

Traitement & Analyse d'Images

Introduction à la recherche



Détection de structure linéaire 3D, détection des vaisseaux sanguins en imagerie médicale : RORPO

Richard FOUQUOIRE

Emily RENARD

E4FI

ESIEE
PARIS

Table des matières

I.	Introduction.....	3
II.	Synthèse de l'article	3
A.	Contexte	3
B.	Objectifs.....	3
C.	Hypothèses.....	4
D.	Méthodes	4
1.	Orientations possibles	5
2.	Classification des réponses d'orientation	6
E.	Méthodologie de validation	7
F.	Résultats	7
G.	Opinion personnelle	8
III.	Étude de l'implémentation.....	8
IV.	Expérimentations	9
V.	Sources	13

I. Introduction

Ce rapport vient conclure notre cours de traitement et d'analyse d'images de quatrième année. Il s'agit d'une introduction à la recherche dont le but est de découvrir une nouvelle méthode de traitement d'image présentée sous la forme d'un article de recherche choisi.

Ce rapport permet de dresser une synthèse de l'article, l'étude de l'implémentation de la technique étudiée et un bilan de nos expérimentations.

Nous avons pour notre part, choisi de s'intéresser à l'imagerie médicale 3D à travers la détection des vaisseaux sanguins. Nous nous basons sur les travaux de recherche de l'article suivant : « *Curvilinear Structure Analysis by Ranking the Orientation Responses of Path Operators* » ([RORPO \(esiee.fr\)](https://esiee.fr/RORPO)).

Toutes les figures, schémas et images utilisées dans ce rapport ont été prises du rapport énoncé ci-dessus de manière à mieux illustrer nos propos.

II. Synthèse de l'article

A. Contexte

Aujourd'hui, de nombreuses techniques permettent l'imagerie médicale, nous sommes capables de créer des images 3D à l'aide de moyens de plus en plus poussés. Il faut aussi analyser précisément ces images pour pouvoir les rendre exploitables : il faut en effet distinguer les différents éléments qui les composent de manière automatique pour les rendre accessibles aux professionnels de la santé.

Cependant, détecter des structures linéaires comme des vaisseaux sanguins dans des images 3D peut être assez complexe. Actuellement, de nombreux filtres d'analyse existent déjà, chacun avec leurs atouts et inconvénients, dont les plus connus sont le *Frangi Vesselness (FV)*, l'*Optimally Oriented Flux (OOF)* et le *Hybrid Diffusion with Continuous Switch (HDSCS)*.

Quatre chercheurs français, Odyssée Merveille, Hugues Talbot, Laurent Najman et Nicolas Passat, nous proposent à travers cet article un nouvel opérateur surnommé **RORPO** : *Ranking the Orientation Responses of Path Operators* qu'on pourrait traduire en français par « Classement des réponses d'orientation des opérateurs connexes », un terme assez flou au premier abord qui résume simplement les techniques utilisées pour son fonctionnement.

B. Objectifs

Les structures curvilignes sont parmi les plus dures à gérer en traitement d'image : cette difficulté réside à la fois dans la rareté de ces structures et par leur géométrie complexe. De plus, ces structures sont assez fragiles de par leur finesse : du simple bruit sur l'image peut par exemple perturber la détection de contours.

Il existe déjà des méthodes pour détecter ces structures développées durant ces deux dernières décennies via des approches linéaires (basées sur la représentation d'espace échelle¹) et non-linéaires

¹ Théorie formelle pour manipuler les structures de l'image à différentes échelles, paramétrée par la taille d'un noyau lissant utilisé pour supprimer les structures dans les petites échelles.

(basées sur la morphologie mathématique). Toutes se basent sur le filtrage d'images 3D en niveau de gris.

Le RORPO s'ajoute à cette famille de méthodes en estimant deux nouvelles caractéristiques sur les structures curvilignes :

- Une caractéristique d'**intensité**, assimilée à une mesure quantitative de la curvilinearité².
- Une caractéristique **directionnelle**, apportant une mesure quantitative de l'orientation de la structure.

C. Hypothèses

Le RORPO est construit en se basant sur les techniques de la morphologie mathématique et des opérateurs connexes.

