utilisé que pour visualiser et synchroniser les diverses images (traitées, originales, seuils...)

Images 3D

Comme support de travail j'ai eu des images 3D, en niveau de gris et traitées à la main par un expert, provenant de scanner rayon X de patients et d'une personne décédée. Ces images étaient sous le format ANALYZE 7.5. Il consiste en une paire de fichiers fonctionnant ensemble (.hdr et .img). Cette paire doit absolument porter le même nom.

Une image 3D correspond à une superposition de couches d'une certaine épaisseur.

Définitions

Voxel

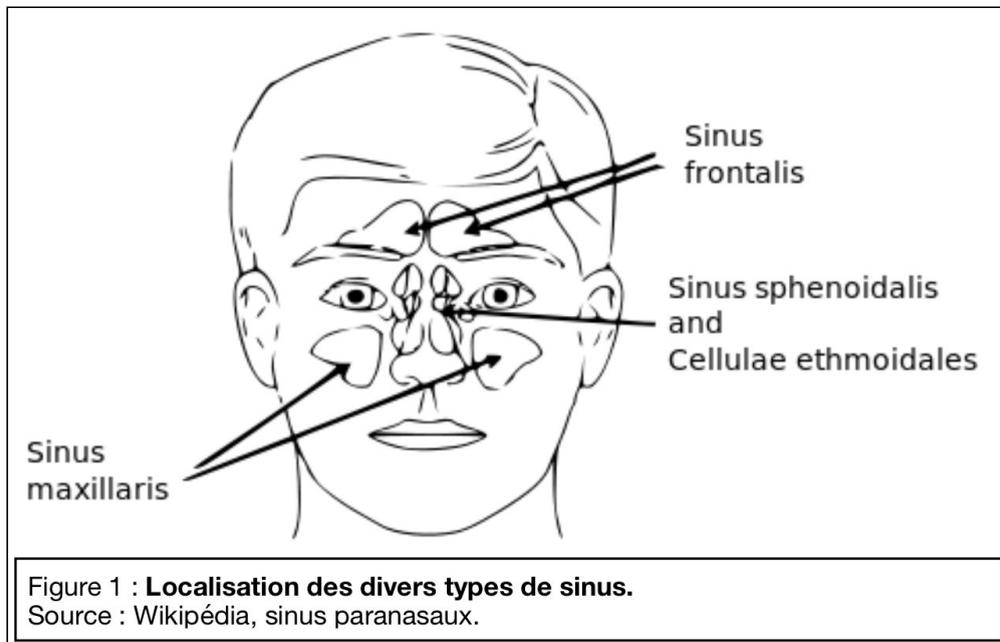
Un voxel (volume element) est un pixel (picture element) ayant une profondeur. Le voxel peut être représenté sous la forme d'un cube, ou d'un pavé selon la taille de celui-ci.

Les sinus paranasaux

Les sinus paranasaux sont des espaces dans les os du crâne remplis d'air. Ils ont pour particularité d'avoir l'air qui entre et sort par le même endroit. Il existe quatre groupes de sinus : maxillaires (sous les orbites), frontaux (au-dessus des yeux et du nez), ethmoïdaux (dans l'os ethmoïde) et sphénoïdaux (dans l'os sphénoïde).

Les sinus ethmoïdaux et maxillaires sont présent dès la naissance et poursuivent leur développement à la croissance.

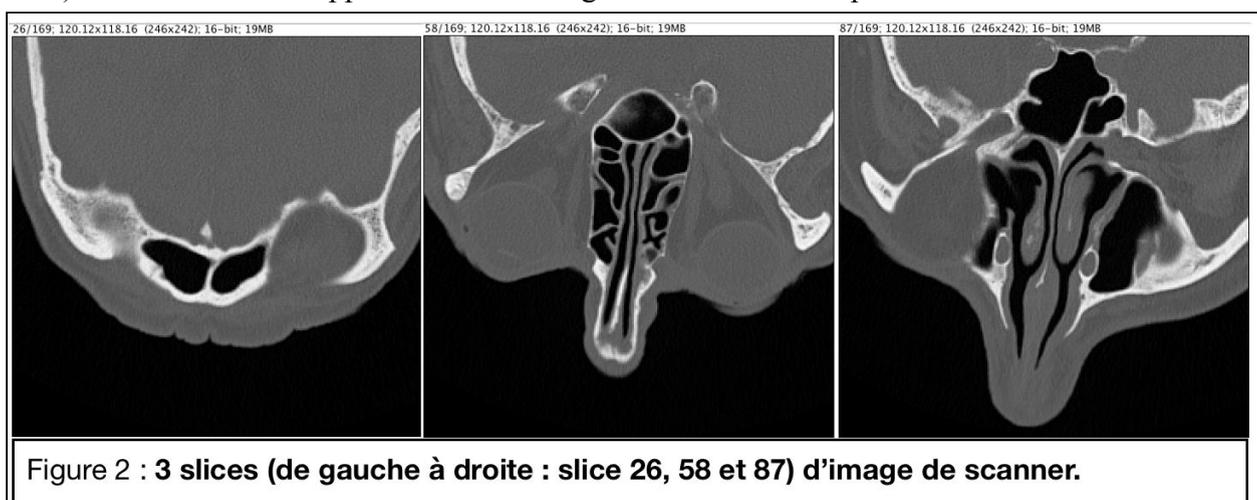
Les sinus frontaux et sphénoïdaux apparaissent durant l'enfance et l'adolescence.



Images de scanner à rayon X (CT-Scanner)

Les images de scanner sont des image 2D ou 3D en niveaux de gris. Ces niveaux de gris correspondent à l'absorption des rayons X par les tissus. Plus l'absorption est forte et plus le voxel équivalent en sortie sera blanc. Il est aussi possible d'améliorer le contraste de tissus en injectant des produits dans ceux-ci (particulièrement efficace dans les vaisseaux sanguins).

Il est possible, avec l'image 3D, de visualiser selon les trois plans (transversal, sagittal et frontal). Dans la suite du rapport toutes les images seront suivant le plan transversal.



Etude Bibliographique

La tomодensitométrie est une technique permettant de visualiser des structures anatomiques sur des images 2D ou 3D. Ces images, en niveau de gris, viennent d'un traitement informatique de mesures d'absorptions des rayons X par les tissus. Cette technologie permet aux médecins d'appréhender la morphologie des organes sans être invasif.

Cependant l'identification automatique, et informatique, de ces structures reste difficile. De plus, la structure des sinus est d'une grande complexité et variabilité.

Articles publiés :

[1] L'utilisation d'opérations de morphologie automatique avec un seuil, permet de définir une zone d'intérêt ainsi qu'un contour initial. Cette zone est ainsi superposée sur l'original et met en évidence les os des sinus et les cavités.

[2] A partir des caractéristiques de densité des sinus, les autres obtiennent des caractéristiques géométriques. Ces caractéristiques géométriques sont ensuite observées et comparées afin de sortir les particularités des orifices de drainage des sinus paranasaux dans les fosses nasales.

Chacune des méthodes ci-dessus a une approche différente sur le sujet, mais aussi valide l'une que l'autre. L'une recherche des zones d'intérêts tandis que l'autre des caractéristiques d'intensités et géométriques des sinus examinés.

Présentation du lieu du stage

La structure dans laquelle j'ai réalisé mon stage est le Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier (LIRMM) situé sur le campus de Saint-Priest de l'Université de Montpellier. C'est une unité de recherche mixte, qui dépend à la fois de l'Université de Montpellier et du Centre National de la Recherche Scientifique.

Les recherches menées au LIRMM sont au coeur des sciences et technologies de l'information, de la communication et des systèmes, de la technologie à l'humain et aux usages.

Les recherches s'orientent autour des domaines propre à celui-ci, l'informatique, l'électronique et l'automatique, ainsi que divers autres tel-que, la biologie, la chimie, les télécommunications, la santé...

Le LIRMM est divisé en trois départements, informatique, microélectronique et robotique, qui sont eux aussi divisés en plusieurs équipes.

J'ai travaillé au sein de l'équipe ICAR (Images & Interactions). Elle est rattachée au département de robotique. Elle travaille à l'interaction et le traitement de données visuelles telles que les images, les vidéos et les objets 3D.

Durant mon stage, j'ai été encadré par M. Gérard SUBSOL, responsable adjoint et un permanent de l'équipe ICAR, et M. Valentin FAVIER, doctorants de l'équipe ICAR et chirurgien ORL.

M. SUBSOL travaille sur la modélisations 3D avec des applications en paléo-anthropologie et en médecine. Il encadre aussi M. Favier dans sa thèse.

M. FAVIER travaille sur l'élaboration de modèle 3D personnalisé et imprimable pour la simulation et la planification de chirurgies de la base du crâne par voie endonasale.

C'est d'ailleurs sur un point en lien direct avec cette thèse que se déroule mon stage.

Contexte, enjeux et problématique

Comme dit précédemment mon stage est directement rattaché à la thèse de Valentin FAVIER. A partir d'images provenant d'un scanner, un traitement automatique est mis en place afin d'identifier tous les os. Une fois identifiés, ces os sont implantés couche par couche sur un logiciel de visualisation 3D. Enfin, un modèle personnalisé est imprimé en 3D.

Il faut que le modèle, une fois imprimé, soit le plus proche possible de l'original. L'objectif de tout cela est la chirurgie endoscopique de la base du crâne. Avec des modèles personnalisés, il sera possible d'améliorer la planification pré-opératoire, d'entraîner les chirurgiens, d'apporter un support physique supplémentaire pour l'apprentissage de l'anatomie, tester des dispositifs médicaux sur un modèle fiable, tester des nouvelles techniques opératoires...

Valentin Favier a fait des segmentations à la main. Cependant, une dizaine d'heures lui furent nécessaire pour faire celles-ci.

