Contributions au recalage pour la réalité augmentée en coelioscopie de l'utérus : détection de contours sémantiques et mise à jour topologique du modèle virtuel à partir d'images peropératoires

Tom François

▶ **To cite this version:**

Tom François. Contributions au recalage pour la réalité augmentée en coelioscopie de l'utérus : détection de contours sémantiques et mise à jour topologique du modèle virtuel à partir d'images peropératoires. Synthèse d'image et réalité virtuelle [cs.GR]. Université Clermont Auvergne, 2021. Français. NNT : 2021UCFAC112 . tel-03886226

HAL Id: tel-03886226
https://theses.hal.science/tel-03886226

Submitted on 6 Dec 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Contributions au recalage pour la réalité augmentée en cœlioscopie de l'utérus : Détection de contours sémantiques et mise à jour topologique du modèle virtuel à partir d'images peropératoires

Tom François

Examinateur :	Éric Marchand, Professeur des Universités, Université Rennes, Inria
Rapporteuse :	Sylvie Chambon , Maîtresse de Conférences, Toulouse INP, IRIT
Rapporteur :	Fabrice Mériaudeau, Professeur des Universités, Université de Bourgogne, IFTIM
Directeur de thèse :	Adrien Bartoli, Professeur des Universités, Université Clermont Auvergne, Institut Pascal
Co-encadrant :	Lilian Calvet, Chercheur post-doctoral, Toulouse INP
Co-encadrant :	Damien Saboul, Ingénieur recherche, Be-Ys Research

Thèse de Doctorat

Soutenance le 10/12/2021
Présentée à l'Université Clermont Auvergne
pour l'obtention du grade de Docteur
(Décret du 5 juillet 1984)

École doctorale : Sciences Pour l'Ingénieur
Spécialité : Informatique

Résumé

La réalité augmentée (RA) consiste à superposer des éléments virtuels à une image, de manière à créer l'illusion que ces éléments coexistent dans la scène réelle. La cœlioscopie est une technique de chirurgie minimalement invasive qui permet de réaliser des interventions chirurgicales via de petites incisions permettant le passage d'outils et d'une caméra. C'est une technique particulièrement utilisée en gynécologie car elle apporte des avantages conséquents, limitant le traumatisme pour la patiente. En revanche, ce type d'intervention est plus complexe pour le/la chirurgien(ne), qui perçoit mal la profondeur et dont les gestes sont contraints par les incisions. Utiliser la RA pour assister le geste chirurgical en cœlioscopie est donc particulièrement pertinent. Cette motivation a donné lieu au développement dans l'équipe EnCoV d'une première solution appelée Uteraug, qui réalise le recalage entre les données préopératoires et le flux vidéo cœlioscopique. Concrètement, avec cet outil, la RA crée un effet de transparence virtuelle de l'organe pour permettre au chirurgien de visualiser les tumeurs à extraire, par exemple dans le cadre de myomectomies. Cette solution a néanmoins plusieurs contraintes qui rendent son utilisation limitée. D'une part, les étapes préliminaires de mise en place de la RA sont chronophages car elles nécessitent des annotations manuelles à réaliser directement en salle d'opération. D'autre part, cette solution limite la RA à la phase préparatoire de l'opération. En effet, une fois que l'organe est incisé, le système de recalage ne fonctionne plus. Nos travaux ont visé à répondre à ces différentes limites. Dans un premier temps, nous nous sommes intéressés à la mise en place de la RA, une étape préliminaire qui nécessite l'annotation des contours occultants de l'utérus. Les contours occultants sont les contours visibles de la silhouette d'un objet. Ils sont ici utilisés pour contraindre le recalage non rigide entre les données préopératoires et peropératoires. Nous avons étudié la mise en place d'une stratégie pour extraire ces contours de manière automatique grâce à un réseau de neurones profond. Nous avons également proposé un nouveau score pour comparer deux contours échantillonnés sur une image. Nous avons réalisé une étude sur dix opérations pour lesquelles les données ont été enregistrées pour simuler l'utilisation du logiciel après l'opération. Nous avons montré que l'annotation automatique proposée permet d'obtenir une précision très proche de celle obtenue avec une annotation manuelle tout en réduisant considérablement le temps nécessaire pour obtenir la RA. Dans un second temps, nous avons proposé une séquence d'étapes pour détecter l'incision de l'utérus dans le flux vidéo et mettre à jour le modèle virtuel peropératoire, permettant ainsi de maintenir la RA au cours de l'opération. Nous avons testé cette séquence d'étapes sur des données ex-vivo et in-vivo et réussit à maintenir la RA de manière cohérente avec l'incision. L'implémentation actuelle de cette séquence d'étapes ne permet pas un usage en temps réel.

Abstract

Augmented reality (AR) consists in superimposing virtual elements on an image in order to create the illusion that these elements coexist with the real scene. Laparoscopy is a minimally invasive surgical technique that allows surgical interventions to be performed through small incisions that allow the passage of tools and a camera. It is a technique particularly used in gynecology because it brings significant advantages, limiting the trauma for the patient. However, this type of surgery is more complex for the surgeon, who has poor depth perception and whose movements are constrained by the entry points. Using AR to assist the surgical gesture in laparoscopy is therefore particularly relevant. This motivation led to the development of a first solution called Uteraug in the EnCoV team, which performs the registration between the preoperative data and the laparoscopic video stream. In concrete terms, with this tool, AR creates a virtual transparency effect of the organ to allow the surgeon to visualize the tumors to be extracted, for example in the context of myomectomies. However, this solution has several constraints that limit its use. On the one hand, the preliminary steps of setting up the AR are time consuming because they require manual annotations to be made directly in the operating room. On the other hand, this solution limits AR to the preparatory phase of the operation. Indeed, once the organ is incised, the registration system does not work anymore. Our work aimed at addressing these different limitations. Firstly, we focused on the AR setup, a preliminary step that requires the annotation of the occluding contours of the uterus. The occluding contours are the visible contours of the silhouette of an object. They are used here to constrain the non-rigid registration between the preoperative and intraoperative data. We have studied the implementation of a strategy to extract these contours automatically using a deep neural network. We also proposed a new score to compare two image contours. We performed a study on ten surgeries for which the data were recorded to simulate the use of the software after the surgery. We have shown that the proposed automatic annotation allows the system to obtain an accuracy very close to the one obtained with a manual annotation while reducing considerably the time needed to obtain the AR. In a second step, we proposed a pipeline to detect the incision of the uterus in the video stream and update the intraoperative virtual model, thus allowing to maintain the AR during the operation. We tested this pipeline on ex-vivo and in-vivo data and managed to maintain AR consistency with the incision. The current implementation of this pipeline does not run in real-time.

Remerciements

Les travaux que j'ai menés pour la réalisation de ce manuscrit et des travaux qui en découlent sont le fruit de la collaboration de nombreuses personnes que j'aimerais ici remercier.

Je tiens à remercier en premier lieu les membres de jury, pour avoir accepté d'entendre et d'évaluer mon travail de thèse. J'ai longtemps attendu la soutenance de cette thèse qui me permet de conclure sur les travaux menés et de récompenser mes efforts. J'adresse ainsi mes sincères remerciements à Éric Marchand pour avoir présidé ma soutenance, Fabrice Mériaudeau et Sylvie Chambon pour vos relectures.

Adrien, je te remercie pour m'avoir proposé cette thèse au moment où l'aventure de création d'entreprise tombait à l'eau. Cette nouvelle aventure m'a demandé de redoubler d'efforts et je te remercie de toujours avoir été là pour les soutenir et pour orienter mes recherches.

Lilian, ces quelques années de travail en ta compagnie m'ont beaucoup plu. J'ai appris grâce à toi que la nature d'un homme peut-être complexe et simple à la fois. Tu m'as donné un bel exemple de relation professionnelle saine avec sincérité et dévouement.

Damien, tu seras mon premier exemple de collaboration à distance. Prendre du recul sur mon travail pour te le partager et répondre à tes questions m'ont apporté énormément. Je suis très content d'avoir pu partager ces quelques années de travail et j'espère pouvoir continuer ainsi.

