

# Projet EPITA

## Recalage d'image de fond d'oeil

Thomas Benseghir

21 nov 2021

### Résumé

Ce projet a valeur d'évaluation pour le cours sur le recalage d'images médicales. Il sera évalué par groupe de deux élèves, devant rendre un rapport commun illustré accompagné de code python. Code et rapport compteront à parts égales pour la note finale. Le tout est à envoyer avant le 31 janvier à l'adresse suivante : [thomas.benseghir@ge.com](mailto:thomas.benseghir@ge.com)

A travers ce projet vous aborderez deux thèmes importants de l'imagerie médicale : le recalage et la segmentation. Il vous faudra aborder les problèmes comme s'ils vous étaient posés lors de votre prochain poste d'ingénieur de recherche. Le but n'est pas ici de vous faire partir de zéro et de vous faire réinventer la roue, mais au contraire de vous faire utiliser les ressources à votre disposition pour résoudre un problème concret avec le minimum d'efforts.

## 1 Introduction et vue d'ensemble

Qu'ont en commun les photos de nuit prises sur smartphone et le diagnostic du cancer du foie sur scanner multiphasés ? A priori, peu de choses. L'une d'entre elles est le recalage. Que ce soit pour accumuler les images peu exposées même en cas de mouvement du photographe, ou pour fusionner des scanners pris à des moments différents, une étape de recalage est nécessaire afin de compenser le mouvement entre chaque prise.

Le recalage d'un objet source  $X$  sur un objet référence  $Y$  consiste à trouver une transformation  $\bar{T}$  telle que  $\bar{T}(X)$  et  $Y$  se correspondent au mieux. L'évaluation de cette correspondance est quantifiée par une fonction  $\mathcal{D}(.,.)$  dite de distance (ou de dissimilarité) que l'on cherche donc à minimiser pour obtenir  $\bar{T}$  :

$$\bar{T} = \operatorname{argmin}_{T \in \Omega} \mathcal{D}(T(X), Y) \quad (1)$$

avec  $\Omega$  l'ensemble des transformations admissibles. Vous pourrez trouver dans [9] une revue de la littérature sur le sujet décrivant différentes fonctions de distance (basées directement sur les images ou sur des structures segmentées), différents types de déformations (rigide, affine, spline, difféomorphique...) ainsi que différents moyens de minimiser l'équation (1).

A travers ce projet, vous apprendrez à effectuer une tâche de recalage entre deux images par le biais d'extraction de structures d'intérêt. Il est organisé en trois parties :

- La construction d'une petite bibliothèque de recalage par ICP (Iterative Closest Point) incluant les fonctions de recalage et les fonctions pour les tester.
- La segmentation de vaisseaux dans des images de fond d'oeil afin d'obtenir un ensemble de lignes centrales à recaler.
- La mise bout à bout des deux précédentes parties pour construire un algorithme de recalage d'images de fond de l'oeil.

Les parties 2 et 3 sont indépendantes l'une de l'autre et pourront être séparées au sein d'un binôme. Pour la troisième, il vous sera demandé d'avoir un regard critique sur cette solution face au problème clinique et de proposer des voies d'amélioration.

Dans la suite, vous trouverez de nombreuses parties optionnelles, notées [OPT], qui sont là pour ceux qui souhaitent approfondir ce cours de recalage et mieux comprendre les méthodes de l'état de l'art en les

implémentant. Ces parties ne vous pénaliseront en rien dans la notation et pourront même donner lieu à des points supplémentaires pour la partie 4. Les parties 2.1 et 3.1 sont à préparer entre les deux sessions de cours (en se les répartissant au sein d'un binôme). Elles ont pour but de vous mettre face à un problème pour lever des difficultés et des questions dont nous pourrions discuter ensemble durant la seconde partie du cours.

## 2 Recalage de nuages de points par ICP

L'algorithme de l'ICP, introduit dans [2], présente de nombreux avantages (intuitif, générique, efficace) qui expliquent son adoption dans des domaines variés. Plusieurs raffinements et variantes ont été proposés ensuite, dont bon nombre sont répertoriés dans [8], mais le principe reste toujours le même :

1. Construction des appariements point-à-point avec leur plus proche voisin pour les ensembles  $X$  et  $Y$ .
2. A appariement fixé, trouver la transformation minimisant la somme des distances au carré entre points appariés.
3. On applique la transformation à l'ensemble source  $X$ , puis on revient au point 1 jusqu'à convergence.