Mon travail consistait donc principalement, à tester, évaluer et mettre en place des méthodes de traitement automatique.

Le principal problème est la grande variété de niveau de gris. Les images sur lesquelles je travaille ont plus de 4 000 valeurs possibles et pouvaient aller d'une valeur de -1024 (correspond à de l'air) à plus de 3 000 (os très dense). Il faut ajouter à tout cela l'artefact (ou effet) de volume partiel et le bruit qui, pour une même structure, font varier la densité lors de la mesure.

L'artefact de volume partiel est lié à la taille de l'objet visualisé et la à taille du(des) voxel(s) le(s) composant(s). Il faut comprendre que le voxel est trop grand et peut contenir plusieurs type de tissu. Cependant un voxel ne peut avoir qu'un seul niveau de gris. Ce niveau de gris correspond à la moyenne d'absorption des éléments composant le voxel. De plus, la présence des muqueuses et tissus mous autour des os, couplée avec l'effet de volume partiel, entraîne une moyenne au niveau des voxels qui rend difficile la distinction entre les os fin et la muqueuse.

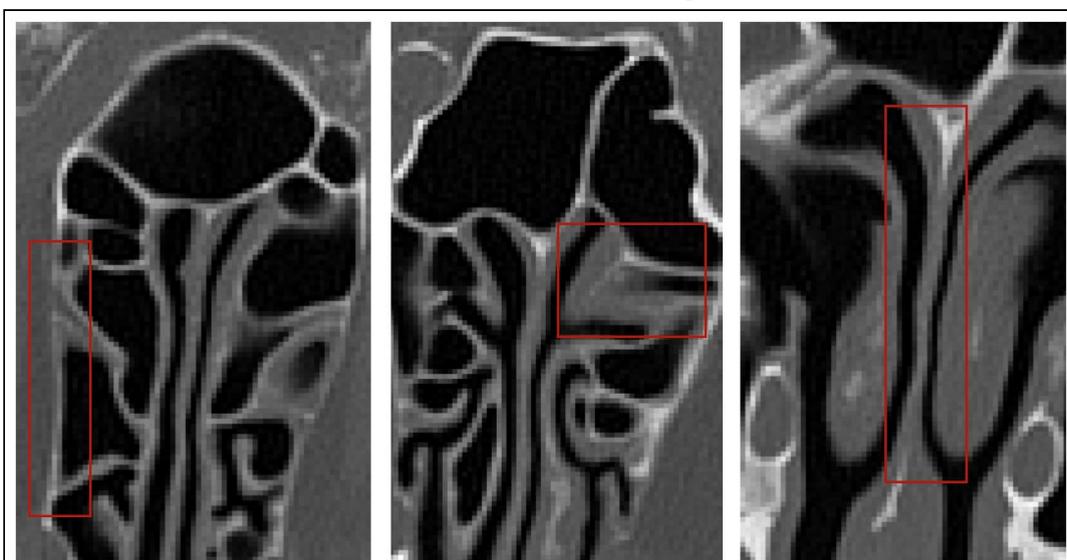


Figure 3: Mise en évidence des zones d'intérêt principal et des zones posant un souci lors de calcul (encadrée en rouge).

Travail réalisé

1. Evaluation du travail déjà fait

1.1. Fonctionnement et principe du programme

Organisation du programme

Avec la bibliothèque CImg, l'image à traiter est utilisée comme un tableau en 3D. Une autre image est créée en début de programme. Ce sera l'image segmentée, ou image binaire, en blanc (1) sera de l'os et en noir (0) tout ce qui n'est pas de l'os. Cette image lors de sa création est initialisée à zéro (noir).

Trois boucles "for" (dans l'ordre : profondeur, hauteur et largeur) sont utilisées pour naviguer dans l'image et être sûr de prendre en compte tous les voxels.

Avant de calculer le seuil local le programme vérifie si la valeur maximum du bloc est supérieure à un seuil d'os relatif fixé par l'utilisateur. Cela permet d'éviter le calcul du seuil dans un bloc de tissu mou ou d'air. On appelle cette étape : détection de présence osseuse.

Après cette vérification, le calcul du seuil local est fait, puis il est comparé au voxel. Si le seuil est inférieur au voxel alors on considère que c'est de l'os, donc le voxel correspondant sur l'image segmentée est fixée 1. Sinon cela ne nous intéresse pas.

Méthodes de calcul

De base, quatre méthodes de calcul de seuil ont été implantées.

La moyenne : établit un seuil en faisant le rapport de la somme des valeurs sur le nombre de valeurs.

La médiane : établit un seuil en prenant la valeur au centre des valeurs triées par ordre croissant.

La moyenne harmonique : établit un seuil avec la moyenne de l'inverse des valeurs, elle est utilisée lorsqu'il existe des liens de proportionnalité inverses.

La méthode d'otsu : établit un seuil avec l'histogramme. Les valeurs vont être réparties en deux ensembles de voxels (premier plan et second plan). Elle calcule ensuite le seuil optimal séparant ces deux ensembles. Ce calcul recherche la variance à l'intérieur de chaque ensemble minimal. C'est l'indice séparant les deux qui est gardé comme seuil.

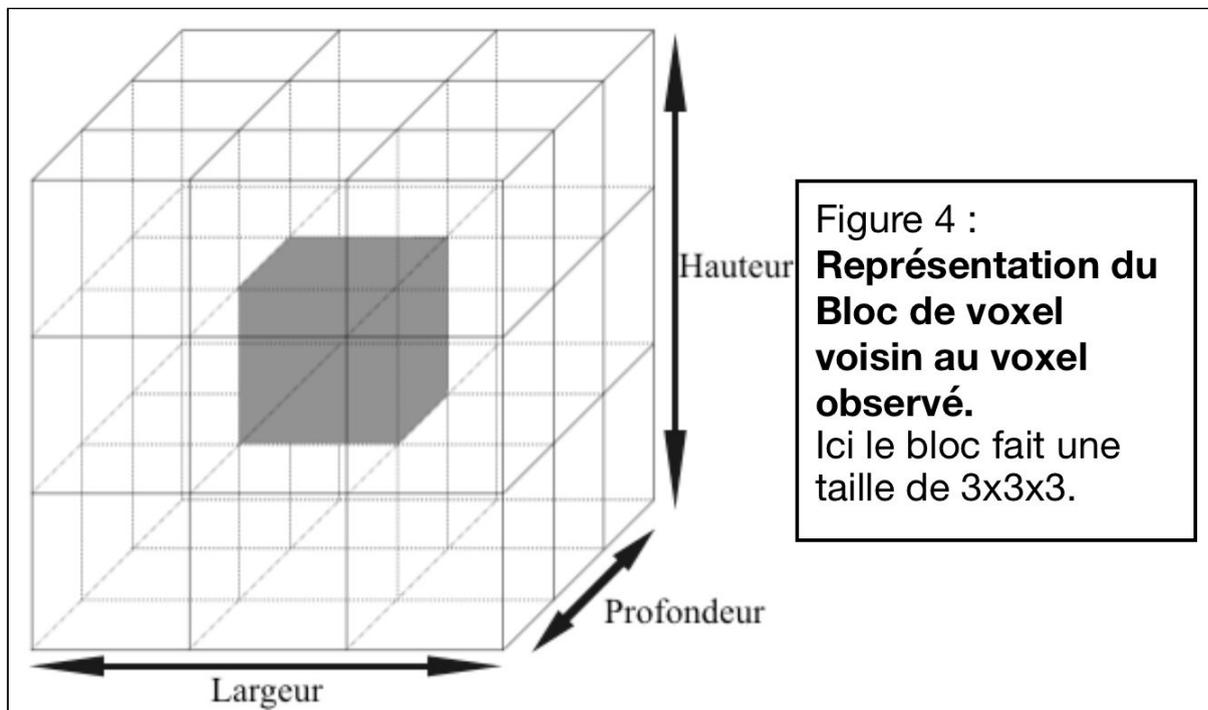
Bloc glissant

Afin de calculer le seuil local, un bloc de voxels voisin à celui observé est créé. Comme l'image celui-ci est en 3D, mais doit respecter la condition d'avoir une taille de côté impaire, afin que le voxel observé soit toujours au centre.

Par exemple, si un bloc de 3x3x3 est choisi, tous les voxels à 1 de distance (haut, bas, droite, gauche, diagonal) constitueront celui-ci. Pour une taille de 5x5x5, ce sera les voxels à 2 de distance...

Afin d'éviter tout problème sur le bord de l'image, les voxels théoriques, en dehors de celle-ci ne sont pas pris. Par exemple un bloc de 3x3x3 dans un coin ne fera qu'une taille de 4.

Une fois les valeurs des voxels constituant ce bloc récupéré, le seuil peut être calculé.



1.2. Fonctions implantées pour évaluer le programme

Afin de faciliter l'évaluation des méthodes actuelles et permettre d'évaluer les futures méthodes, j'ai développé plusieurs fonctions et programmes.

Comparaison des images

Travaillant à la fois sur l'originale et la segmentée, par la main d'un expert (Valentin Favier), j'ai mis en place un programme de comparaison simple. Ce programme prend deux images en entrée (la témoins et celle segmentée par programme).