Le travail que j'ai pu présenté ne serait rien sans tous les médecins qui m'ont accompagnés pour collecter, conseiller, proposer, annoter. Un grand merci à Sabrina, Claire, Callyane, Guillaume et Nicolas pour leur aide. Mention spéciale à toi, Sabrina, pour avoir fourni tous ces efforts pour appréhender le monde étrange de l'informatique et de la vision par ordinateur.

J'ai passé les 7 dernières années de ma vie à travailler dans les locaux de l'équipe EnCoV et je dois à toutes les personnes que j'ai croisées d'avoir contribué à me rendre moins enclin à la folie. Merci à Prasad, Simone, Alexis, Robin, Émilie, Laurent, Shaifali, Ajad, Mathias, Bastien, François, Isabelle, Damien, Carlos, et plein d'autres.

Yamid, tu as été un excellent compagnon pendant ma thèse, je te souhaite beaucoup de réussite pour la suite de ta carrière scientifique.

Richard, merci de m'avoir montré la voie et pour tes petites questions de 9h du matin.

Camille, tellement généreux avec les autres et dur avec toi-même, je garde un souvenir intarissable de cette nuit blanche pour MICCAI.

Je ne me serais sûrement pas sorti de cette thèse si je n'avais pas pu pas compter sur mes amis en dehors du travail. Petite dédicace aux Tourne-Disc de Clermont-

Ferrand, après avoir passé une semaine sur un problème complexe, courir après un frisbee comme un chien a quelque chose de cathartique. Merci à Cava, Gaël, Maxime et Marine, Manal, Nathan, Sordide et Oucha, Mario, Tek, Charlotte, Steven, Pierre-Em, Sylvain, Lucas, Pipo.

Je pense également aux amis vétos d'Alfort, Dodo, Prenant, Thorel, pour m'avoir partager l'art de la communion, particulier à leur savoir-faire, entremêlant les cinq sens, pour des soirées loin des problèmes informatiques.

Merci également aux collègues de la clinique vétérinaire du Val de la Dore : Éléna et Adélaïde tout particulièrement, pour nos moments de détentes avec la team libre.

J'en arrive à mes amis d'école d'ingénieur avec qui j'ai gardé des contacts grâce à nos passions pour l'Ultimate, le jeux vidéo et le plaisir de s'engueuler.

Cyril, mon dernier binôme de TP, la rencontre de la province et du 93, j'admire toujours ce que tu es capable de faire quand tu es motivé, pour le travail comme pour tes voyages.
Théo, le djeun's de l'équipe, ta sensibilité n'a d'égal que ta capacité à faire le pitre. Tu aimes être ce personnage qui peut être à la fois le plus lourd d'une pièce pour ensuite aborder les sujets les plus profonds.
Kathleen, tu respires ce calme et cette sérénité qui donne envie de croire en tout.
Hugues, j'ai dû mal à t'imaginer sans sourire. Ça doit être parce que tu es toujours là pour raconter des conneries.
Émile, mon premier coloc et compagnon de la duo bot, je t'en ai fait un peu baver pendant toutes ces années et malgré cela tu restes le plus jovial de l'équipe. Tu resteras toujours cette personne qui peut passer une semaine sur 2K13 sans voir le jour et la semaine suivante, donner tout ce que tu as sur un projet qui te tiens à cœur. Tâche de garder cette foi pour le futur, couplée à ta curiosité, elle t'ouvrira toutes les portes.
Rose, la SICOM, les tournois avec les Licornes Roses, tu as cette chaleur qui permet de d'aller droit au but (c'est cadeau), qui permet de parler des choses vraies sans détour.
Brice, plus belle rencontre en ligne que j'ai pu faire, c'est toujours un plaisir de râler ou de refaire le monde avec toi. Je te souhaite de trouver ta voie les jours où tu ne télé-travailles pas.
Amélie, le monde est petit, le collège GrandVille n'est pas un point commun banal. Au plaisir de se recroiser dans les bars de Nantes ou sur un terrain d'Ultimate.
Clément, mon coloc/collègue, mon Maconnard préféré, mon compagnon de jeu par excellence, tu n'es pas le mec parfait mais qu'est-ce que tu le fais bien. Avant d'avoir une chambre pour enfant avec Amandine, on aura une chambre pour toi.

La suite de texte est dédiée à ma première bande de copains et donc la dernière de cette liste, c'est la team de "Nancy". A nos premières heures sur les bancs du collège puis du lycée, en passant par nos parties de tarot, de baby et de loup garous, de la cour de récré au GR20, je parcourrais le monde avec vous.
Gaëtan, ("ton corps te dira merci"), le camarade de sport : je garde des souvenirs impérissables de la prépa où on révisait ensemble quand on allait pas faire les cross du coin.
Delphine, tu as la charge de vivre avec l'impécabilité de Gaëtan et les contre-temps

que cela représente, et pour cela tu as tout mon respect et mon soutien.
Rémi, tu nous as enseigné à quel point la fourberie et une tête d'ange sont un mélange puissant pour profiter de toutes les situations. Tu es le petit con qu'on veut tous avoir comme ami. Tu as cette engouement qui rend les choses simples à réaliser.
Marion, tu es la douceur incarnée, à parfois vouloir t'effacer pour ne pas risquer de prendre de la place, sauf celle qui te revient dans nos cœurs.
JB, notre GO, tu as franchis les étapes pour reprendre le flambeau familial. Tu es un homme accompli maintenant. Qu'est ce qui vient alors ensuite ?
Émilie, si JB est le papa du groupe alors tu dois être la maman. Je te souhaite bien du courage pour continuer dans le monde hospitalier de la pédiatrie. N'oublies jamais que ton rire peut dérider n'importe qui.
Luc, le frère que je n'ai jamais eu, tu m'as fait découvrir la magie du trek et le plaisir des choses simples : manger, marcher, dormir, refaire le monde en discutant, et jouer encore et toujours. J'adore profiter de ton recul et de tes avis sur le monde qui nous entoure.
Fanny, les cours de latin, Mme Shöpfel, la 205, la prépa, on a passé des sacrés moments. Que tu rages ou que tu kiffes, tu fais toujours les choses à fond, c'est une de tes grandes forces.
Manon, tu es ce mélange détonnant entre complicité et gros coup de pied dans la fourmilière. Les escapades à Nancy me font toujours le plus grand bien avec toi.

Finalement, cette réussite, je la dois à ma famille qui m'a toujours soutenu dans les moments de doutes et accompagné dans toutes les étapes de ma vie. Je ne suis pas toujours présent pour les rendez-vous familiaux et sachez qu'une partie de moi voudrait ne jamais rien manquer.
Papa, tu m'as appris le goût de l'effort, à me surpasser quand il faut, qu'un long silence peut vouloir dire plus que tous les mots, et surtout qu'on peut toujours se débrouiller.
Maman, tu m'as appris a être bienveillant avec les autres, et à penser à moi quand il le faut. Je chéris nos digressions à rallonge qui fatiguent le reste de la famille et quitte à être en retard pour rentrer.
Chloé, je ne suis pas sûr d'avoir été un grand frère formidable, mais je suis fier de ce que tu es devenue et de ce que tu cherches à devenir. Tu auras toujours mon soutien.
François, t'as l'air sympa (c'est toi qui as commencé).

Petite mention à ma partenaire de confinement, celle qui m'a supervisé pendant ces deux dernières années et qui ne pourra pas lire ces lignes, ma chienne Jaïn qui me lève tous les matin d'un bon coup de truffe.
Amandine, (Dans ma clé USB, j'ai une photo de toi ♪). Tu m'as aidé à surmonter cette épreuve en me mettant face à mes responsabilités, et en me félicitant quand il fallait. Avec toi, je peux être le fou, le lâche, l'idiot que je ne peux être avec personne d'autre. Paradoxalement, tu es aussi la personne pour laquelle j'ai envie d'être le plus intelligent, le plus courageux et le plus fort. Tu es ma raison d'être une personne meilleure. J'ai hâte de faire ce voyage avec toi.

Table des matières

Acronymes

CNN Convolutional Neural Network, Réseaux convolutifs en français.

CPP Comité de Protection des Personnes, équivalent français de l'Institutional Review Board (IRB).