Comparé aux autres opérateurs, il est non-linéaire, non-local et discret. Cette méthode se distingue des autres filtres car elle apporterait 8% plus de « vrai positifs » et 50% moins de « faux positifs ».

En effet, les filtres orientés utilisés jusqu'à maintenant déterminent la structure linéaire dans un voisinage isotrope, dépendant de la taille et de l'échelle de la structure recherchée. Une approche non optimale étant donné que les structures linéaires sont très anisotropes, cela peut mener à des fausses détections, une mauvaise estimation de l'orientation des structures, surtout au niveau des frontières des structures.

Le RORPO utilise les opérateurs connexes pour contrer ce problème, ils permettent de déterminer les structures linéaires dans un environnement anisotrope. De plus, ils arrivent à mieux gérer la tortuosité³ d'une structure linéaire : si la structure diverge de l'orientation choisie, les opérateurs connexes peuvent quand même détecter cette structure.

Enfin, les opérateurs connexes demandent très peu de paramètres, le seul paramètre pris en compte correspond à la taille du chemin voulu, corrélé à la taille de notre structure linéaire voulue.

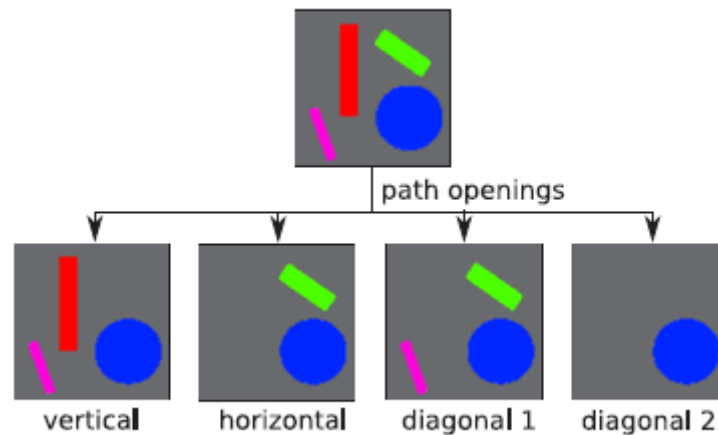
D. Méthodes

La stratégie du RORPO est de pouvoir détecter les structures linéaires en comptant le nombre de réponses fortes sur différentes orientations 3D. C'est-à-dire qu'on va pouvoir déterminer une structure linéaire en fonction du nombre de fois où elle va apparaître selon les filtres orientés (en appliquant nos opérateurs connexes) appliqués.

On reprend la figure 6 du rapport pour pouvoir illustrer cette classification des filtres orientés sur une image à deux dimensions :

² L'habilité pour un objet d'être composé par une ou plusieurs courbes. Plus un objet est curvilinéaire, plus il sera « courbé » et « fluide », à l'inverse, moins il le sera, plus il sera « droit » et « rigide ».

³ La tortuosité définit pour une courbe ou une structure le fait d'être plus ou moins tordu comme le fait d'avoir plusieurs virages.



Notre image d'origine se situe tout en haut, avec une barre verticale (en rouge), deux barres obliques (verte et rose) et un disque (bleu).

En appliquant un filtre vertical, on ne garde donc seulement les objets étant présents dans l'orientation verticale : notre barre verticale (logique), notre disque (orienté dans toutes les directions) et notre barre oblique rose. De même, dans le filtre horizontal, on va retrouver le disque et notre barre oblique verte. Etc.

Nous verrons plus tard que nos filtres d'orientation sont assez permissibles, d'où le fait que l'on puisse retrouver nos barres obliques roses et vertes dans des filtres verticaux ou horizontaux, on utilise pour cela des cônes angulaires.

Appliquer ces filtres est intéressant dans le cas de la recherche de structures curviligne car on sait que ces structures sont détectées dans moins d'orientations que les autres structures. Plus précisément, on sait qu'une structure linéaire est détectée dans, au plus, 3 orientations différentes en 3D.

1. Orientations possibles

On rappelle que les opérateurs connexes sont définis en fonction d'une relation d'adjacence. Cette relation d'adjacence peut définir une orientation générale. Cela se complexifie avec les images 3D, qui, avec une dimension en plus, définissent encore plus d'orientations possibles.