Deux vérifications sont faites, la taille des deux images est la même et l'image segmentée n'est pas constituée que de blanc ou noir (car dans ce cas là $\min=\max$ et les comparaisons ne fonctionnent plus)

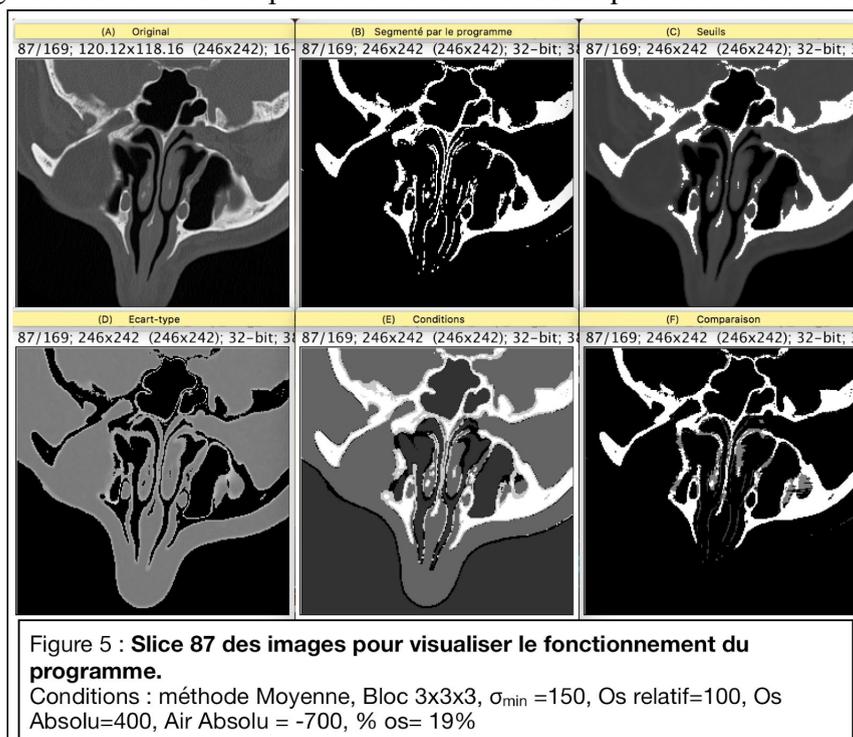
Comme pour le programme principal, ce programme navigue dans les images avec trois boucle "for".

En sortie de ce programme il y a une image de la comparaison. Sur cette image quatre niveaux de gris sont visibles : blanc (os sur les deux images), noir (rien sur les deux images), gris clair (os sur l'image témoin et pas d'os sur la deuxième images) et gris foncé (os sur la deuxième imag et pas d'os sur la témoin).

Visualisation du fonctionnement

Pouvant être intéressant à observer, j'ai ajouté une troisième image, l'image des seuils. Chaque seuil calculé est ainsi sauvegardé.

Plus tard j'ai ajouté une quatrième et cinquième image : image des conditions et image des écart-type. L'image des écart-type a exactement le même fonctionnement que l'image des seuils tandis que l'image des conditions me permettait d'évaluer sous quelle condition évalué le voxel.



Sauvegarde de valeurs et compteurs

J'ai utilisé les compteurs à deux endroits, lors de la segmentation afin de m'aider à résoudre des problèmes et visualiser si certaines conditions fonctionnaient ...

J'ai cependant principalement utilisé les compteurs lors de la comparaison entre deux images. Cela m'a permis d'une part d'évaluer rapidement les images, d'autre part l'efficacité des divers méthodes et seuils.

Même si je pouvais visualiser un grand nombre d'image en même temps avec Fiji j'étais confronté à deux problèmes; la taille de mon écran m'empêchant de visualiser plus de dix image en même temps, le fait que je n'avais que mes impressions pour évaluer.

En récupérant le nombre de chaque cas, j'ai pu mettre en place un tableau d'évaluation rapide (Tableau Annexe 1). Avec la somme, je pouvais vérifier que tous les voxels étaient bien calculés, évaluer à travers un pourcentage les voxels faux, la sensibilité, la spécificité et l'évolution du comportement en fonction d'un seuil.

Évaluation du bloc

Malgré les images supplémentaires et les compteurs en sortie, je me suis trouvé certaines fois à voir qu'il y avait des problèmes, mais pas leurs provenances. Pour avoir une vision claire, il aurait fallu que je sauvegarde d'une manière ou d'une autre toutes les valeur du bloc.

Cependant en navigant dans une image de 246x242x169 (10 060 908 voxel), sauvegarder un bloc contenant largeur x hauteur x profondeur donne très rapidement énormément de donnée. De plus, la plupart de ces données ne nous intéresse pas.

En discutant avec mon responsable, l'idée nous est venue de faire un programme à coté qui ne sert qu'à évaluer et visualiser un bloc à la fois. Il prend en entrée l'image à traiter et tourne en boucle autant de fois que désiré tout en offrant la possibilité de modifier tous les paramètres d'entrés du programme principal (taille bloc en X, Y et Z, seuil relatif de présence osseuse ...).

```

    Modifier la taille du bloc ? 0
BlocSizeX : 5
BlocSizeY : 5
BlocSizeZ : 3
    Modifier le mode de selection ? 0
Choisissez le mode de selection, 1 pour main, 2 par coordonnée : 2
    Modifier l'intensité ' ? 0
Rentrez un seuil d'intensité osseux : 100
    rentrer les coordonnées du voxel
X : 152
Y : 164
Z : 66

    Spécificité du bloc autour du voxel
VoxelValue : -319
NumberOfVoxelInTheBloc : 75
min_Blocs : -1024
max_Blocs : 64
sigmaValue : 364.691
voxelAverageValue : -419.133
voxelMedianValue : -366
voxelHarmonicValue : -390.769
voxelOtsuValue : -1024
seuil d'intensité osseux : 100
pourcentage de voxel supérieure au seuil : 0

    enregistrer les valeurs calculé du bloc ?
N
Appuyez sur ENTRER pour revenir au menu ou n'importe quel touche+ENTRER pour quitter :

```

Figure 6 : Terminal de commande du programme d'évaluation du bloc.

1.3. Evaluations des résultats et mise en évidence des problèmes

En annexe 2 se trouve le détail de six coupes d'une section d'image sur laquelle fut évalué le programme. L'annexe 2.1 représente la coupe 1ère, l'annexe 2.2 la coupe 11ème, 2.3 la 20ème, 2.4 la 28ème, 2.5 la 32ème et enfin la 2.6 la 42ème. La taille de l'image utilisée est de 21x32x42 voxels.

Première observation : méthodes de calcul

La méthode d'otsu donne les résultats les plus grossiers des quatre méthodes. Peu importe le type de coupe original ou la taille du bloc, l'image segmentée en fin de programme est la moins nette.

Il est possible que le problème vienne de ce que compose le bloc. En effet la méthode d'otsu est efficace s'il y a deux pics sur l'histogramme. Or, l'air, les tissus mous et les os (dense) peuvent faire apparaître trois pics. Avec ces trois pics, le seuillage en est peut-être perturbé.

Cependant, cela ne peut être le seul problème, car l'erreur de trois pic ne peut normalement pas être toujours présente. Or, on visualise sur les six figures de l'annexe 2, qu'à aucun moment la méthode d'otsu n'est efficace.

Sur l'ensemble des slices des différentes méthodes et taille, la méthode de la moyenne simple semble être la plus performante. Cependant, les différences de résultats actuelles ne sont pas assez importantes pour ne se baser que sur celle-ci.

Deuxième observation : Taille du bloc

Plus la taille du bloc est importante, plus sensible en sont les méthodes de calculs. Une taille importante apporte une meilleure détection sur des zones à faible niveau d'intensité (annexe 2.4, 2.5 et 2.6).

Cependant, l'augmentation de la sensibilité n'a pas que des avantages. En effet, dans des zones à tendance homogène, mais n'étant pas de l'os, des os sont détectés. L'amélioration de la sensibilité rend les méthodes de calculs plus sensibles au bruit (annexe 2.1, 2.2 et 2.3).

Avec l'augmentation de la taille du bloc, le temps de calcul augmente lui aussi. Pour calculer sur les 28 224 voxels avec un bloc de 3x3x3 prend environ 10 s avec la moyenne et augmente jusqu'à 1 min 20s pour la méthode d'otsu. L'image sur laquelle on va être amené à calculer est une image 10 millions de voxels, soit environ 357 fois plus.

La taille du bloc agit évidemment sur le nombre de voxel le composant. Une comparaison simple entre le maximum du bloc et un seuil relatif (fixé par l'utilisateur) sera plus simple avec un grand nombre de voxels.

Il faut ici mettre en place une (ou plusieurs) solution(s) permettant de réduire le temps de calcul en ignorant des voxels ne nous intéressant pas, régler le problème de sensibilité dans les zones homogènes et complexifier la comparaison relative.

2. Amélioration du programme

2.1. Idées de solutions

Les seuils absolus pour le temps de calcul.

En observant l'image, on se rend compte qu'un certain nombre de voxels représentant de l'air et de l'os sont facilement identifiable sans avoir besoin de calculer de seuil. Avant de constituer le bloc une double vérification sera faite.

Cette vérification sera la comparaison successive, de la valeur du voxel à un seuil absolu d'air et d'os. Ces deux seuils absolus sont deux nouvelles conditions d'entrée du programme. Leurs valeurs doivent correspondre à des valeurs dont on est sûr.

Écart-type pour les zones homogènes.

Une fois le bloc constitué, avant la vérification de présence osseuse, j'ajoute une étape où le programme va calculer l'écart-type du bloc afin de déterminer s'il est homogène.

S'il est homogène alors cela ne peut être du tissu mou, de l'os ou de l'air. Or, les seuils absolus permettent d'enlever les cas de bloc d'air ou os, homogènes.

Cependant, définir si une zone est homogène demande de comparer l'écart-type avec un seuil d'écart-type. Ce seuil fait alors partie des conditions d'entrées supplémentaires du programme.

Si l'écart-type calculé est supérieur au seuil, alors c'est une zone hétérogène et l'on vérifie la présence osseuse.

Pourcentage de présence osseuse relative.

Afin de rendre la détection de présence osseuse plus fiable, la présence osseuse ne vérifie plus si un voxel vérifie cette condition, mais un pourcentage de voxel qui vérifie cette condition.

Cela va ajouter une condition d'entrée supplémentaire, le pourcentage.

La mise en application est assez simple. Une fonction prenant le bloc de valeur, le pourcentage et le seuil osseux relatif.