DICOM Digital Imaging and COmmunications in Medicine.

DL Deep Learning, apprentissage profond en français.

EC Entropie Croisée, cross entropy en anglais.

FEM Finite Element Method, Méthode des Éléments Finis.

FBDSD Feature-Based Deformable Surface Detection.

IRM Imagerie par Résonance Magnétique.

MITK The Medical Imaging Interaction Toolkit est un logiciel open-source pour le développement d'un logiciel interactif de traitement d'images médicales.

ML Machine Learning, apprentissage statistique en français.

RA Réalité Augmentée.

RANSAC RANdom SAmple Consensus.

RBF Radial Basis Function.

RGB Rouge Vert Bleu, système de codage informatique des couleurs.

RNN Recurrent Neural Network, Réseaux de Neurones Récurrent en français.

RSMR Root Mean Square Residual.

SfM Structure-from-Motion.

SIFT Scale-Invariant Feature Transform.

SLAM Simultaneous Localization And Mapping, localisation et cartographie simultanée en français.

SURF Speeded Up Robust Features.

TDM Tomodensitométrie.

TPS Thin-Plate Spline.

Chapitre 1

Introduction

Cette thèse a été réalisée entre septembre 2017 et octobre 2021 au sein de l'équipe Endoscopy and Computer Vision [1] (EnCoV) de l'Institut Pascal, et de Be-Ys Research [2], filiale du groupe Be-Ys, à Clermont-Ferrand.

1.1 Vision par ordinateur

1.1.0.1 Définition de la vision par ordinateur

La vision par ordinateur, *computer vision* en anglais, est une discipline scientifique qui consiste à traiter les informations contenues dans les images numériques de manière automatique. Ses tâches incluent l'acquisition, le traitement, l'analyse et la compréhension des images.

La vision par ordinateur est apparue dans les années 60. Les chercheurs se sont initialement inspirés des recherches en neurosciences, notamment celles qui visaient à expliquer le fonctionnement de la vision humaine et animale [57, 78, 48].

En particulier, ces études ont montré que le système de vision humaine repose sur une étape préliminaire qui repère des indices visuels simples comme les bords orientés, les contrastes de luminance, le mouvement et les textures. Le travail des scientifiques en vision par ordinateur s'est alors concentré sur la conception de représentations des images sous la forme d'un ensemble de caractéristiques, aussi connues sous la dénomination de descripteurs ou encore *features* en anglais, et d'algorithmes permettant le calcul de ces caractéristiques. Une fois extraites, ces caractéristiques forment la donnée d'entrée pour la réalisation d'une tâche donnée.

Les tâches de vision par ordinateur sont aujourd'hui généralement scindées en trois catégories, à savoir les tâches de reconnaissance, de reconstruction 3D et de recalage, parfois appelées "3Rs". La reconnaissance consiste à déterminer les objets, les actions, les événements présents sur une image ou une vidéo. Elle se décompose en de nombreux sous-domaines comme la détection d'objets ou la segmentation sémantique, à savoir l'opération qui consiste à classer l'ensemble des pixels d'une image, par exemple selon une classe d'appartenance à un objet. Le recalage consiste à mettre en correspondance des éléments provenant d'entrées différentes qui correspondent à une même réalité physique. Cela peut consister à trouver le déplacement d'un objet entre deux images prises au sein d'une vidéo, ou encore à aligner des

1. http://igt.ip.uca.fr/encov/
2. https://www.be-ys-research.com/

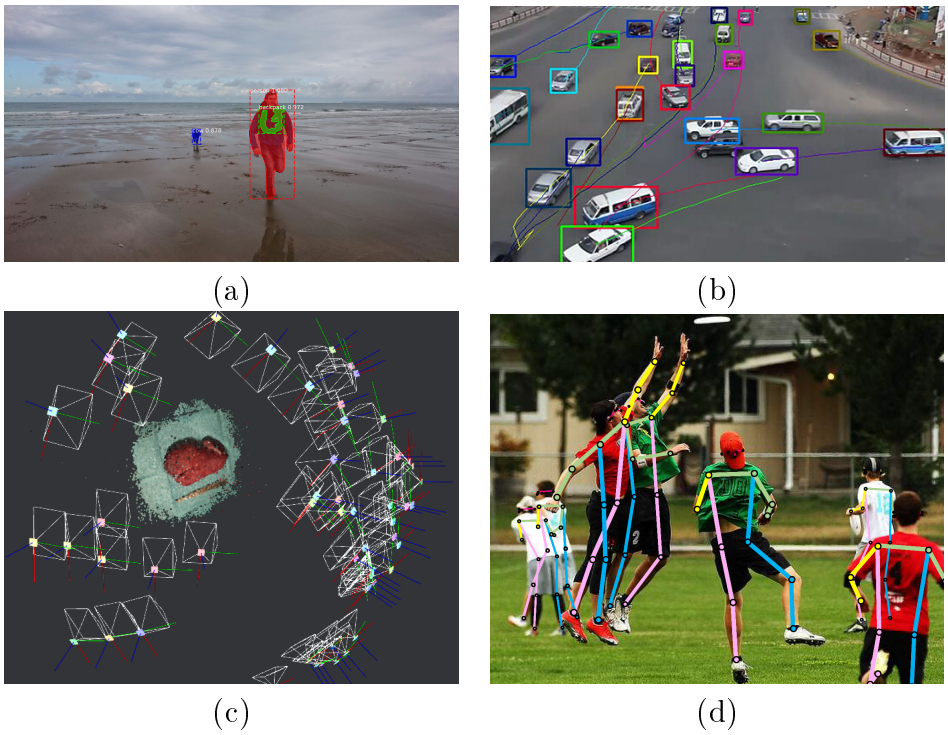


FIGURE 1.1 – Quelques exemples de tâches de vision par ordinateur. (a) Segmentation sémantique, (b) Détection et suivi d'objets, (c) Reconstruction 3D, (d) Estimation de pose 3D.

éléments issus d'images acquises selon différentes modalités d'imagerie. La reconstruction 3D consiste quant à elle à déduire la forme, en trois dimensions, d'une scène capturée à partir d'une ou plusieurs images.

Parmi les représentations d'images longtemps utilisées dans la littérature en vision par ordinateur, les représentations basées sur les contours, à savoir les zones de l'image associées à une forte variation de signal, sont notables. Ces zones indiquent en général des éléments de structures capturées par l'image. Sur la base de techniques utilisées en traitement du signal, des filtres ont été élaborés afin d'extraire les zones où le gradient de l'image est maximal, tel que le filtre de Sobel, ou encore des zones où la dérivée seconde du signal s'annule, par exemple lors d'un filtrage basé sur l'opérateur Laplacien. Ces filtres, dits convolutifs, appliquent des opérations sur un pixel et son entourage. Ils permettent de transformer l'image pour repérer des zones d'intérêt. Un exemple de représentation d'une image à partir de ses contours obtenus par un filtre de Sobel est illustrée sur la figure 1.2. De telles approches présentent l'inconvénient de se limiter à un traitement local de l'information et ne sont pas adaptées à l'extraction d'informations liées au contexte.

1.1.0.2 Apprentissage profond

Il est aujourd'hui possible de capturer des informations plus globales et associées au contexte grâce notamment aux méthodes basées sur de l'apprentissage profond, plus connu sous sa dénomination anglaise de *deep learning* (DL). Ces méthodes sont aujourd'hui l'état de l'art pour un très grand nombre de tâches visuelles. Elles font intervenir des réseaux de neurones artificiels dont l'unité élémentaire est appelée neurone formel.