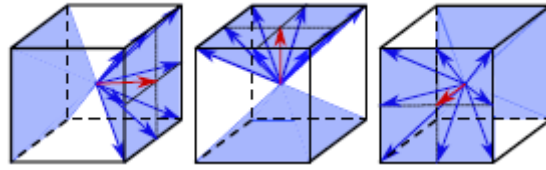
Afin de préserver les structures linéaires dans toutes les orientations possibles, plusieurs opérateurs connexes avec différentes orientations doivent être combinés.

Pour le RORPO, on décide de choisir un set de 7 orientations possibles :

- Les 3 principales correspondent au repère orthonormé (x, y, z) de l'espace 3D, définis respectivement par les vecteurs $\{1, 0, 0\}$, $\{0, 1, 0\}$ et $\{0, 0, 1\}$ que l'on nomme $Ce1, Ce2, Ce3$.
- Les 4 « diagonales » de l'espace 3D, définies respectivement par les vecteurs $\{-1, 1, -1\}$, $\{-1, 1, 1\}$, $\{1, 1, 1\}$ et $\{1, 1, -1\}$ que l'on nomme $Cd1, Cd2, Cd3, Cd4$.

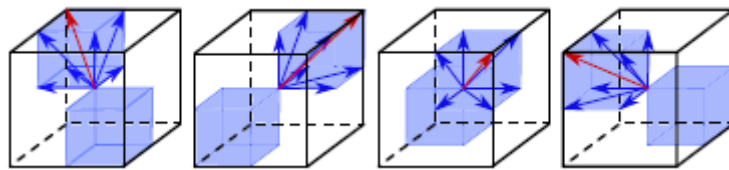
On utilise des cônes angulaires (dont on peut définir l'ouverture) autour de ces vecteurs pour englober plus facilement l'orientation de notre structure linéaire qui est imprécise. En effet, les modèles de base pour chacune des orientations doivent couvrir le voisinage de chaque point.

Pour représenter plus facilement ces orientations, on utilise schématiquement un cube qui permet de représenter notre espace 3D. On reprend les illustrations de la figure 5 du rapport :



On voit ci-dessus en rouge les vecteurs qui définissent l'orientation globale. De gauche à droite, on retrouve nos 3 orientations définies par notre repère orthonormé : d'abord sur l'axe x ($\{1, 0, 0\}$), puis sur l'axe z ($\{0, 0, 1\}$) et enfin sur l'axe y ($\{0, 0, 1\}$). ($Ce1, Ce2, Ce3$)

Les flèches bleues représentent la relation d'adjacence avec les limites du cône angulaire qui englobe notre orientation globale.



On retrouve ici la même chose avec le reste de nos orientations globales (les 4 « diagonales » de l'espace 3D). ($Cd1, Cd2, Cd3, Cd4$).

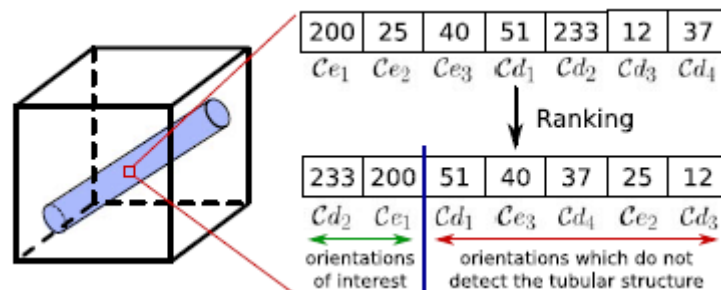
2. Classification des réponses d'orientation

Nous avons besoin d'obtenir une décision binaire pour trouver les orientations dans laquelle notre structure curviligne est impliquée : soit une orientation détecte une structure curviligne, soit elle ne la détecte pas.

Cependant, notre filtre utilise des niveaux de gris et allant sur un octet (0 à 255), plus le nombre est élevé, plus la réponse à l'orientation est forte (c'est-à dire à quel point l'objet est orienté dans cette direction).

On propose donc de séparer nos 7 filtres à orientations différentes en deux classes : celles qui détectent les structures curvilignes (surnommées **orientations d'intérêt**) et les autres.

On reprend la figure 7 du rapport pour illustrer cette classification :



On voit que notre filtre curvilinéaire a une forte réponse dans les filtres $Ce1$ et $Cd2$, qu'on identifie donc comme nos orientations d'intérêt.