Chaque valeur du bloc va ensuite être comparée au seuil relatif. S'il est supérieur un compteur s'incrémente de un, sinon rien.

Une fois toutes les comparaisons faites, le compteur est divisé par le nombre de voxels composant le bloc, donnant ainsi le pourcentage.

En sortie du programme on compare si le pourcentage calculé est supérieur au pourcentage entré. Si oui la fonction sort "True" sinon "False".

Couplé avec les seuils absolus et l'écart type, le seuil relatif peut être abaissé afin de rendre le programme encore plus sensible et améliore la finesse de la détection de présence osseuse.

Réécriture de la méthode d'otsu.

La méthode d'otsu fut le dernier travail de mon stage. En effet, M. SUBSOL trouvant que les résultats de la méthode d'otsu n'étaient pas satisfaisant, me demanda de reprogrammer celle-ci.

La nouvelle méthode d'otsu fonctionne de la manière suivante tel que décrit sur la figure 7.

Conditions d'entrées :

- Bloc de valeur

En sortie :

Une valeur correspondant au seuil

Programme :

Range dans l'ordre croissant les valeurs sous la forme d'un tableau

Calcul la variance de la 1ère valeur à la 1ère valeur (σ_{moins})

1 valeur dans le sous bloc moins (n_{moins})

Calcul la variance de la 2ème valeur à la dernière valeur (σ_{plus})

Toutes les valeurs moins 1 valeur dans le sous bloc plus (n_{plus})

Calcul de $\sigma_{moins} * n_{moins} + \sigma_{plus} * n_{plus}$ puis sauvegarde (Val)

Sauvegarde de la valeur du 2ème voxel

Pour i allant de la 2ème valeur à l'avant dernière valeurs

 Ajouter 1 à (n_{moins})

 Enlever 1 à (n_{plus})

 Calcul la variance de la 1ère valeur à la i -ème-1 valeur (σ_{moins})

 Calcul la variance de la i -ème-1 valeur à la dernière valeur (σ_{plus})

 Calcul d'un nouveau $\sigma_{moins} * n_{moins} + \sigma_{plus} * n_{plus}$

 Comparaison de celui-ci avec Val

 S'il est plus petit alors :

 | Changement de la valeur Val

 | i -ème voxel sauvegardé

fin

Sortie du voxel sauvegardé

Figure 7 : Synthèse du fonctionnement du programme d'Otsu.

2.2. Comparaison nouvelle / ancienne structure

Les deux programmes de la figure 8 sont les programmes de test de méthodes. C'est pour cette raison que l'image témoin est présente dans les conditions d'entrées. Cette condition n'aura plus lieu d'être à la fin.

La nouvelle version du programme possèdent quatre conditions d'entrées supplémentaires que le premier. On retrouve les conditions présenté précédemment : les seuils absolus d'air et d'os, l'écart-type (sigma) et le pourcentage de présence osseuse.

Améliorer la précision augmente le nombre seuil. On passe d'un seuil en entrée à 5 seuils d'entrées.

Le programme fonctionne toujours avec un système de boucle "if else". Si une condition est validée, le programme passe au calcul du voxel suivant.

<u>Première version</u>	<u>Dernière version</u>
<u>7 conditions d'entrée :</u>	<u>11 conditions d'entrée :</u>
- image à traiter	- image à traiter
- méthode	- méthode
- seuil de densité osseuse	- sigmaMinimal
- image témoin	- seuil absolu pour l'aire
- taille du bloc en X, Y et Z	- % présence os
	- image témoin
	- taille du bloc en X, Y et Z
	- seuil de densité osseuse relatif
	- seuil absolu pour l'os
<u>Abréviations :</u>	
- voxel : V	
- blocs : B	
- seuil : S	
Programme :	Programme :
<pre> for all V_{Trt} (image à traiter) calcul B(V_{Trt}) if V ∈ B > Seuil densité os calcul S_(B) if S(B) < V_{Trt} alors V_{Bin} = 1 else V_{Bin} = 0 else V_{Bin} = 0 </pre>	<pre> for all V_{Trt} (image à traiter) if V_{Trt} > seuil absolu os alors V_{Bin} = 1 else if V_{Trt} < seuil absolu air alors V_{Bin} = 0 else calcul B(V_{Trt}) calcul σ(B) if σ(B) < σ_{min} alors V_{Bin} = 0 else if % V ∈ B calcul S_(B) if S(B) < V_{Trt} alors V_{Bin} = 1 else V_{Bin} = 0 </pre>
<p>□ P1</p> <p>□ Cas 1</p> <p>□ Cas 2</p>	<p>□ Cas 3</p> <p>□ Cas 4</p> <p>□ Cas 5</p> <p>□ P1</p> <p>□ Cas 1</p> <p>□ Cas 2</p>
<div style="border: 1px solid black; padding: 5px; display: inline-block;"> Figure 8 : Comparaison entre l'ancien et le nouveau programme. </div>	

2.3. Nouveaux résultats

Effet des seuils absolus & temps de calculs :

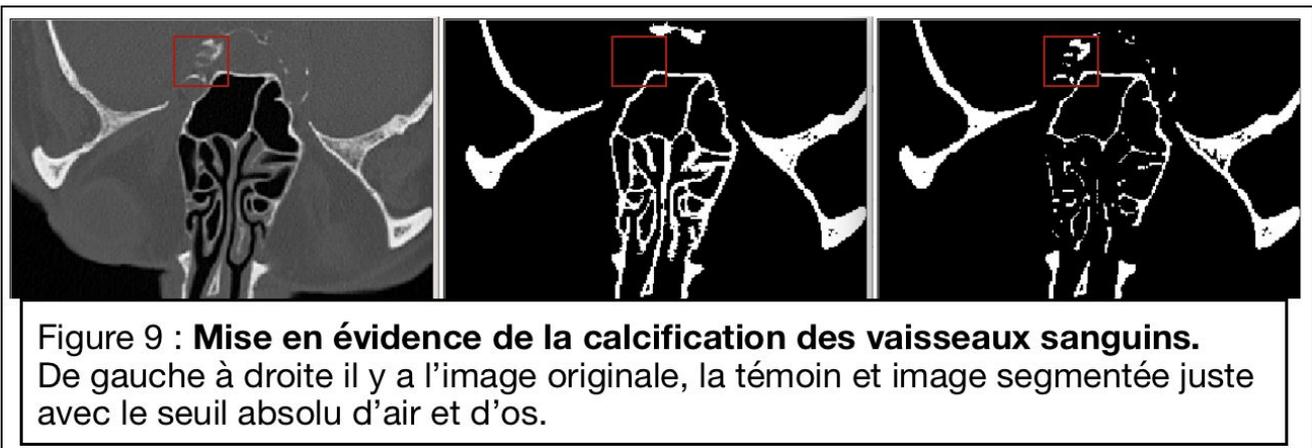
Après quelques tests, je suis arrivé à déterminer des seuils réduisant grandement le temps de calculs tout en minimisant les erreurs apportées.

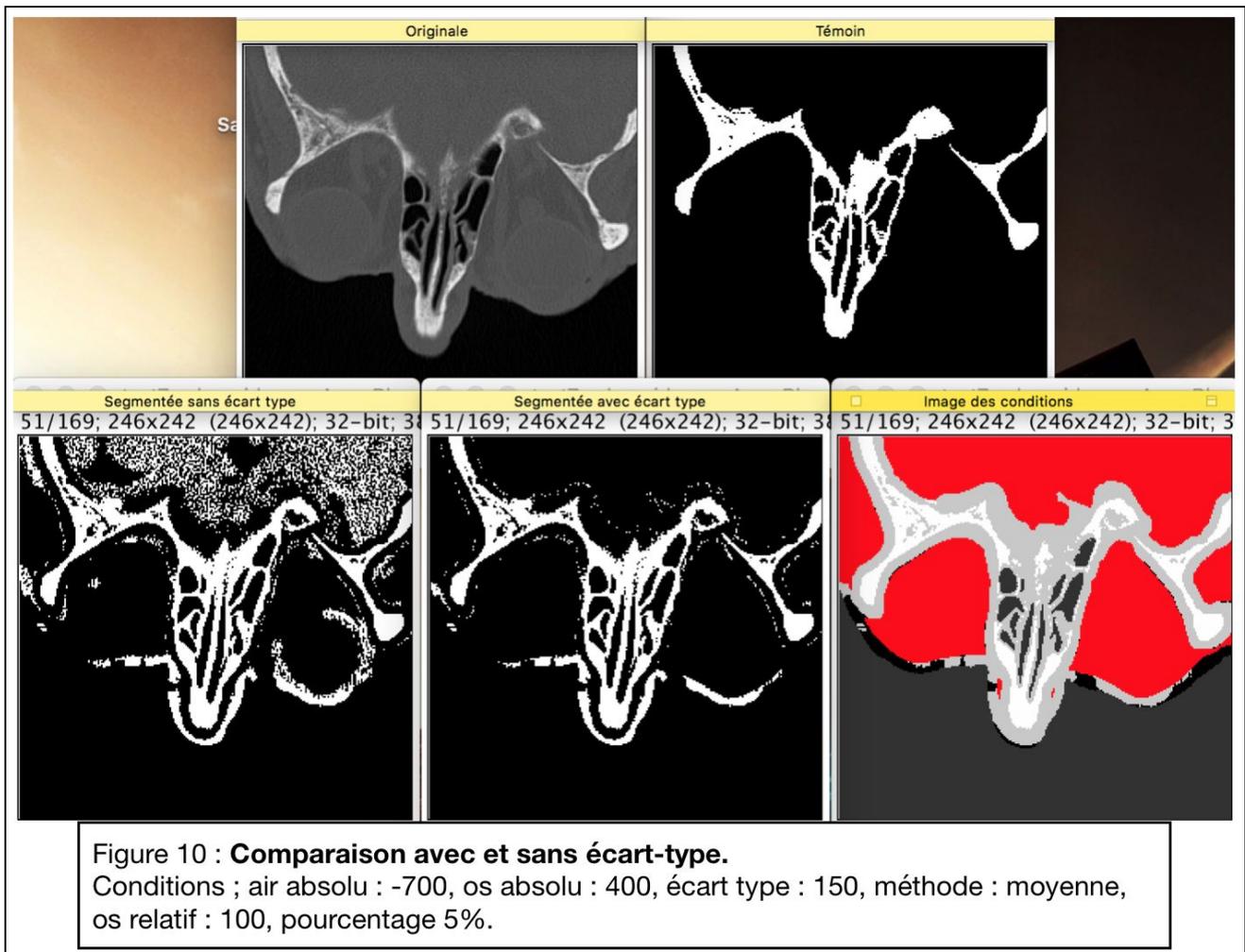
Avec un seuil d'os absolu à 400, je suis arrivé à détecter 777 433 voxels correspondants à de l'os. Sur l'image témoin il y a 943 444 voxels correspondant à de l'os. Sur les 777 433 seuil 2 480 voxel sont faux. Cela représente un taux d'erreur à 0,0246 %, pour une détection de 82,14 % de voxels d'os.