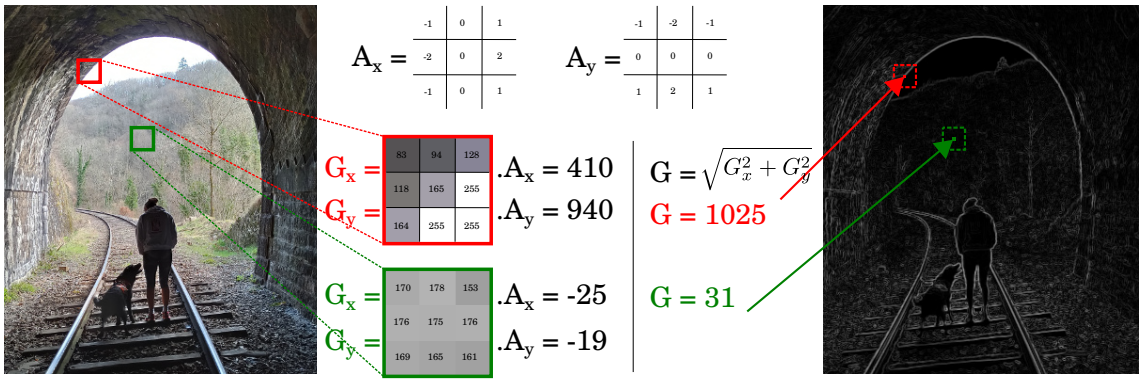


FIGURE 1.2 – Illustration de l'application d'un filtre convolutif de Sobel sur l'image de gauche. Les convolutions des filtres de Sobel A_x et A_y sont appliquées à une portion de l'image afin d'obtenir le gradient de l'image. On réalise cette opération en balayant toute l'image ce qui permet d'obtenir la représentation de droite (les pixels blancs indiquent une norme de gradient élevée et les pixels sombres une norme faible). Nous avons illustré cette opération sur deux portions de l'image (cadres vert et rouge). Le cadre rouge obtient une valeur élevée par rapport au cadre vert. Le cadre rouge est susceptible de contenir un contour. Cette transformation permet d'obtenir une segmentation grossière de la scène capturée.

La notion de neurone formel est apparue dans les années 1950. Le premier réseau de neurones artificiel, appelé Perceptron, a quant à lui été proposé en 1957. Il s'agissait d'un classifieur linéaire qui ne permettait pas encore de résoudre des problèmes complexes.

Il a fallu attendre l'année 1986, lors de laquelle les méthodes de rétro-propagation du gradient [102] ont été proposées, pour permettre l'apprentissage de modèles composés de plusieurs couches de neurones. Cette nouveauté a été utilisée dans le NeoCognitron [41], puis pour la conception du réseau LeNet-5 [65] qui ont été les premiers réseaux de neurones convolutifs. Ces méthodes ont été utilisées pour reconnaître des chiffres manuscrits afin de permettre la numérisation des codes postaux aux États-Unis dans [65]. Néanmoins, ces méthodes n'étaient pas encore adaptées à des problèmes plus complexes tels que la détection d'objets.

L'utilisation des réseaux de neurones artificiels a par la suite bénéficié de l'apparition de grandes puissances de calcul, des améliorations logicielles telles que la parallélisation et enfin de la démocratisation des systèmes d'acquisition d'images numériques ayant largement contribué à la mise à disposition massive d'images numériques. Ces améliorations ont permis à des réseaux de neurones comprenant un très grand nombre de couches de pouvoir être entraînés. Chaque couche ajoutée permet alors de représenter une image à partir d'un niveau d'abstraction supérieur. C'est de la multiplication de ces couches et la profondeur du réseau ainsi généré que le terme d'apprentissage profond fait référence.

À la fin de l'année 2012, c'est la première fois qu'un réseau de neurones profond, appelé AlexNet [63], surpasse l'ensemble des autres méthodes lors d'une compétition de classification d'images sur la base de données ImageNet [29] en atteignant pour la première fois un taux d'erreur de 15,3% dans le top-5. Ce taux d'erreur est obtenu en comptant le nombre de fois où la classe attendue ne figure pas dans le top 5 des prédictions.

L'apprentissage profond à connu un vif succès ces dernières années pour les tâches de reconnaissance. Sa popularité peut en grande partie se justifier par sa capacité à extraire automatiquement les caractéristiques spécifiques et discriminantes pour le problème traité, à la différence de méthodes d'apprentissage automatique plus anciennes qui nécessitaient une définition en partie manuelle des caractéristiques de l'image.

1.2 Chirurgie cœlioscopique en gynécologie

Les chirurgies gynécologiques sont prévues dans la prise en charge des maladies affectant les organes génitaux de la femme qui incluent le vagin, la vulve, l'utérus, les ovaires, et les seins. Ces maladies sont dépistées à l'aide d'examens comme la colposcopie, qui permet de confirmer un diagnostic après un frottis anormal, une biopsie, qui consiste à prélever un fragment de la muqueuse de l'utérus pour une analyse et une hystéroscopie, qui vise à explorer la cavité utérine à l'aide d'une caméra. Ces chirurgies consistent le plus souvent à des ablations partielles, pour retirer une tumeur, ou des ablations complètes, qui peuvent être des hystérectomies, ovariectomies dans des situations plus délicates. Dans le cadre de cette thèse, nous nous intéresserons uniquement à la chirurgie de l'utérus et en particulier à la myomectomie. Ces opérations consistent à retirer une ou plusieurs tumeurs bénignes, appelées myomes ou fibromes. Ces tumeurs se développent à partir du muscle utérin et du tissu fibreux de l'utérus. On estime qu'environ un tiers des femmes en âge de procréer en Europe sont touchées par ces tumeurs. Ces tumeurs sont bien souvent asymptomatiques mais elles peuvent se caractériser par des saignements abondants lors ou en dehors de la période de règles, et des douleurs abdominales. Elles peuvent également altérer la fonction reproductrice. Il existe trois voies d'abord chirurgicale pour ce traitement chirurgical. Le choix de la voie d'abord est en grande partie défini par la localisation des tumeurs. Dans le cas de tumeurs de moins de 4cm situées dans la cavité utérine, une hystéroscopie opératoire par les voies naturelles est généralement privilégiée. Lorsqu'il y a moins de 3 fibromes de dimension inférieure à 8 cm, la cœlioscopie est préférée. Dans les autres cas, on réalise une laparotomie qui consiste à réaliser une large incision dans l'abdomen pour pouvoir accéder à l'utérus. Un grand nombre de ces opérations sont aujourd'hui réalisées par cœlioscopie pour lesquelles la localisation des structures anatomiques internes, comme des tumeurs intra-murales, est un enjeu majeur pour le chirurgien [79].

La cœlioscopie, *laparoscopy* en anglais, consiste à réaliser de petites incisions dans l'abdomen par lesquelles l'équipe médicale fait passer les outils nécessaires à l'opération. Parmi ces outils, on utilise une caméra avec un tube optique pour observer l'intérieur de la cavité abdominale et contrôler les outils. Le flux vidéo provenant de la caméra est affiché sur un écran disposé face au chirurgien en salle d'opération. Pour créer un espace de travail à l'intérieur de la cavité abdominale, la cœlioscopie nécessite l'insufflation d'un gaz. Cet espace, appelé cavité péritonéale, est l'espace entre les deux membranes qui tapissent l'abdomen, le péritoine. Il contient des organes tels que le foie, les intestins par exemple. L'utérus et les trompes utérines sont dit sous-péritonéaux, c'est-à-dire en dessous du sac fermé que constitue le péritoine (voir la figure 1.3). Au niveau des incisions, des trocarts sont utilisés pour d'une part étanchéifier la cavité sous-pression et d'autre part permettre le passage des outils et le tube optique.

FIGURE 1.3 – (Gauche) Illustration du principe de cœlioscopie. L'écran en haut à droite affiche le retour du cœlioscope pour permettre à la /au chirurgien(ne) de visualiser l'action des outils. (Droite) Photographie présentant une opération utilisant la cœlioscopie à l'hôpital Estaing à Clermont-Ferrand (©La Montagne).