E. Méthodologie de validation

Pour évaluer leur méthode, les chercheurs ont d'abord comparé avec les méthodes les plus connues, déjà énumérées plus tôt : *Frangi Vesselness (FV)*, *l'Optimally Oriented Flux (OOF)* et le *Hybrid Diffusion with Continuous Switch (HDCS)*, avec les paramètres optimisés indépendamment pour chaque méthode. Chacune des méthodes retourne une image en niveaux de gris, qui est comparée avec les autres.

On a ajouté le RORPO 7 et le RORPO 13 pour comparer ces méthodes (respectivement la méthode RORPO avec 7 classifications d'orientation possibles et la méthode RORPO avec 13 classifications d'orientation possibles).

Trois critères d'évaluation standards pour **la caractéristique d'intensité** ont été utilisés :

- Le Coefficient de Corrélation de Matthews (Matthews Correlation Coefficient (MCC)).
- Le taux de vrais positifs (TPR pour True Positives Rate).
- Le taux de faux positifs (FPR pour False Positives Rates).

Concernant **la caractéristique directionnelle**, on compare la de différence d'angle (en degrés) entre la vraie direction et la direction donnée par le filtre testé. La moyenne est la déviation standard de ces deux valeurs constitue le critère d'évaluation. L'erreur maximum entre deux angles est de 90°.

Sachant que des images publiques concernant les aspects curvilignes sont assez rares, les chercheurs ont dû utiliser des données synthétiques produites par le logiciel *VascuSynth*.

F. Résultats

Concernant la caractéristique d'intensité :

En exploitant les différentes données des résultats, on remarque que le RORPO a généralement mieux marché pour la détection des structures curvilignes. Il détecte en effet bien moins de faux positifs que les autres méthodes, en gardant un taux de vrais positifs élevé.

C'est seulement à partir d'un certain niveau de bruit gaussien que d'autres techniques comme *l'OOF* performe mieux. Cependant, les chercheurs indiquent que ce niveau de bruit n'est pas représentatif du bruit qui serait présent sur de réelles images.

Les résultats sur la caractéristique d'intensité sera globalement un peu meilleure avec le RORPO 13 que le RORPO 7 mais avec un temps d'exécution double.

Concernant la caractéristique directionnelle :

Étant donné que la méthode *HDCS* ne dispose pas de caractéristique directionnelle, on a cette fois-ci seulement pu comparer le RORPO 7, le RORPO13 avec *l'OOF* et le *FV*.

Les meilleures directions sont données par le RORPO 7 sur à peu près tous les niveaux de bruit gaussien. Cela démontre la caractéristique principale du RORPO qui utilise des opérateurs connexes, mieux adaptés pour la détection de structures curvilignes dans un environnement anisotrope à l'inverse de *l'OOF* et du *FV*.

On s'aperçoit qu'augmenter le nombre d'orientations avec le RORPO 13 n'améliore pas forcément les résultats : l'échelle choisie doit être plus grosse pour le RORPO 13 ce qui veut dire un chemin plus long mais qui induit une adaptation locale moins bonne pour les structures curvilignes.

G. Opinion personnelle

Le sujet est très intéressant et montre à quel point on peut utiliser le traitement d'images pour mener des projets de recherche poussés, on l'a spécialement pris pour ce critère particulier.

L'article est assez compliqué à lire au premier abord assez long avec un riche vocabulaire développé. Compliqué à comprendre même pour un natif anglais, étant donné qu'il introduit de nombreux termes avec un vocabulaire propre à la recherche et au traitement d'images, on a dû faire des recherches approfondies pour mieux comprendre de nombreux mots.

Avec une comparaison directe avec les autres méthodes, le rapport est pertinent, montrant directement les avantages et les inconvénients de la méthode comparé aux autres filtres. Le RORPO se place en tant qu'alternative forte par rapport aux filtres existants, en apportant des nouvelles caractéristiques essentielles pour la détection de structures curvilignes.

III. Étude de l'implémentation

Le code source fourni en accompagnement à l'article du projet est composé de sept fichiers, contenant les scripts présentés dans l'article. La documentation du code et les instructions d'utilisation du programme permettent de reproduire les trois exemples détaillés dans l'article (cf. Partie 3 : Expérimentations). Les expérimentations illustrent le fonctionnement du programme, ce qui met en avant son utilité dans ses diverses applications.