De plus, une grande partie du taux d'erreur ne provient pas du calcul mais des conditions liées à l'image. En effet, des parois de vaisseaux sanguins sont calcifiées et ont une densité similaire à de l'os. Ce type d'erreur ne peut être réglé qu'à la main par un expert. Sur la figure 9 j'ai mis en évidence ces vaisseaux.

Avec un seuil d'air absolu à -700, je suis arrivé à détecter 3 781 708 voxels correspondants à de l'air. Sur l'image témoin, le nombre de voxels en noir montent à 9 283 475. Il faut cependant prendre en compte que, sur la témoin, les voxel noirs ne représentent pas que l'air, mais les tissus mous.

En tout, les seuils absolus enlève 4 559 141 voxels, soit presque la moitié de 10 060 908 voxels. Cela permet de gagner tout le temps de calcul sur : constitution du bloc, calcul écart type et enfin calcul de la méthode (détail des temps de calculs dans le tableau 1).

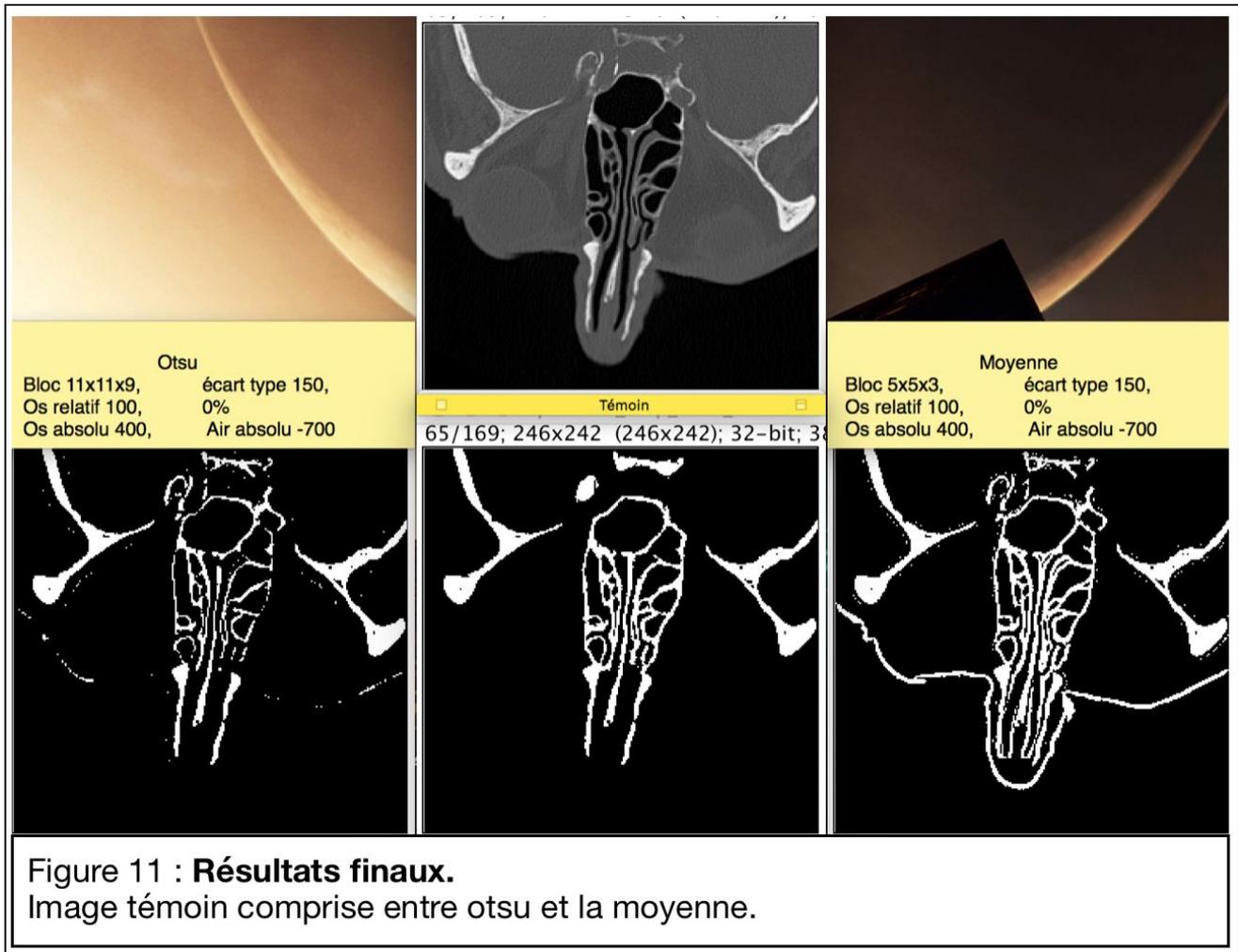


Effet de l'écart type & temps de calculs :

Sur la figure 10, l'utilité de l'écart type est nettement visible. Avec un bloc important comme ici (11x11x9) les méthodes (ici la moyenne) sont beaucoup plus sensibles.

Sur l'image des conditions, le code couleur est le suivant : rouge voxels identifiés avec l'écart type, gris clair ceux calculés avec la méthode (moyenne), blanc ceux avec le seuil d'os absolu, gris foncé ceux avec le seuils absolu d'air et en noir ceux qui ne rentrent dans aucunes conditions.

De plus, on peut observer sur le tableau 1 en annexe 1 que l'écart type, en théorie peut réduire du temps de calcul. Cela reste cependant relatif à la méthode de calcul. La méthode d'otsu profite de la réduction du temps.

Une image segmentée proche & des conditions encadrant l'image témoin

La mise en commun de toutes les solutions développées, est présentée sur la figure 11. On peut y observer l'efficacité ainsi que la finesse de la nouvelle méthode d'otsu. Contrairement aux trois autres méthodes, la méthode d'otsu a moins de faux positifs que de faux négatifs (3 fois moins).

L'augmentation de la taille du bloc rend la méthode d'otsu de plus en plus précise. Cependant, les temps de calculs deviennent extrêmement longs. L'image segmentée sur la figure 11 d'Otsu a demandé plus de deux heures de calcul. Arrivant en fin de stage, je n'ai pas eu le temps de tester avec des blocs plus grand.

Sur les 3 autres méthodes, le pourcentage a permis d'affiner les résultats. Cependant, la méthode d'Otsu n'a pas besoin de l'affinement qu'apporte le pourcentage de présence osseux. Les os détectés sont suffisamment fins, contrairement à la moyenne comme on peut le voir sur la figure 11.

La moyenne avec un bloc 5x5x3 permet d'encadrer l'image témoin. Elle pose une limite haute, et la méthode d'otsu pose une limite basse.

Conclusion et perspectives

Tout le travail réalisé rentre dans le contexte d'imprimer le crâne, et particulièrement les sinus paranasaux. Tous les enjeux du développement de cette technique sont présentés dans la partie contexte et enjeux du stage.

Plusieurs pistes sont envisageables afin d'améliorer la segmentation.

On a vu que l'image témoin se situait entre l'image d'Otsu et l'image de la moyenne. On peut essayer de mettre en place un programme comparant les deux, et utilisant leurs images de seuils respectives.

Il est possible de lancer le programme actuel avec un plus grand bloc et attendre le temps nécessaire.

Afin de réduire le temps de calcul, il serait possible de réfléchir à une optimisation (au sens informatique) de la méthode d'Otsu. N'étant pas un informaticien de base, il est probable que mon programme prenne des ressources, de la RAM et du temps de calcul du processeur pour rien.

L'image en sortie aura dans tous les cas besoin d'une mise en place de post-traitements. Cela permettra de résoudre les quelques voxels isolés visibles sur la figure 11, pour la méthode d'Otsu.

Il est envisageable de mettre en place un programme permettant de vérifier quelles structures sont attachées entre elles. Les structures isolées sont présentées à l'expert qui valide leur suppression ou au contraire pense qu'elles sont importantes.

Annexes

Annexe 1 : Tableaux

Tableau 1 : Evaluation du temps de calcul pour un Bloc de 5x5x3 et 11x11x9

Remarque : Les temps de calculs totaux de chaque méthode ne sont pas les temps exact car ils prennent le cas le plus long

Attention les temps du tableau ne sont pas les temps immuable. En effet, ce temps dépend de la puissance de calcul de mon ordinateur. Les temps servent juste à visualiser quelles méthodes de sont les plus courtes, lente...