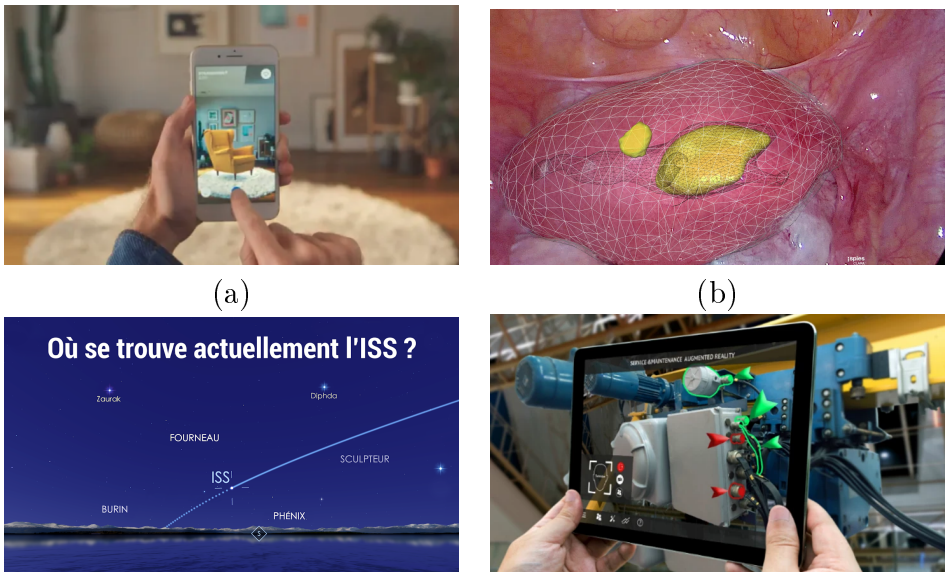
Cette technique a été utilisée pour la première fois en 1940 à des fins diagnostiques seulement. C'est en 1972 que la cœlioscopie est utilisée en chirurgie pour la première fois par le Docteur Mouret, à Clermont-Ferrand.

Le terme cœlioscopie peut à la fois désigner l'acte de diagnostic et celui de la chirurgie. Dans ce dernier cas, on peut préférer le terme de cœliochirurgie pour être moins ambigu. Dans ce mémoire, nous ne ferons référence qu'à l'acte chirurgical.

La cœlioscopie est une technique de chirurgie dite *minimalement invasive* qui offre plusieurs avantages pour le/la patient(e) :

— diminution de la douleur post-opératoire ;
— diminution du risque infectieux ;
— diminution du temps d'hospitalisation et de la période de convalescence ;
— diminution du risque d'adhérence dans le péritoine. Une adhérence est une lésion qui se forme suite à une opération chirurgicale, au moment de la cicatrisation. La peau vient se coller aux tissus, aux organes et aux muscles situés sous la peau.

Cependant cette méthode présente plusieurs difficultés techniques, en particulier pour le/la chirurgien(ne). Le champ de vision est très réduit comparé à la laparotomie. En effet, les mouvements de la caméra sont contraints par le trocart et le retour visuel se fait via un écran externe ce qui limite la perception de la profondeur de la scène et modifie la perception de la direction des mouvements réalisés. Par ailleurs, il n'est pas possible de palper les tissus, ce qui peut être utile pour repérer certains éléments que l'on ne peut voir à travers la surface de l'organe. Cette absence d'accès direct empêche de pouvoir stopper rapidement une hémorragie. Finalement, les outils sont également contraints par la position des trocarts ce qui limite leurs mouvements.



(a) (b) (c) (d)

FIGURE 1.4 – Quelques exemples d'applications de RA : (a) application Ikea Place pour visualiser l'ajout potentiel d'un nouvel achat (©ikea.com) ; (b) RA du logiciel Uteraug pour assister le/la chirurgien(ne) ; (c) application Star Walk 2 pour indiquer les étoiles et objets célestes en observant le ciel avec un smartphone (©vitotechnology.com) ; (d) exemple d'application RA pour la formation (©stilla-studio.fr).

1.3 Guidage du geste chirurgical par réalité augmentée

Pour pallier ces difficultés techniques, des systèmes de réalité augmentée (RA) ont été proposés afin de guider le geste du chirurgien pendant l'opération. L'objectif clinique de cette thèse est l'amélioration d'un système existant de guidage du geste chirurgical par réalité augmentée dans le cadre de la chirurgie gynécologique. Notons que les améliorations proposées pourront naturellement être étendues à des systèmes de guidage utilisés dans d'autres services de chirurgie tels que par exemple la chirurgie hépatobiliaire ou néphrologique.

1.3.0.1 Définition de la réalité augmentée

La réalité augmentée consiste à insérer des éléments virtuels (appelés augmentations) dans une image ou une vidéo réelle en temps réel. Le rendu visuel crée ainsi une illusion où les augmentations et la scène réelle semblent coexister de manière cohérente. La réalité augmentée nécessite un écran sur lequel est affichée la superposition entre les éléments réels et virtuels. La réalité augmentée diffère de la réalité virtuelle qui propose à l'utilisateur une immersion au sein d'un environnement entièrement virtuel. Pour assurer l'illusion de cohérence entre les éléments virtuels et réels, la RA doit suivre et positionner les augmentations correctement à partir de la connaissance d'une partie de la géométrie de la scène filmée et du mouvement de la caméra. Ce suivi est rendu possible grâce à des techniques de vision par ordinateur.

1.3.0.2 Enjeux d'un logiciel de réalité augmentée pour la chirurgie cœlioscopique

La cœliochirurgie est un contexte très favorable pour l'application de techniques de vision par ordinateur.

D'une part, le protocole utilise déjà une caméra numérique qui enregistre un flux vidéo. Ce flux vidéo est déjà en partie traité par une unité de calcul présente sur la colonne d'endoscopie avant d'être affiché sur les écrans en salle d'opération. D'autre part, la vision par ordinateur est aujourd'hui une discipline scientifique suffisamment mature et les capacités de calcul sont suffisamment importantes pour permettre la mise en œuvre des traitements des images en temps réel.

La réalité augmentée présente de nombreux intérêts dans le domaine médical. Cependant, un système de RA dans ce domaine est soumis à un certain nombre d'exigences qui, si elles ne sont pas vérifiées, peuvent entièrement remettre en question son utilisation. Premièrement, le système doit être fiable. En France, un logiciel utilisé dans le domaine médical doit être accrédité en fonction de son degré d'implication dans le processus du praticien. On différencie ainsi le logiciel d'agenda de l'hôpital de celui capable d'assister le chirurgien dans son diagnostic et son geste. Cela ne veut pas dire que le logiciel doit garantir un rendu de RA dans tous les cas mais il doit prévoir des moyens pour permettre au personnel de santé de continuer la procédure dans de bonnes conditions même dans une situation pour laquelle le logiciel ne peut pas fonctionner correctement. À des fins de recherche, un chirurgien peut réaliser des expérimentations qui ont été validées auparavant par le Comité de Protection des Personnes (CPP). Ce comité est chargé de vérifier que les expérimentations menées sont réglementaires et pertinentes et d'évaluer si le rapport bénéfice/risque est suffisant. Au préalable de chaque expérimentation, le/la patient(e) doit avoir donné son consentement éclairé pour mener ces recherches.

Deuxièmement, le système doit être précis. La réalité augmentée crée une illusion qui doit aider le personnel de santé. Si la tumeur à afficher est systématiquement décalée dans le monde réel par rapport à ce qui est affiché à l'écran, le risque de provoquer des complications pour le/la patient(e) devient élevé. Il est important de valider les méthodes développées sur des situations réelles pour lesquelles on est en mesure d'établir la précision du système.

En plus de ces contraintes légales et nécessaires au bon fonctionnement, il est important que le logiciel apporte une modification du protocole proportionnée à l'amélioration qu'il permet. Dans le cadre de la RA, l'amélioration de la vision doit permettre un gain en temps et en précision du geste chirurgical. Cependant, si ce gain nécessite une mise en place matérielle trop complexe ou une augmentation du personnel, alors le déploiement de ce logiciel sera grandement limité.