En termes de structuration, le code est présenté de manière claire et bien documenté ce qui permet de rapidement se familiariser avec pour l'utiliser.

Le projet est divisé en plusieurs fichiers contenant chacun une des étapes nécessaires à la détection de SC avec la méthode RORPO. Pour exécuter les scripts, il faut builder le projet avec CMake, qui permet de configurer et générer le makefile, ce qui permet ensuite d'exécuter le programme RORPO. Puisque le code est bien structuré, il est facilement réutilisable dans d'autres contextes. Par exemple, les algorithmes de tri et d'opérations basiques sont assez universels donc peuvent être réutilisés dans d'autres projets. Si on considère le projet dans sa globalité, il est très spécifique donc peuvent difficilement servir à autre chose que la détection de SC, mais il peut être utilisé en complément à d'autres scripts.

Nous avons relevé un point qu'il serait bien d'améliorer dans une prochaine version, concernant le type d'image qui doit être utilisé. En effet, en l'état actuel, le programme impose d'utiliser une image avec des SC claires sur fond sombre. Il pourrait être bien d'ajouter un algorithme pour coloriser les images de manière que le sujet soit clair et l'arrière-plan sombre, ce qui éviterait une manipulation supplémentaire en dehors des interfaces du programme.

L'article présente six algorithmes en pseudocode, que nous retrouvons dans le répertoire du code source du projet RORPO. Ceci permet de pouvoir retracer les étapes décrites dans l'article en parcourant les chaque scripts correspondant à un de ces algorithmes. Les scripts sont l'application directe des pseudo codes en C++ ou en python pour la représentation directionnelle des SC. En dehors des scripts, le code a permis de reproduire toutes les expérimentations de l'article, ce qui montre aussi sa fiabilité.

IV. Expérimentations

L'article présente trois exemples d'utilisation de RORPO, que vous avons décidé de reproduire, de sorte à tester la fiabilité du programme. La détection de SC se fait grâce au programme en C++ et la visualisation de la feature directionnelle se fait grâce au script python *plot_directional_feature_reviewer.py* qui affiche les flèches directionnelles sur l'image étudiée.

Les trois images de tests représentent une photographie de rétine, une photographie de fissures et une image de SC artificielle. Elles illustrent bien le potentiel du programme RORPO, puisqu'elles présentent des SC différentes.

Les images doivent avoir un fond sombre et des objets clairs, si le cas inverse est présent il faut d'abord inverser les couleurs.

Le programme RORPO2D utilise au total six paramètres, qui sont les suivants :

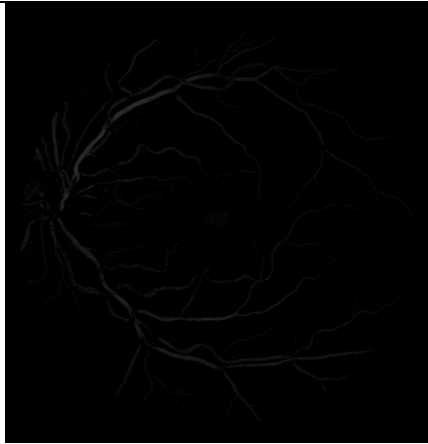
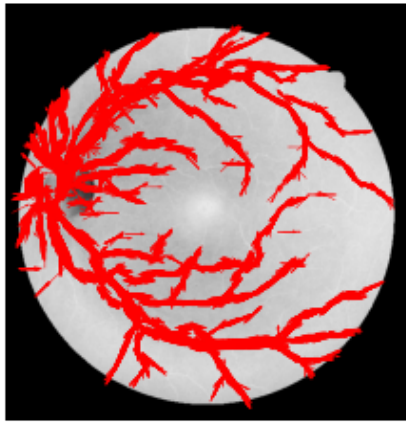