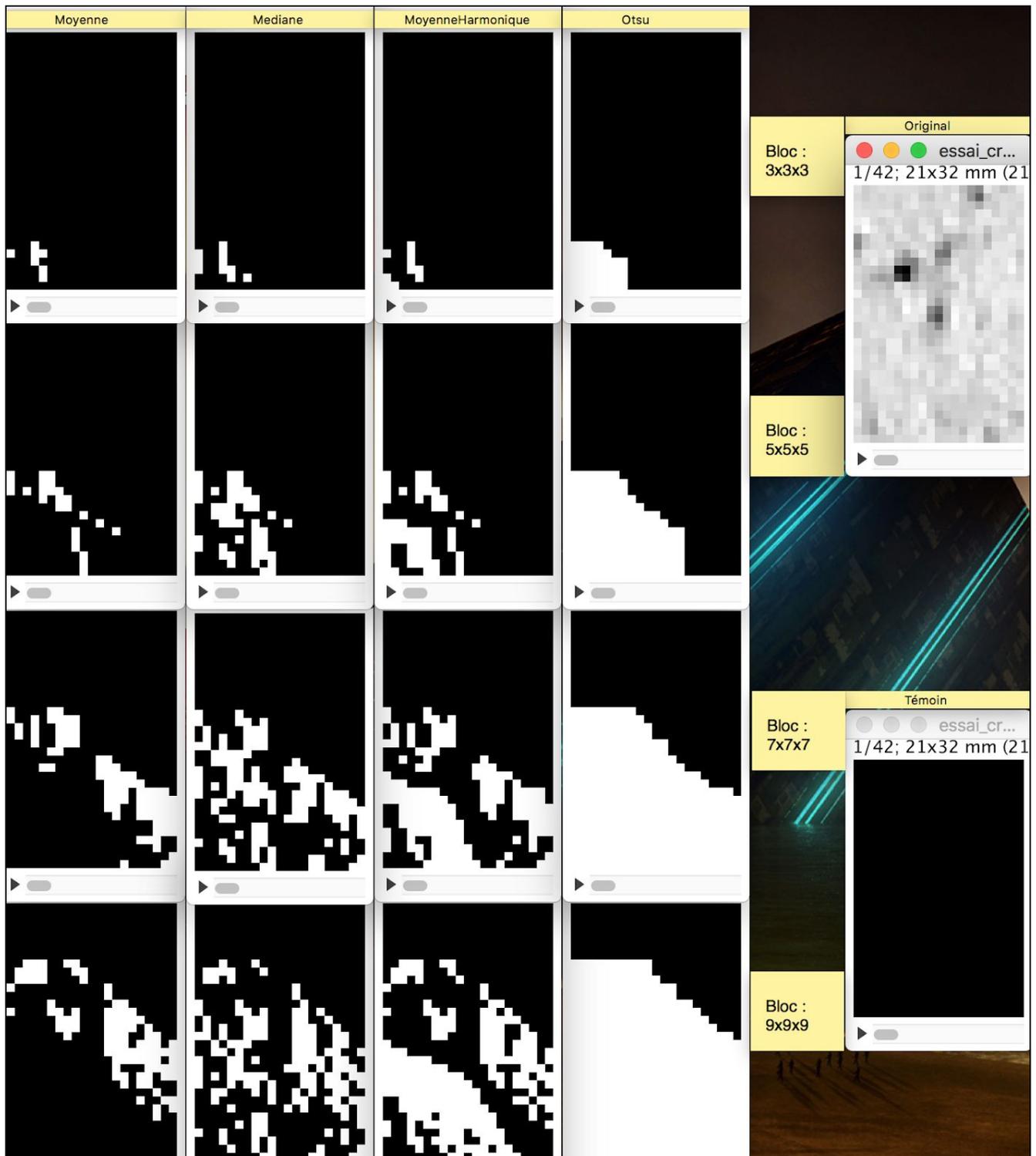
Temps d'exécutions (s)	Bloc 5x5x3	Bloc 11x11x9	10 060 908		Min		heure	
Constitution du	7,800E-05	8,200E-05	784,751	824,994	13,1	13,7	0,2	0,2
Ecart type	3,000E-06	6,000E-06	30,183	60,365	0,5	1,0	0,0	0,0
Calcul pourcentage	4,000E-06	1,200E-05	40,244	120,731	0,7	2,0	0,0	0,0
moyenne seul	3,000E-06	5,000E-06	30,183	50,305	0,5	0,8	0,0	0,0
Total	8,800E-05	1,050E-04	885,360	1 056,395	14,8	17,6	0,2	0,3
Médiane seul	9,000E-06	7,200E-05	90,548	724,385	1,5	12,1	0,0	0,2
Total	9,400E-05	1,720E-04	945,725	1 730,476	15,8	28,8	0,3	0,5
Moyenne	1,100E-05	7,400E-05	110,670	744,507	1,8	12,4	0,0	0,2
Total	9,600E-05	1,740E-04	965,847	1 750,598	16,1	29,2	0,3	0,5
Otsu seul	7,900E-05	1,697E-03	794,812	17 073,361	13,2	284,6	0,2	4,7
Total	1,640E-04	1,797E-03	1 649,989	18 079,452	27,5	301,3	0,5	5,0

Tableau 2 : Evolution du comportement de l'image en fonction des divers seuils

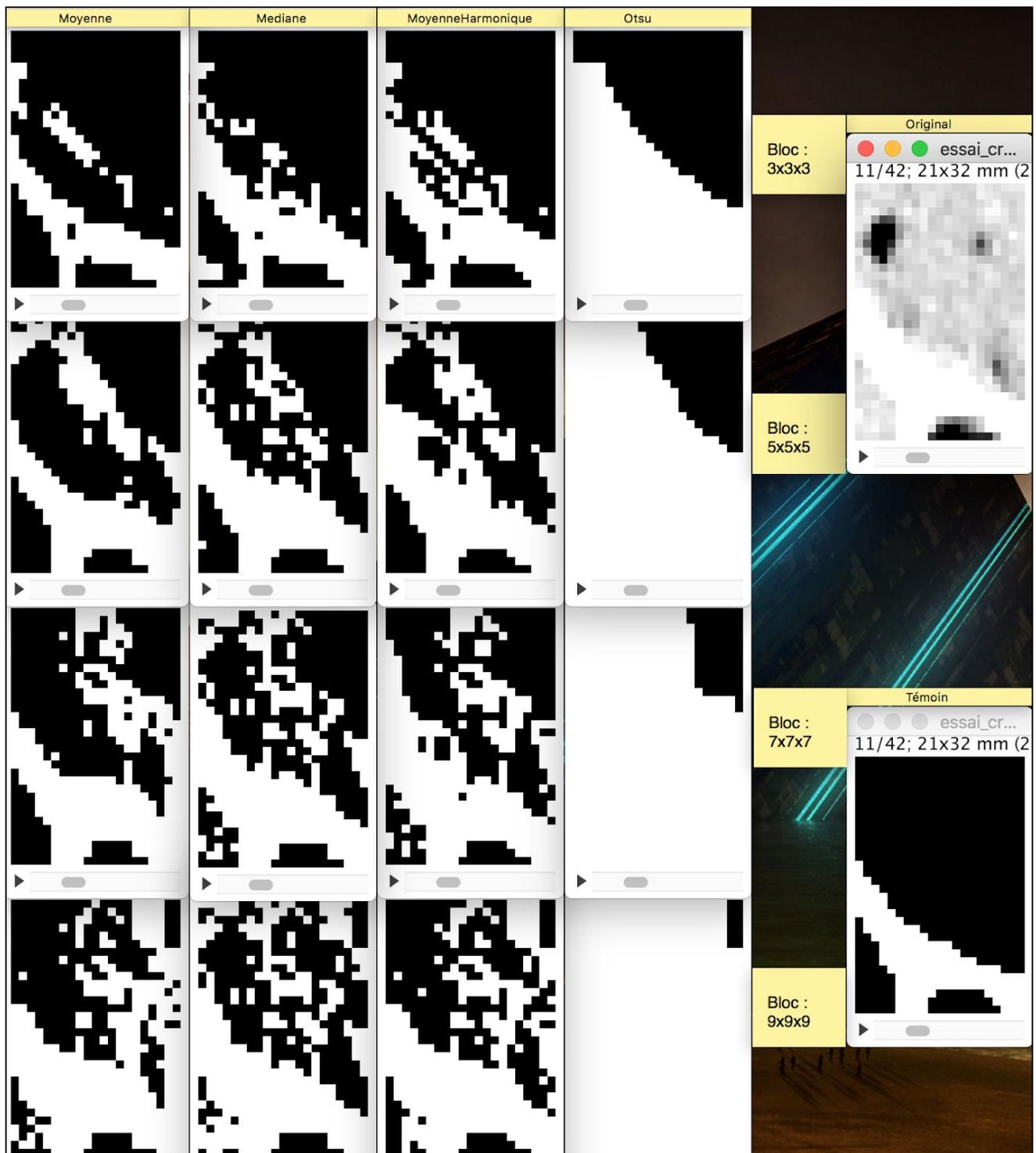
Changements	methode	bloc XYZ	sigma Min	Seuil os	seuil abs Os	seuil abs Air	% Os	Présence os	Présence air ou tissu	Fausse présence tissu ou autre	Fausse présence os	Nombre de points total	% Faux négatif	% Faux Positif	sensibilité	spécificité	
Variation %	mediane	775	100	300	300	-700	7	84 640	355 359	20 661	27 240	487 900	4,2347	5,5831	0,8038	0,9288	
	mediane	775	100	300	300	-700	8	83 684	356 480	21 617	26 119	487 900	4,2347	5,5831	0,8038	0,9288	
	mediane	775	100	300	300	-700	9	82 223	358 151	23 078	24 448	487 900	4,7301	5,0109	0,7808	0,9361	
	mediane	775	100	300	300	-700	10	81 189	359 210	24 112	23 389	487 900	4,7301	5,0109	0,7808	0,9361	
	mediane	775	100	300	300	-700	11	80 129	360 251	25 172	22 348	487 900	4,9420	4,7938	0,7710	0,9389	
	mediane	775	100	300	300	-700	12	78 639	361 590	26 662	21 009	487 900	5,1593	4,5804	0,7610	0,9416	
	mediane	775	100	300	300	-700	13	77 595	362 642	27 706	19 957	487 900	5,4646	4,3060	0,7468	0,9451	
	Variation seuil Intensité os	mediane	775	100	250	300	-700	10	85 191	356 641	20 110	25 958	487 900	4,1217	5,3204	0,8090	0,9322
		mediane	775	100	260	300	-700	10	84 449	357 167	20 862	25 432	487 900	4,2738	5,2125	0,8020	0,9335
		mediane	775	100	270	300	-700	10	83 596	357 668	21 705	24 931	487 900	4,4487	5,1099	0,7939	0,9348
mediane		775	100	280	300	-700	10	82 846	358 196	22 455	24 403	487 900	4,6024	5,0016	0,7868	0,9362	
mediane		775	100	290	300	-700	10	82 076	358 716	23 225	23 883	487 900	4,7602	4,8951	0,7794	0,9376	
mediane		775	100	300	300	-700	10	81 189	359 210	24 112	23 389	487 900	4,9420	4,7938	0,7710	0,9389	
Variation seuil absolu d'os		mediane	775	100	300	100	-700	10	88 561	344 636	16 740	37 963	487 900	3,4310	7,7809	0,8410	0,9008
		mediane	775	100	300	110	-700	10	87 995	348 758	17 306	33 841	487 900	3,5470	6,9361	0,8357	0,9115
		mediane	775	100	300	120	-700	10	87 486	351 720	17 815	30 879	487 900	3,6514	6,3290	0,8308	0,9193
		mediane	775	100	300	130	-700	10	87 005	353 785	18 296	28 834	487 900	3,7499	5,9098	0,8263	0,9246
	mediane	775	100	300	140	-700	10	86 522	355 191	18 779	27 408	487 900	3,8489	5,6175	0,8217	0,9284	
	mediane	775	100	300	150	-700	10	86 048	356 091	19 253	26 508	487 900	3,9461	5,4331	0,8172	0,9307	
	mediane	775	100	300	160	-700	10	85 595	356 685	19 706	25 914	487 900	4,0389	5,3113	0,8129	0,9323	
	Variation écart type (sigma min)	mediane	775	70	300	300	-700	10	81 189	359 210	24 112	23 389	487 900	4,9420	4,7938	0,7710	0,9389
		mediane	775	80	300	300	-700	10	81 189	359 210	24 112	23 389	487 900	4,9420	4,7938	0,7710	0,9389
		mediane	775	90	300	300	-700	10	81 189	359 210	24 112	23 389	487 900	4,9420	4,7938	0,7710	0,9389
mediane		775	100	300	300	-700	10	81 189	359 210	24 112	23 389	487 900	4,9420	4,7938	0,7710	0,9389	
mediane		775	110	300	300	-700	10	81 189	359 210	24 112	23 389	487 900	4,9420	4,7938	0,7710	0,9389	
mediane		775	120	300	300	-700	10	81 188	359 210	24 113	23 389	487 900	4,9422	4,7938	0,7710	0,9389	
mediane		775	130	300	300	-700	10	81 183	359 225	24 118	23 374	487 900	4,9432	4,7907	0,7710	0,9389	
mediane		775	140	300	300	-700	10	81 149	359 261	24 152	23 338	487 900	4,9502	4,7834	0,7706	0,9390	