Dans la suite, on considérera que les ensembles  $X$  (source) et  $Y$  (références) sont composés de points dans  $\mathbb{R}^n$ , avec  $n$  arbitraire étant la dimension de l'espace (e.g.  $n = 2$  pour les images,  $n = 3$  pour les volumes et les vidéos...).

### 2.1 [OPT] Implémentation de l'algorithme ICP

Quoi de mieux pour bien comprendre un algorithme que de le recoder ? C'est ce que je vous propose ici.

OPT Coder la fonction `nearest_neighbor(X,Y)` qui renvoie la liste des appariements entre une source  $X$  et une target  $Y$ . Pour chaque point  $X[i]$  de  $X$ , on cherche le point  $Y[j]$  le plus proche dans  $Y$ . On ajoute le couple  $(X[i], Y[j])$  à la liste des appariements.

OPT Coder la fonction `EstimateRigidTransformation(Pairings)` qui la transformation rigide minimisant la distance entre deux ensembles de points appariés `Pairings` ( construit grâce à la fonction `nearest_neighbor(X,Y)` ) en se basant sur [4].

OPT Implémenter l'algorithme ICP en alternant appariement, estimation de la transformation et application de la transformation à l'ensemble  $X$ .

### 2.2 Test de l'algorithme ICP

Que vous ayez implémenté l'algorithme ICP vous-même, ou que vous l'ayez récupéré de [5], il serait bon de tester si cette implémentation est correcte. Tester est une tâche cruciale en entreprise afin de gagner en confiance sur une implémentation, gagner en intuition et permettre un contrôle de l'évolution d'une application. Pour ce faire, je vous propose d'implémenter les fonctions de test suivantes :

1. Créer une fonction `randomTransformation2D(maxRot=30, maxTrans=1)` qui génère une transformation rigide aléatoire de rotation centrée à l'origine du repère et d'angle  $\theta \in [-\text{maxRot}; \text{maxRot}]$  d'amplitude de translation (norme du vecteur translation) inférieure à `maxTrans`.
2. Créer une fonction `randomTransformation3D` sur le même modèle que précédemment.
3. Créer un carré (2D) source centré en  $(0,0)$  composé de 10 points sur chacun de ses côtés. Dupliquez le en un carré target et lui appliquer une transformation rigide aléatoire (rotation de moins de 30 degrés).
4. Appliquer la fonction `nearest_neighbor` entre les deux précédents carrés et proposez une manière de visualiser les appariements. Commenter.
5. Appliquer l'algorithme ICP et visualiser l'évolution au cours des itérations de la position du carré source et ses appariements au carré target. Commenter
6. Générer des points aléatoirement sur les côtés d'un carré puis faire de même sur les faces d'un cube.

OPT Étendre cette implémentation à la génération d'un hypercube de n'importe quelle dimension <sup>1</sup>.

7. Générer deux cubes grâce au point 4, en transformer un avec une transformation aléatoire puis effectuer le recalage entre les deux objets. Qu'observez vous quant au comportement de l'algorithme par rapport au bruit.

## 2.3 Variantes de l'algorithme ICP

Dans la section précédente, vous avez utilisé une implémentation simple de l'algorithme ICP. Elle ne vous permet notamment de ne recalculer que des objets ayant le même nombre de points (ce qui en pratique est très limitant). De nombreuses variantes de l'ICP existent dans la littérature : gestion d'outliers, appariements multiples, transformations non-rigides... Mais ici nous nous contenterons de relâcher la contrainte sur le nombre de points dont sont composés les structures.

1. Générer un cube avec une face manquante  $C_1$  ainsi qu'un cube complet  $C_2$ . Appliquer une transformation aléatoire à  $C_2$ .
2. Implémenter une variante de l'ICP permettant de recalculer un objet A inclus (au sens ensembliste) dans un objet B sans avoir à savoir qui est la source et qui est la target. Tester votre implémentation sur l'exemple créé en 2.2.4.
3. Effectuer un recalage de  $C_1$  (source) vers  $C_2$  (target). Faites de même de  $C_2$  vers  $C_1$ . Qu'observez vous ?