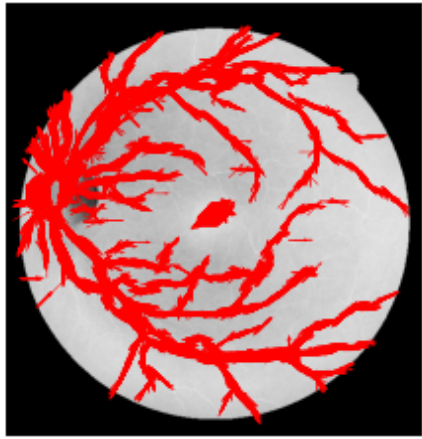
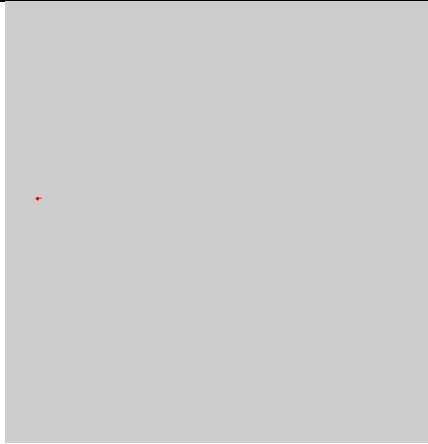
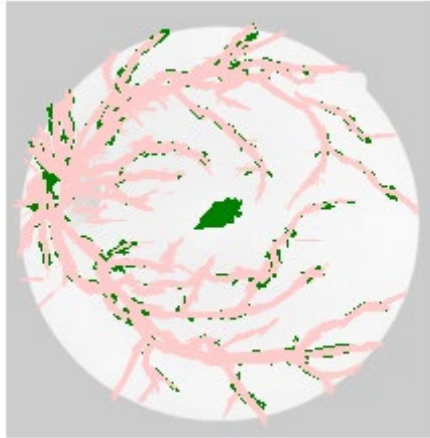
- PARAM 1 : le chemin de l'image qu'on souhaite étudier.
- PARAM 2 : la longueur minimale du chemin
- PARAM 3 : le facteur d'échelle
- PARAM 4 : le nombre d'échelles
- PARAM 5 : la robustesse
- PARAM 6 : le chemin de l'image de sortie

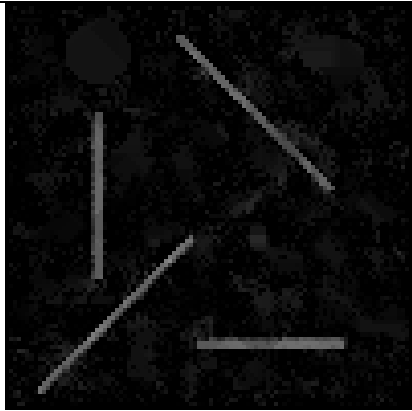
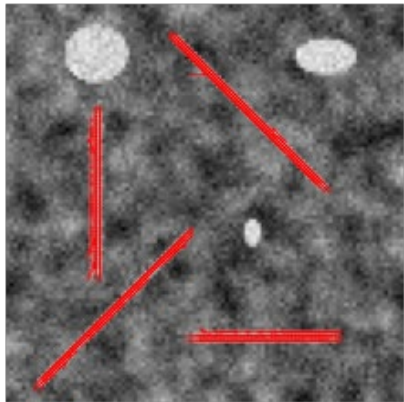
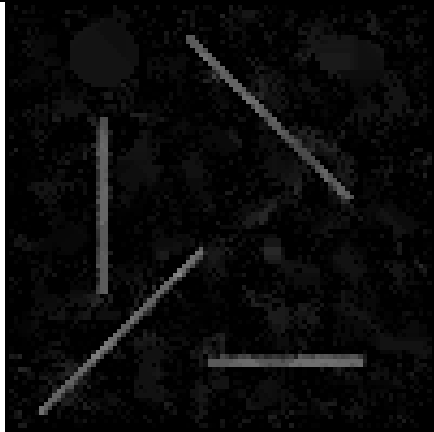
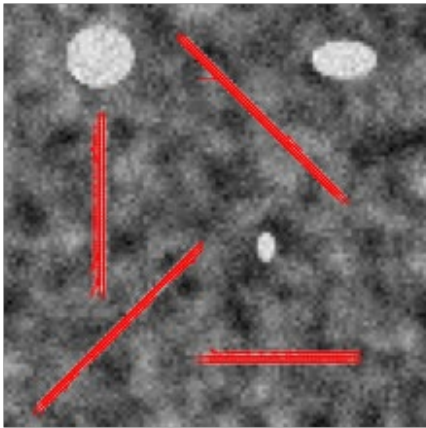
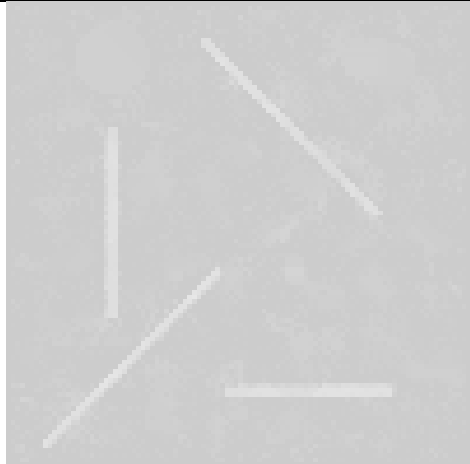
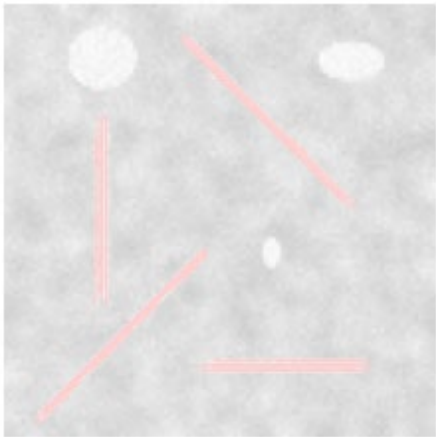
Pour notre première expérience, nous avons choisi de tester le programme avec l'image de la rétine pour arriver à reproduire l'image montrée dans le rapport. Pour ce faire nous avons utilisé la commande ***./RORPO2D ../images/retinal_image.png 25 1.5 3 0 ../results/test_output_retinal.png*** après la génération de l'exécutable du projet.