Annexe 2 : Les 1ers résultats

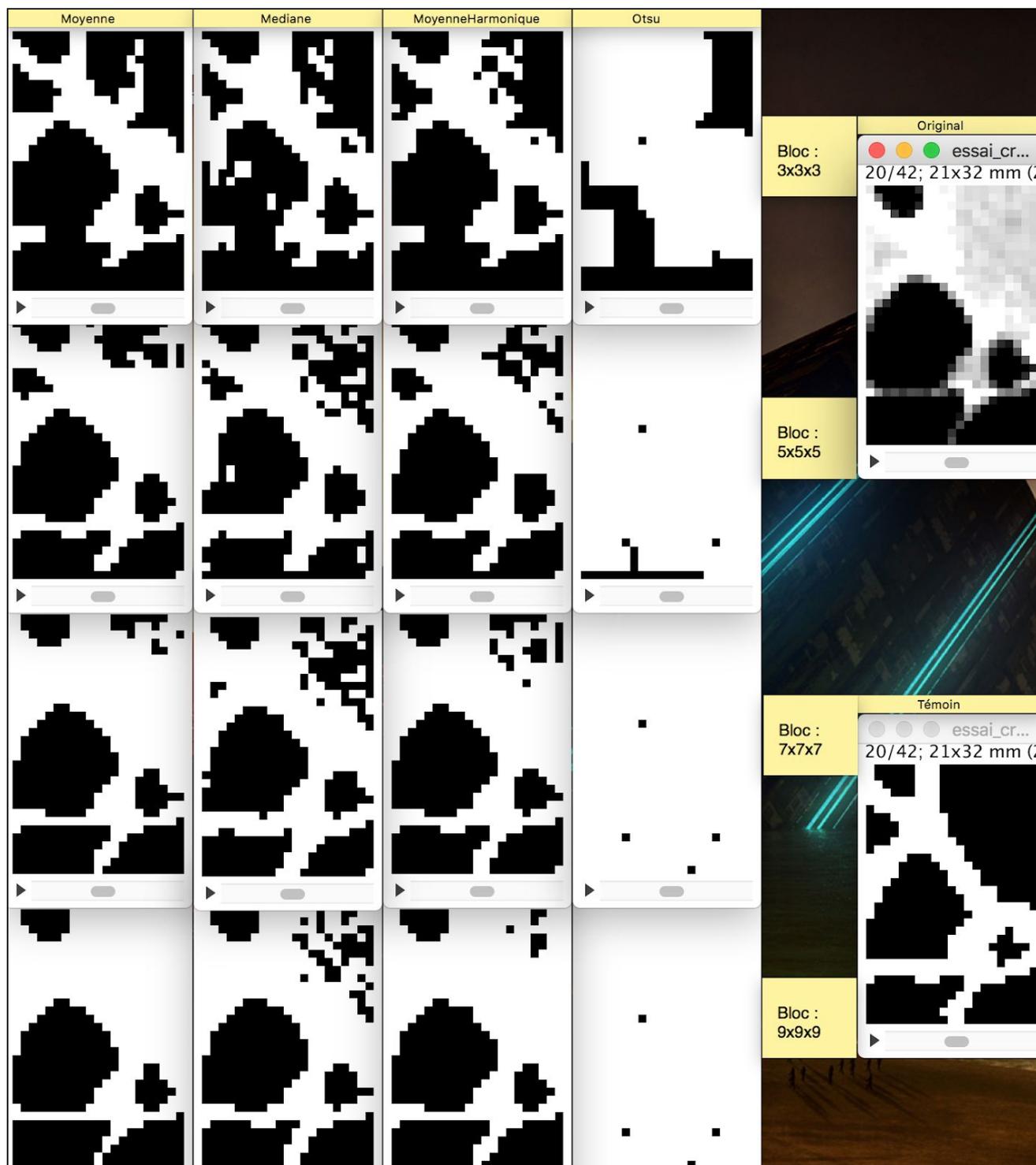
Annexe 2.1 : Coupe 1 de l'image de test



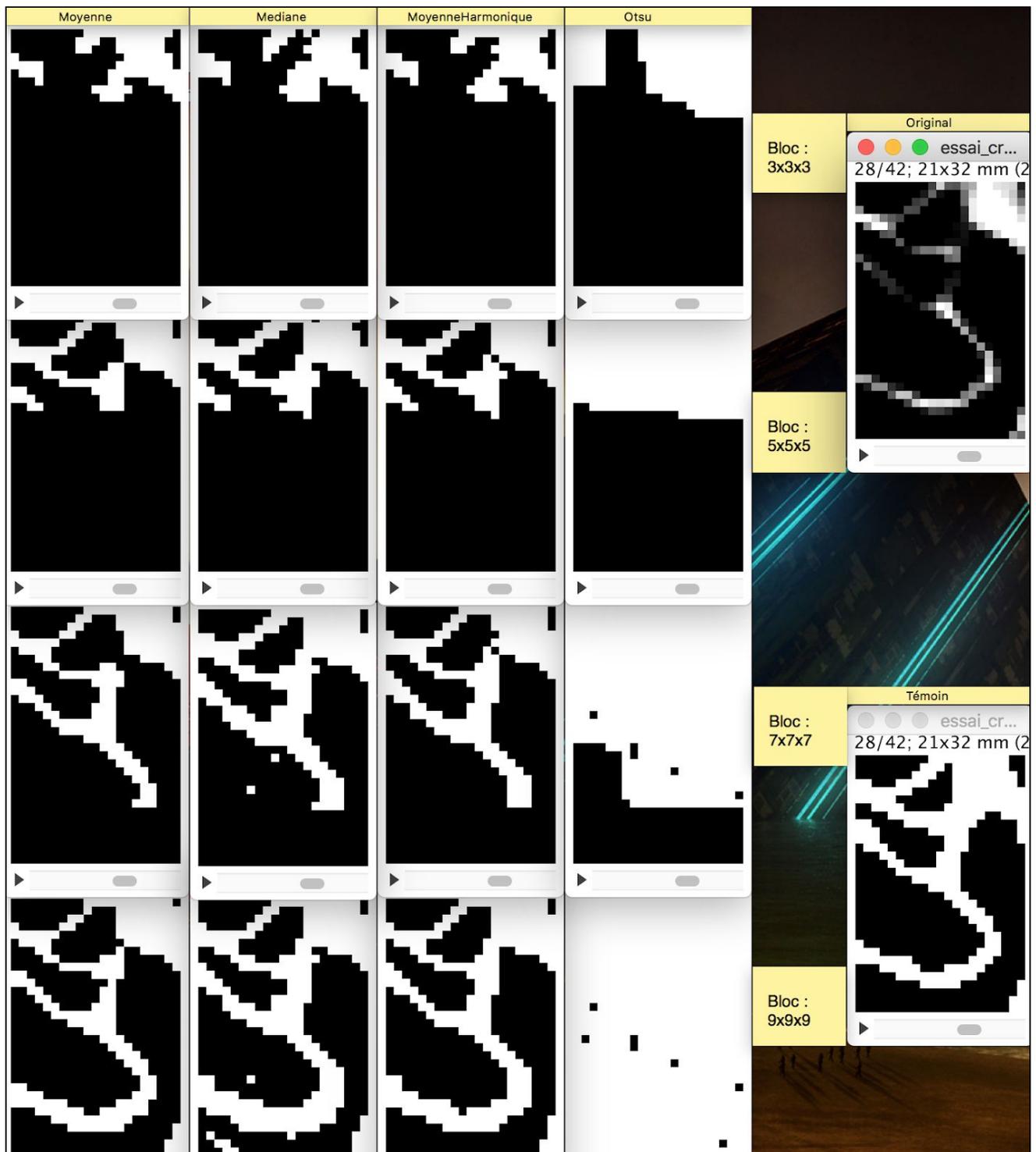
Annexe 2.2 : Coupe 11 de l'image de test