1.3.0.3 Méthodes actuelles pour la cœlioscopie augmentée

La cœlioscopie guidée par réalité augmentée, que l'on appelle également cœlioscopie augmentée, est aujourd'hui permise par un certain nombre de techniques. Ces techniques sont très spécifiques aux modalités des organes opérés. Elles nécessitent de recaler les images préopératoires volumiques, obtenues à partir d'un scanner TDM ou d'une IRM, qui contiennent les informations sur les structures internes (tumeurs, cavité, vascularisation, etc) sur les images cœlioscopiques du même organe. Ces images préopératoires, obtenues en amont de l'opération, nécessitent une

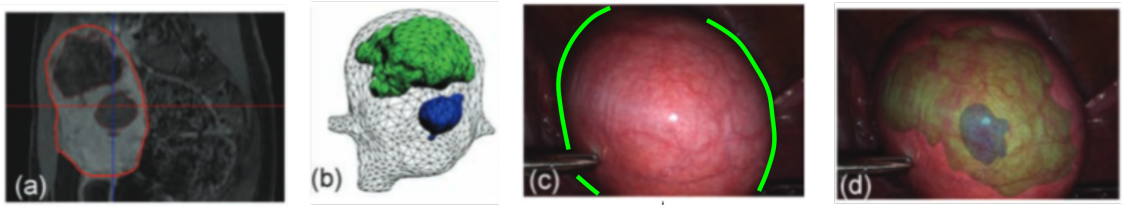


FIGURE 1.5 – Schéma de fonctionnement de la cœliochirurgie augmentée. (a) couche extraite d'une IRM sur laquelle l'utérus (bordure rouge) et les tumeurs ont été segmentés (régions sombres dans l'utérus). (b) Modèle 3D des différents éléments segmentés depuis l'IRM : utérus (gris), tumeurs (bleu et vert). (c) image courante du flux cœlioscopique avec les contours occultants de l'utérus (vert). (d) Rendu de la réalité augmentée après le calcul du recalage entre (b) et (c).

segmentation de l'organe et de ses structures internes. Cette segmentation peut-être réalisée de manière manuelle ou semi-automatique avec des logiciels comme MITK.

La plupart du temps, il est nécessaire de déformer le modèle préopératoire par rapport aux images courantes acquises en salle d'opération. L'insufflation de l'abdomen pour réaliser une cœlioscopie déforme les organes par rapport à la situation dans laquelle l'IRM ou le scanner a été réalisé. Ce recalage déformable est une étape coûteuse en temps de calcul. Il n'est donc pas réalisable de réitérer ce calcul pour chaque instant de l'opération. En effet, l'organe étudié est lui-même sujet à des déformations au cours de l'opération en fonction des actions du/de la chirurgien(ne). L'approche couramment employée par les méthodes actuelles [26, 55, 51] est de réaliser ce recalage pour un instant donné. Ce recalage est ensuite mis à jour en temps réel pour les images suivantes de manière rigide. Pour l'utérus ou le rein, cette contrainte n'est pas problématique car l'utérus est un organe relativement rigide et le rein n'est pas mobilisé avant d'être incisé.

Nous nous intéressons à la cœlioscopie monoculaire car c'est la solution standard dans de nombreuses salles d'opération. Toute solution pour la cœlioscopie monoculaire peut être étendue à la cœlioscopie stéréo. Il est à noter que la cœlioscopie augmentée assistée par ordinateur est aujourd'hui possible grâce à des robots tels que le Da Vinci. Ces robots permettent de manipuler les outils à la place du chirurgien. Le/la chirurgien(ne) contrôle le robot via une console dans laquelle il est plus confortable d'opérer, ses mouvements peuvent être démultipliés pour être plus précis et une visualisation en trois dimensions est fournie. Les inconvénients de ce type de solutions sont : le coût, de l'achat du robot et de ses consommables, et l'absence de retour haptique qui permet à l'opérateur de sentir une réaction au toucher.

L'équipe EnCoV a développé un logiciel nommé Uteraug pour permettre d'utiliser la réalité augmentée pour assister le geste chirurgical dans les conditions réelles d'une opération (voir la figure 1.5). Ce logiciel superpose les tumeurs repérées en préopératoire à l'intérieur de l'utérus avec le flux vidéo cœlioscopique au cours de l'opération. Sa mise en place reste relativement longue par rapport à son utilisation : environ 5 minutes de préparation pour une utilisation qui se limite à la planification des incisions par le/la chirurgien(ne). En effet, une fois l'organe incisé, le système ne peut plus estimer la pose de l'organe de manière fiable.

1.4 Contributions et organisation du mémoire

Cette thèse porte sur plusieurs améliorations pour le logiciel Uteraug, en particulier grâce à l'utilisation de solutions d'apprentissage profond. L'objectif de ce travail était de rendre l'utilisation plus accessible en automatisant certaines étapes préliminaires à la réalité augmentée ; et d'étendre l'utilisation de la réalité augmentée à la phase d'incision de l'utérus.

Le chapitre 2 introduit les notions nécessaires à la compréhension du mémoire, en particulier sur les méthodes d'optimisation et d'apprentissage profond. Ce chapitre présentera également les différentes étapes du logiciel de réalité augmentée Uteraug à son stade de développement correspondant au début de ce travail. Différents éléments de vision par ordinateur nécessaires à la bonne compréhension de ce travail comme par exemple le modèle de caméra sténopé ou certaines techniques de reconstruction 3D y sont présentés. Le chapitre 3 décrit nos travaux sur la détection de contours sémantiques de l'utérus. Dans un premier temps, nous présentons un nouveau score pour évaluer la prédiction de contours. La suite présentera nos stratégies mises en place pour améliorer la prédiction de contours sémantiques à partir d'un réseau convolutif, en particulier pour limiter leur épaisseur. Finalement, nous avons intégré cette méthode pour annoter automatiquement les contours dans le logiciel de réalité augmentée et comparé à l'approche manuelle. Le chapitre 4 présente nos travaux sur la détection d'incision pour mettre à jour la topologie du modèle virtuel. Ces travaux permettent d'étendre l'usage de la réalité augmentée lorsque le/la chirurgien(ne) réalise l'incision de l'organe, ce qui n'était pas possible pour le moment. Ce chapitre présente l'ensemble des procédés utiles pour permettre cette mise à jour et constitue un premier jalon pour montrer la faisabilité de cette méthode. Enfin, le chapitre 5 présente les conclusions et les perspectives de ce travail de thèse. En annexe, nous présentons une expérience qui nous a permis de tester l'impact de la taille de notre jeu de données sur les performances de deux tâches : la segmentation sémantique de l'utérus et la détection de ses contours occultants.

Chapitre 2

Notions de base

Dans ce chapitre, nous introduisons les notions de base et les prérequis à la compréhension de ce mémoire. Dans un premier temps, nous faisons un rappel sur les méthodes d'optimisation numériques et des notions de base liées à l'apprentissage profond mises en jeu dans ce travail. La dernière partie de ce chapitre présente les différentes étapes du logiciel de réalité augmentée Uteraug. Ces éléments permettent à la fois de donner du contexte aux contributions apportées par les chapitres 3 et 4, mais également de détailler plusieurs notions de vision par ordinateur qui sont directement ou indirectement utilisées dans les chapitres de contributions.

2.1 Notation

M	Matrice
$\mathbf{v}$	Vecteur
s	scalaire
$\mathbb{R}^n$	Espace vectoriel de dimension n
$\lvert \cdot \rvert$	Norme l1
$\lVert \cdot \rVert$	Norme l2
$\lVert \cdot \rVert_{\mathcal{F}}$	Norme de Frobenius
$\mathbf{a}.\mathbf{b}$	Produit scalaire des vecteurs $\mathbf{a}$ et $\mathbf{b}$
$\mathsf{A} \cdot \mathsf{B}$	Produit matriciel de Hadamard des matrices A et B
$\mathsf{M}^\top$	Matrice transposée
M^{-1}	Matrice inverse
$\mathsf{M}^\dagger$	Matrice pseudo-inverse
∇f	Gradient de la fonction f
$\frac{df}{dx}$	Dérivée première de la fonction f
$\frac{\partial f}{\partial x}$	Dérivée partielle de la fonction f en fonction de la variable x
log	Fonction logarithme naturel ou népérien
exp	Fonction exponentielle

2.2 Optimisation numérique

Le terme optimisation fait référence au calcul d'une inconnue $\mathbf{x}$ qui minimise ou maximise une fonction $f(\mathbf{x})$. La fonction à optimiser est appelée fonction de coût ou critère. Le terme optimisation sera synonyme de minimisation dans la suite de

ce manuscrit. Les notions présentées dans ce chapitre ont été extraites des ouvrages de référence sur la vision par ordinateur [37] et sur l'apprentissage profond [45].