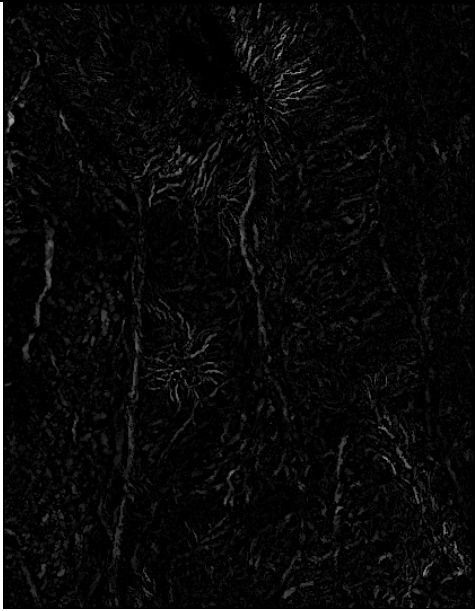
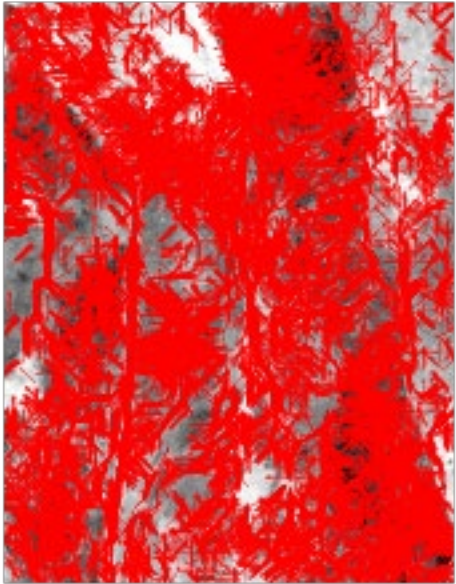
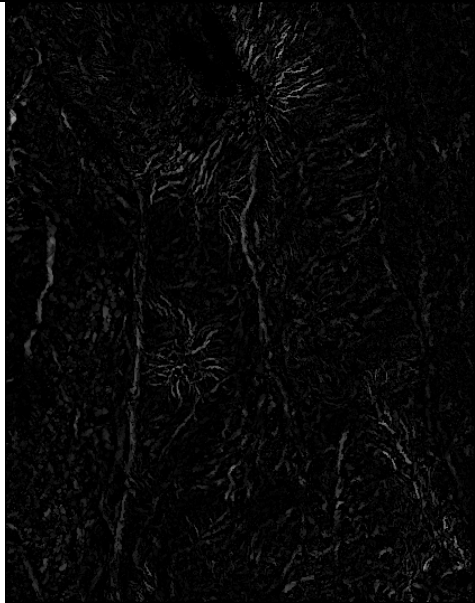
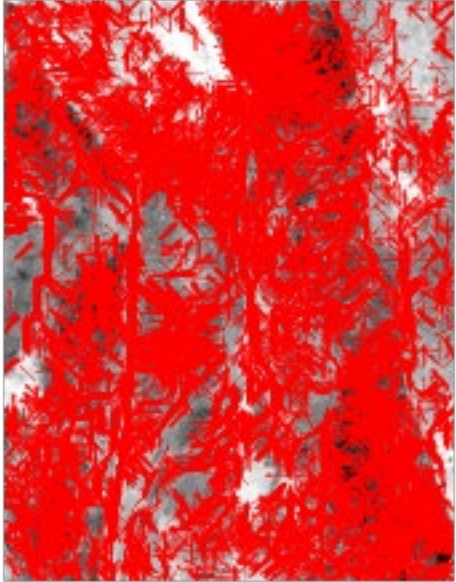
Les deux autres images ont été traitées de manière analogue.