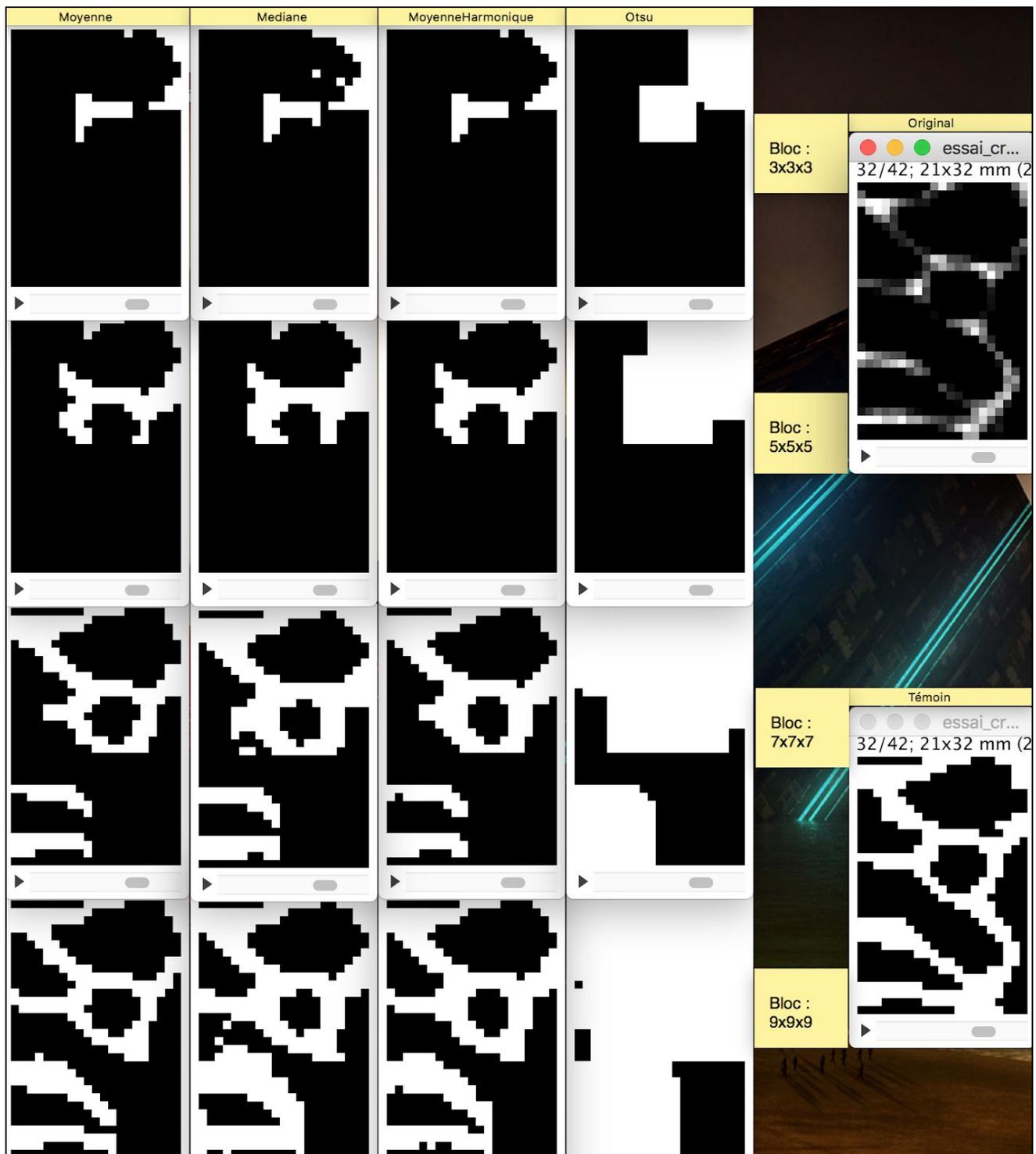
Annexe 2.3 : Coupe 20 de l'image de test



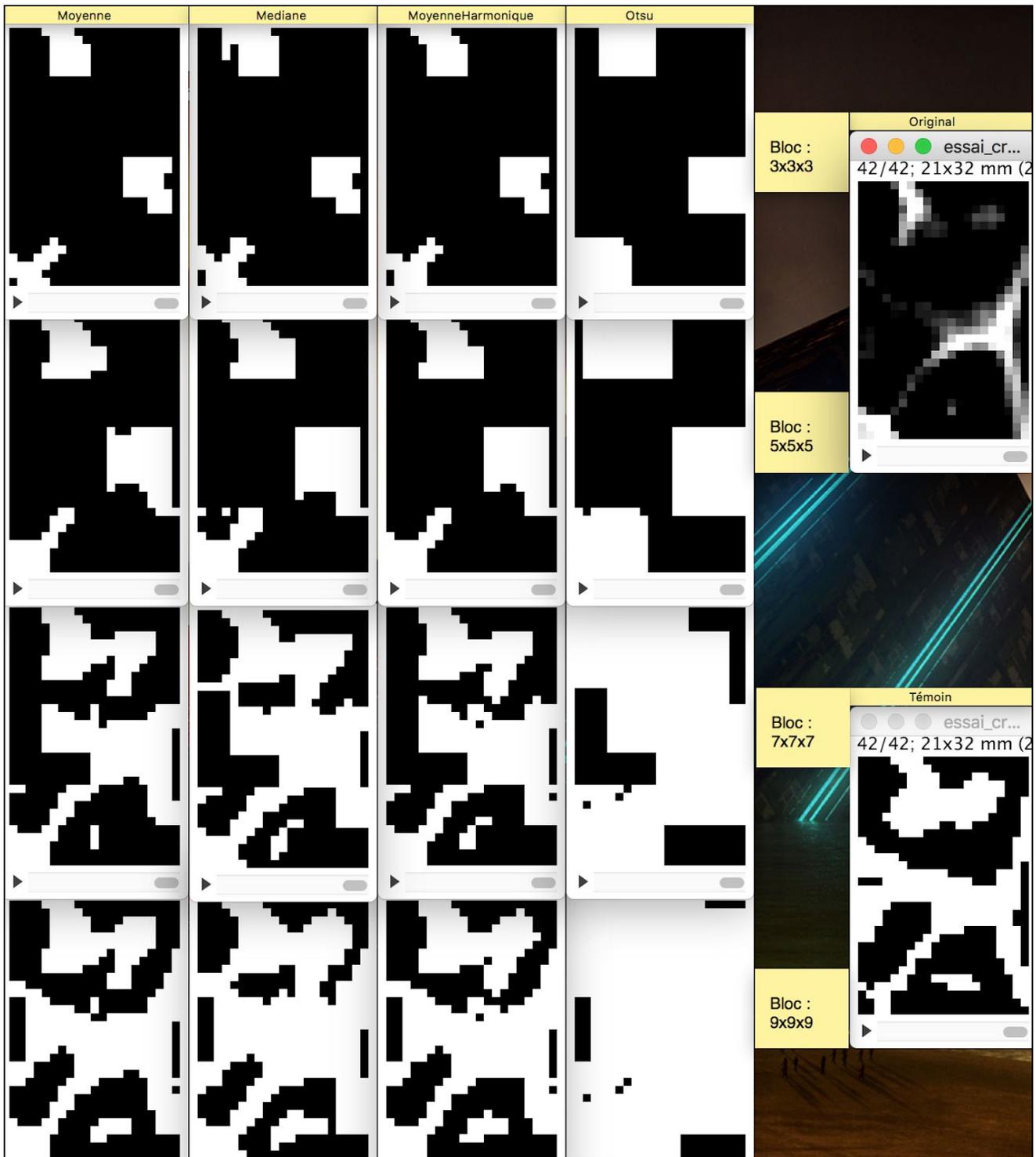
Annexe 2.4 : Coupe 28 de l'image de test



Annexe 2.5 : Coupe 32 de l'image de test



Annexe 2.6 : Coupe 42 de l'image de test



Bibliographie

[1] N. L. Bui, S. H. Ong and K. W. Foong, “Automatic segmentation of the nasal cavity and paranasal sinuses from cone-beam CT images”, 2014

[2] H. Shi, W. C. Scarfe, and A. G. Farman, “Maxillary sinus 3d segmentation and reconstruction from cone beam ct data sets”, 2006.