2.2.1 Méthodes linéaires

Lorsque le problème peut se formuler sous la forme d'un système d'équations linéaires sur les inconnues du problème, il existe des méthodes matures, robustes et rapides pour trouver la solution optimale.

2.2.1.1 Moindres carrés linéaires

Le problème le plus classique est appelé moindres carrés linéaires. Il consiste à résoudre une équation de la forme :

$$\mathsf{A}\mathbf{x} = \mathbf{b}, \tag{2.1}$$

avec $\mathbf{b} \in \mathbb{R}^m$ et $\mathsf{A} \in \mathbb{R}^{m \times n}$ connus et $\mathbf{x} \in \mathbb{R}^n$ inconnu en minimisant :

$$f(\mathbf{x}, \mathsf{A}, \mathbf{b}) = \|\mathsf{A}\mathbf{x} - \mathbf{b}\|^2. \tag{2.2}$$

Il est possible de dériver une solution directe même si les méthodes numériques actuelles obtiennent des résultats plus stables et plus rapides en évitant l'inversion de matrice :

$$\mathbf{x} = \operatorname{argmin} f(\mathbf{x}, \mathsf{A}, \mathbf{b}) = (\mathsf{A}^\top \mathsf{A})^{-1} \mathsf{A}^\top \mathbf{b} = \mathsf{A}^\dagger \mathbf{b}, \tag{2.3}$$

avec $\mathsf{A}^\dagger$ la pseudo-inverse de A.

2.2.1.2 Décomposition en valeurs singulières

Une autre approche très utile pour la résolution de systèmes linéaires est la décomposition en valeurs singulières. Toute matrice $\mathsf{A} \in \mathbb{R}^{m \times n}$ admet une décomposition de la forme suivante :

$$\mathsf{A} = \mathsf{U}\mathsf{S}\mathsf{V}^\top, \tag{2.4}$$

où $\mathsf{U} = \mathsf{U}^\top \in \mathbb{R}^{m \times m}$, $\mathsf{S} \in \mathbb{R}^{m \times n}$ de coefficients diagonaux $\mathbf{s} \in \mathbb{R}^m$ et $\mathsf{V} = \mathsf{V}^\top \in \mathbb{R}^{n \times n}$. Les composantes de S sont appelées valeurs singulières. Les colonnes de U et V sont les vecteurs singuliers respectivement à gauche et à droite. Le nombre de valeurs singulières non nulles est égal au rang de la matrice A. La décomposition en valeurs singulières est un des algorithmes numériques les plus utilisés et bénéficie d'implémentations très efficaces.

2.2.2 Méthodes non-linéaires

Lorsque la fonction de coût est non linéaire, il est possible de résoudre le problème posé en utilisant une méthode itérative. Soit une fonction $f(\mathbf{x})$ à minimiser, en partant d'un jeu de paramètres $\mathbf{x}_n$, chaque itération produit un incrément $\Delta\mathbf{x}_n$ tel que $\mathbf{x}_{n+1} = \mathbf{x}_n + \Delta\mathbf{x}_n$ soit plus proche de la solution recherchée. Ces méthodes font toutes l'hypothèse que la fonction f est convexe. Dans le cas contraire, il est toujours possible d'utiliser ces méthodes avec une bonne initialisation, mais sans garantie de convergence vers le minimum global.

Un problème de minimisation aux moindres carrés minimise une fonction de la forme :

$$f(\mathbf{x}) = \sum_{i=1}^{m} f_i(\mathbf{x})^2. \tag{2.5}$$

Voici les méthodes principalement utilisées en vision par ordinateur.

2.2.2.1 Descente de gradient

Supposons que nous ayons une fonction $y = f(x)$, avec x, y des nombres réels. La *dérivée* de cette fonction est notée $f'(x)$ ou $\frac{dy}{dx}$. La dérivée donne la pente de la fonction f au point x. On peut comprendre la pente comme le facteur qui, associé à une petite variation autour du point x, donne la variation correspondante sur la sortie : $f(x + \epsilon) \approx f(x) + \epsilon f'(x)$. La dérivée est donc utile pour minimiser une fonction puisqu'elle nous dit comment modifier x pour qu'il améliore y. On sait, par exemple, que la valeur de $f(x - \epsilon f'(x))$ est plus petite que $f(x)$ pour une petite valeur de ϵ. On peut ainsi réduire $f(x)$ en modifiant x avec des petits pas dans le sens opposé de la dérivée. Cette technique est appelée *descente de gradient* et peut être généralisée à des fonctions f de $\mathbb{R}^n \mapsto \mathbb{R}$.

La descente de gradient est une méthode de résolution du premier ordre. La direction de déplacement choisie est directement liée au gradient de la fonction étudiée :

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \alpha \nabla f(\mathbf{x}_n). \tag{2.6}$$

La longueur du pas $\alpha \in \mathbb{R}$ est cruciale pour garantir la convergence en un nombre d'itérations le plus petit possible. Un pas trop important engendre une divergence du procédé de minimisation et un pas trop petit permet de converger (de façon sûre dans le cas d'un problème convexe) mais pour un nombre d'itérations élevé. L'avantage de cette approche est qu'elle converge efficacement même si le jeu de paramètres initial est éloigné du minimum recherché.

2.2.2.2 Newton

La méthode de Newton est une méthode du second ordre, basée sur une approximation quadratique de la fonction à minimiser. Le développement de Taylor de la fonction de coût f s'écrit :

$$f(\mathbf{x} + \Delta\mathbf{x}) \approx f(\mathbf{x}) + \nabla f(\mathbf{x})\Delta\mathbf{x} + \frac{1}{2}\Delta\mathbf{x}^\top \mathsf{H}(\mathbf{x})\Delta\mathbf{x}, \tag{2.7}$$

où la matrice hessienne H de f est définie par :

$$\mathsf{H}(\mathbf{x}) = \begin{pmatrix} \frac{\partial^2 f(\mathbf{x})}{\partial \mathbf{x}_1^2} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial \mathbf{x}_1 \partial \mathbf{x}_n} \\ \vdots & & \vdots \\ \frac{\partial^2 f(\mathbf{x})}{\partial \mathbf{x}_n \partial \mathbf{x}_1} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial \mathbf{x}_n^2} \end{pmatrix}. \tag{2.8}$$

Un extremum est atteint si et seulement si le gradient de l'équation (2.7) par rapport à $\Delta\mathbf{x}$ est nul :

$$\nabla f(\mathbf{x}) + \mathsf{H}(\mathbf{x})\Delta\mathbf{x} = 0 \Leftrightarrow \Delta\mathbf{x} = -\mathsf{H}(\mathbf{x})^{-1}\nabla f(\mathbf{x}). \tag{2.9}$$

L'incrément de Newton est donc :

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \mathsf{H}(\mathbf{x})^{-1}\nabla f(\mathbf{x}). \tag{2.10}$$

La méthode de Newton assure une convergence plus efficace lorsque l'approximation quadratique est valide, ce qui est habituellement le cas lorsque les paramètres sont proches de la solution.

La variante dite de Gauss-Newton permet d'éviter le calcul coûteux de la hessienne. Elle s'applique uniquement aux problèmes de moindres carrés avec un coût de la forme :

$$f(\mathbf{x}) = \sum_{i=1}^{m} f_i(\mathbf{x})^2, \tag{2.11}$$

avec m le nombre de résidus à minimiser et n le nombre de paramètres tel que $\mathbf{x} \in \mathbb{R}^n$. La hessienne peut alors être approximée par : $\mathsf{H} \approx \mathsf{J}^\top\mathsf{J}$. L'incrément de Gauss-Newton est donc :

$$\mathbf{x}_{n+1} = \mathbf{x}_n - (\mathsf{J}(\mathbf{x})^\top\mathsf{J}(\mathbf{x}))^{-1}\mathsf{J}(\mathbf{x}). \tag{2.12}$$

2.2.2.3 Levenberg-Marquardt

La méthode d'optimisation non linéaire de Levenberg-Marquardt repose sur les deux approches précédemment citées afin de profiter de leur avantage respectif. Plus stable que celle de Gauss-Newton, elle trouve une solution même si elle est démarrée très loin d'un minimum. Cependant, pour des fonctions très régulières, elle peut converger légèrement moins vite.