## 3 Segmentation de vaisseau du fond de l'oeil

La segmentation de structures d'intérêt est un problème fréquent en imagerie médicale. Ses applications sont nombreuses : segmentation du foie en CT servant à déterminer le rapport entre volume tumoral et volume sain pour planifier une intervention ; segmentation d'objets métalliques pour améliorer la qualité image d'une reconstruction ; ou la segmentation de vaisseaux de fond d'oeil pour fusionner deux images entre elles.

La segmentation consiste à obtenir un ensemble d'objets d'intérêt à partir d'une image (d'une vidéo, d'un volume...). La représentation de ces objets peut prendre différentes formes que nous illustrons ici pour la segmentation de vaisseaux :

- Image binaire indiquant pour chaque pixel s'il appartient à un vaisseau ou non. Cette représentation est fidèle à l'image mais est peu pratique à utiliser car elle est lourde en mémoire et ne se prête pas bien au recalage.
- Un ensemble de courbes 2D représentant le centre des vaisseaux (appelées lignes centrales) ainsi que le rayon du vaisseau en chaque point de cette ligne. Cette représentation est en général un bon compromis entre fidélité de la représentation et facilité d'utilisation.
- Un graphe 2D dans lequel les sommets sont des bifurcations et les edges sont des lignes centrales. Cette représentation est à la fois fidèle et légère mais est souvent plus difficile à obtenir.

### 3.1 [OPT] Segmentation de vaisseaux par deep learning

Depuis quelques années, segmentation et deep learning vont souvent de pair. En effet les résultats sont souvent impressionnants et plutôt faciles à reproduire. Si on dispose d'une base de données de qualité bien sûr.

OPT Identifier les types de réseaux candidats pour la segmentation de vaisseaux. Choisir celui qui vous paraît le plus pertinent et expliquer votre choix.

OPT Mettre en place une segmentation de vaisseaux par deep learning et le tester sur une sous partie de la base de données disponible en [1].

OPT Comparer vos résultats à ceux de [10].

OPT Appliquer votre réseau sur le jeu de données de recalage [7]. Que constatez-vous ?

---

1. N.B. Un point  $x$  appartient à une face d'un hypercube centré en  $(0,0,0)$  de côté 1 si et seulement si  $\|x\|_\infty = 0.5$

### 3.2 Segmentation de vaisseaux par méthode conventionnelle

Avant l'ère du deep learning, existait des filtres de réhaussement créés de toutes pièces par l'intuition et un peu de mathématiques. Je vous propose de replonger dans cette époque.

1. Implémenter un filtre de "vesselness" comme décrit dans [6]. Vous pouvez vous aider de ressource que vous trouverez sur le web.
2. Régler (manuellement ou non) les échelles du filtre de vesselness pour que sa réponse soit optimale pour les images de la base de données [1].
3. Définir une procédure de seuillage pour avoir le meilleur résultat sur la base de données [1]. (Vos cours et internet seront certainement d'une grande aide)

### 3.3 Segmentation de lignes centrales

1. A partir d'une image segmentée provenant de [1], extraire un ensemble de lignes centrales correspondant à ces structures segmentées.
  2. En vous aidant de [3], représenter ces lignes centrales par un graph de lignes centrales connectées entre-elles.
  3. Appliquer les deux étapes précédentes à la sortie de votre segmentation de vaisseaux par filtre "vesselness et seuillage". Quels sont les défauts que vous observez ?
- OPT Proposer (et pourquoi pas implémenter) une ou plusieurs méthodes pour améliorer la qualité de votre graph de lignes centrales.