Le choix des paramètres 2 à 5 dépend du type d'image qu'on analyse. Ceux-ci peuvent être compliqué à déterminer, puisqu'ils varient en fonction de l'objet d'étude. Dans le cadre de nos expériences vous avons repris les valeurs recommandées dans l'article pour obtenir des résultats semblables aux résultats attendus. On trouve les bonnes valeurs de paramètres dans le README du programme, ce qui nous permet d'obtenir les résultats ci-dessous.

Les tableaux ci-dessous comparent les résultats présentés dans l'article et les résultats de nos expériences.

Rétine	Intensité	Direction
Article		
Code		
Différence		

Artificielle	Intensité	Direction
Article		
Code		
Différence		

Fissures	Intensité	Direction
Article		
Code		

Différence		
------------	---	--

Les trois tableaux illustrent bien le fonctionnement du programme RORPO. Comme nous pouvons le voir, celui-ci est plutôt fiable, même si on a quelques différences lorsqu'on étudie des SC plus complexes. Seule l'image artificielle ne présente aucune erreur, ce qui montre que le programme fonctionne mieux pour des images simples. Dans les trois exemples, la feature d'intensité présente très peu de différences (seulement quelques pixels pour la rétine) entre les résultats attendus et obtenus. La feature directionnelle présentent plus d'écarts, mais cela ne perturbe pas l'analyse de l'image. Une dernière observation concerne la différenciation de SC et de tâche. Sur l'image de la rétine, on voit que le résultat attendu ignore la tâche centrale, tandis que le résultat montre que le programme n'a pas su faire la différence entre cette tâche de surexposition et les SC claires autour. Il faudrait mener des tests complémentaires pour en déceler la cause.

Nous n'avons pas eu le temps de mettre en œuvre des expérimentations complémentaires, mais nous avons élaboré une liste d'éléments que nous trouvons importants à tester pour tester la robustesse de cette technique et pour mettre en avant les limites imposées par l'algorithme. Comme les expériences l'ont démontré, le programme RORPO est plutôt fiable, mais n'est parfait que pour une image artificielle. Nous proposons de tester plus d'images de ce type de sort à identifier les facteurs qui font qu'on obtient des résultats différents en fonction de la complexité de l'image. Ces tests permettraient de comprendre d'où proviennent les erreurs ainsi que d'identifier de potentielles points d'amélioration pour le traitement des images dans le programme.

V. Sources

Espace échelle : [Espace d'échelle — Wikipédia \(wikipedia.org\)](https://fr.wikipedia.org/wiki/Espace_d%27%C3%A9chelle)

Tortuosity : [Tortuosity - Wikipedia](https://en.wikipedia.org/wiki/Tortuosity)

Curvilinear Structure Analysis by Ranking the orientation Responses of Path Operators : [RORPO \(esiee.fr\)](https://esiee.fr/~lorpo/).

Outil de comparaison d'images : <https://online-image-comparison.com/>