## 4 [OPT] Recalage de vaisseaux de fond de l'oeil

Cette section est faite pour les passionnés et ceux qui ne supportent pas un travail inachevé. Elle a pour but d'illustrer les applications du recalage. Grâce aux deux sections précédentes et étant données 2 images, vous êtes désormais capables de segmenter un ensemble de lignes centrales dans chacune d'elles et de recaler ces deux ensembles entre eux. La transformation obtenue à convergence de la procédure de recalage a pour but d'aligner les deux ensembles de points donnés en paramètres. Mais appliquée à l'image elle va vous permettre plusieurs applications utiles en imagerie médicale :

1. **Mosaïcking** : en recalant deux images ayant une partie en commun on peut agrandir artificiellement le champs de vue. C'est la base des applications "panorama" sur téléphone. Choisir une paire d'image de la base de données [7] ayant un préfixe "P" (par exemple "P04\_1.jpg" et "P04\_2.jpg") et effectuer les tâches suivantes :
  - (a) Effectuez un recalage entre ces deux images en vous servant de la vérité terrain. N.B. identifiez bien si votre recalage aligne l'image 1 sur l'image 2 ou bien l'image 2 sur l'image 1.
  - (b) Copier chaque image au centre d'une image 9 fois plus grande (de dimensions  $(3 \times \dim_x, 3 \times \dim_y)$ ) en complétant autour par du noir.
  - (c) Appliquer la transformation à l'image agrandie à recaler. Vérifiez que cette image transformée est bien recalée avec l'autre image agrandie.
  - (d) Construire une image mosaïque à partir des deux images précédentes.
2. **Suivi temporel** : le suivi de l'évolution d'une pathologie au cours du temps est au coeur des décisions cliniques pour l'établissement d'un traitement. Que ce soit pour mesurer l'évolution d'une tumeur cancéreuse, établir si un traitement est efficace ou non, ou bien pour vérifier qu'une zone suspecte n'évolue pas ; le recalage d'image est un outil important pour faciliter la tâche du clinicien. En effet, lorsque l'on doit comparer deux examens d'un même patient effectués à deux dates différentes, il faut aligner (au moins mentalement) les deux images. Afin de faciliter la tâche du clinicien et d'ouvrir la porte à l'analyse automatique de pathologie, il faut passer par une étape de recalage. Choisissez une paire d'image de la base de données [7] ayant un préfixe "A" (par exemple "A10\_1.jpg" et "A10\_2.jpg") et effectuez les tâches suivantes :
  - (a) Effectuez un recalage automatique grâce aux parties 2 et 3.

- (b) Proposez un mode de fusion d'images recalées permettant au clinicien d'évaluer plus facilement l'évolution de la pathologie du fond de l'oeil.
- (c) Proposez un mode de visualisation pour évaluer l'apport de votre recalage (e.g. fusion avant recalage, fusion après recalage).

## Références

- [1] Thomas Köhler et AL. *High-Resolution Fundus (HRF) Image Database*. URL : <https://www5.cs.fau.de/research/data/fundus-images/>.
- [2] P. J. BESL et N. D. MCKAY. « A method for registration of 3-D shapes ». In : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14.2 (1992), p. 239-256.
- [3] Yan xiaolong et CHARLES N. CHRISTENSEN. *GitHub Skeleton Network*. URL : <https://github.com/Image-Py/sknw>.
- [4] David W EGGERT, Adele LORUSSO et Robert B FISHER. « Estimating 3-D rigid body transformations : a comparison of four major algorithms ». In : *Machine vision and applications* 9.5-6 (1997), p. 272-290.
- [5] Clay FLANNIGAN. *GitHub ICP*. URL : <https://github.com/ClayFlannigan/icp>.
- [6] Alejandro F FRANGI et al. « Multiscale vessel enhancement filtering ». In : *International conference on medical image computing and computer-assisted intervention*. Springer. 1998, p. 130-137.
- [7] Carlos HERNANDEZ-MATAS et al. « FIRE : fundus image registration dataset ». In : *Journal for Modeling in Ophthalmology* 1.4 (2017), p. 16-28.
- [8] Szymon RUSINKIEWICZ et Marc LEVOY. « Efficient variants of the ICP algorithm ». In : *Proceedings third international conference on 3-D digital imaging and modeling*. IEEE. 2001, p. 145-152.
- [9] A. SOTIRAS, C. DAVATZIKOS et N. PARAGIOS. « Deformable Medical Image Registration : A Survey ». In : *IEEE Transactions on Medical Imaging* 32.7 (2013), p. 1153-1190.
- [10] Juntang ZHUANG. *LadderNet : Multi-path networks based on U-Net for medical image segmentation*. 2019. arXiv : 1810.07810 [cs.CV]. URL : <https://github.com/juntang-zhuang/LadderNet>.