2.2.3 Robustesse

De nombreuses méthodes d'ajustement utilisent des fonctions d'erreur carré. En pratique, la moindre donnée inappropriée peut alors dominer les erreurs provoquées par les autres bonnes données. Cette erreur peut créer un biais dans le processus d'ajustement. Cet effet vient de l'utilisation de la fonction carré. Il est difficile d'empêcher l'apparition de ce type de données, appelés *outliers* en anglais. Cela peut venir d'une erreur lors de la collecte des données ou tout simplement d'un effet rare qui est négligé par le modèle considéré. Les étapes de mise en correspondances sont particulièrement sujettes à la génération d'outliers.

Ce problème peut être résolu soit en réduisant l'impact des données loin de l'estimation ou en cherchant à identifier ces données incompatibles avec le modèle utilisé. La première solution consiste à remplacer la fonction carré par une autre fonction plus adéquate. C'est l'approche des M-estimateurs. La seconde option est plus complexe mais nous présenterons une méthode, celle de RANSAC, qui permet de trouver les points compatibles avec le modèle.

2.2.3.1 M-estimateurs

Un M-estimateur permet d'estimer les paramètres en remplaçant le terme d'erreur au carré par un terme qui limite l'impact des outliers. Soit r_i le résidu de l'erreur associée à l'observation $(\mathbf{x}_i, \mathbf{y}_i)$, c'est-à-dire la différence entre l'observation $\mathbf{y}_i$ et la valeur prédite par le modèle estimé à partir de l'entrée $\mathbf{x}_i$. La méthode des

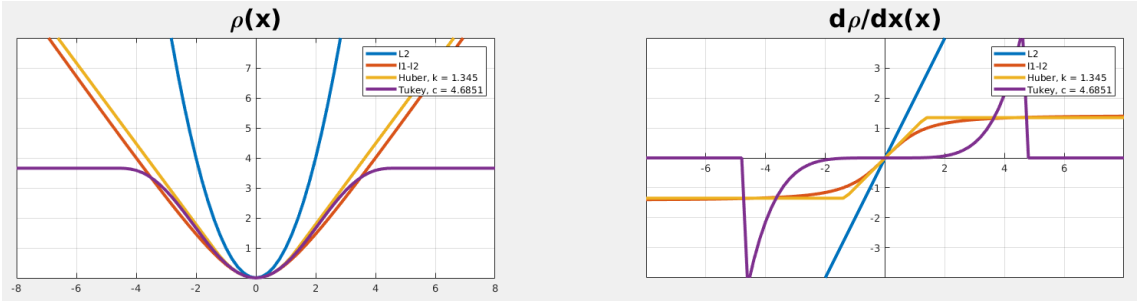


FIGURE 2.1 – (Gauche) Exemples de plusieurs M-estimateurs comparés à l'évolution de l2 (moindres carrés). On constate que tous les M-estimateurs proposent une croissance plus faible à mesure que l'erreur est grande en norme. (Droite) Dérivées associées aux M-estimateurs présentés et l2.

moindres carrés essaye de minimiser $\sum_i r_i^2$ mais devient instable lorsque des outliers sont présents. Le M-estimateur résout alors le problème suivant :

$$\min \sum_i \rho(r_i), \tag{2.13}$$

avec ρ une fonction réelle, symétrique, définie positive avec un minimum unique en 0 et choisie pour croître moins vite que la fonction carré. En utilisant ces fonctions, on limite l'impact des résidus très grands qui ont de grandes chances d'être des outliers. Un autre critère pour choisir la fonction ρ est d'avoir un comportement proche de la fonction carré sur les petites erreurs.

Ils existent de très nombreux M-estimateurs. En voici quelques uns :

$$\rho_1(x) = 2(\sqrt{1 + x^2/2} - 1), \tag{2.14}$$

$$\rho_2(x) = \begin{cases} \frac{x^2}{2} & \text{si } |x| < k \\ k(|x| - \frac{k}{2}) & \text{si } |x| \geq k, \end{cases} \tag{2.15}$$

$$\rho_3(x) = \begin{cases} \frac{c^2}{6}(1 - [1 - x/c)^2]^3) & \text{si } |x| \leq c \\ \frac{c^2}{6} & \text{si } |x| > c. \end{cases} \tag{2.16}$$

Ces fonctions sont connues sous les noms suivants : ρ_1 correspond au M-estimateur l1-l2, ρ_2 à celui de Huber et ρ_3 à celui de Tukey.

2.2.3.2 RANSAC

Plutôt que de modifier la fonction de coût, une autre approche consiste à identifier les données compatibles avec le modèle et à les séparer des outliers. L'algorithme *RANdom SAmple Consensus* (RANSAC) [36] consiste à estimer les paramètres du modèle sur un sous-ensemble des données et vérifier si le modèle obtenu est compatible avec les autres données.

L'algorithme se décompose en plusieurs étapes. Tout d'abord, un sous-ensemble de données est sélectionné de manière aléatoire et uniforme. On ajuste ensuite les paramètres du modèle pour ce sous-ensemble. On compare le modèle obtenu avec les autres données laissées de côté. Si un certain nombre de données suffisamment proches du modèle est trouvé alors on considère ce modèle comme bon. On ajuste alors le modèle avec les données utilisées précédemment et celles qui étaient suffisamment proches et on calcule l'erreur d'ajustement sur toutes ces données. On

répète le processus un certain nombre de fois. A chaque itération, un modèle est soit rejeté parce que trop peu de données sont assez proches, soit réajusté sur les données retenues. Finalement, on retient le modèle qui a obtenu l'erreur d'ajustement la plus faible parmi tous les modèles retenus.

Cet algorithme nécessite de définir quatre paramètres : la taille du sous-ensemble de données tirés aléatoirement, le seuil pour décider si une donnée est proche du modèle ajusté ou non, le nombre seuil de données proches pour considérer le modèle comme bon candidat et finalement le nombre d'itération.

2.3 Apprentissage profond

L'apprentissage profond, *deep learning* (DL) en anglais, est un type d'apprentissage statistique, *machine learning* (ML) en anglais. Le but de cette section est d'illustrer plusieurs principes et modalités utilisés en apprentissage statistique et apprentissage profond.

Un algorithme d'apprentissage est un algorithme qui est capable d'apprendre à partir de données. Que veut dire apprendre dans le contexte d'une machine ? Une définition a été proposée par Mitchell (1997) [87] : On dit qu'un programme informatique apprend d'une expérience E, vis-à-vis d'une tâche T et d'une mesure de performance P, lorsque sa performance pour la tâche T, mesurée par P, s'améliore avec E. Cette définition relativement large permet d'englober un très grand nombre d'entités derrière ces notions d'expérience, de tâche et de mesure de performance.

Un algorithme d'apprentissage repose bien souvent sur deux étapes : la première consiste à estimer un modèle à partir de données, parfois appelées observations, qui sont disponibles en nombre fini. À partir de ces observations, on essaye d'obtenir un jeu de paramètres qui permet de répondre au mieux à la tâche visée en fonction des observations disponibles. On appelle cette phase l'entraînement. La seconde phase consiste à utiliser le modèle avec les paramètres retenus pendant la phase d'entraînement sur de nouvelles données. On appelle cette phase l'inférence. Les paramètres de l'algorithme sont alors fixés.

On appelle jeu de données, *dataset* en anglais, une collection d'exemples ou d'observations. Ces exemples peuvent être des données brutes : un signal électrique échantillonné dans le temps, la valeur d'une mesure physique, une image. Ces données peuvent également être des caractéristiques (*features* en anglais) obtenues à partir de la transformation de données brutes.

L'apprentissage profond constitue une sous-classe de l'apprentissage automatique. Il se distingue en particulier par la combinaison d'un grand nombre de couches d'unités de traitement non linéaire. Le terme de profondeur traduit alors la quantité de couches entre la couche d'entrée et la couche de sorties. Les architectures d'apprentissage profond se caractérisent ainsi par la quantité immense de paramètres que l'on distingue en poids et seuils. Cette caractéristique a notamment été exploitée pour intégrer l'extraction de caractéristiques dans la phase d'apprentissage.

2.3.1 Quelques types de tâches

L'intérêt des méthodes d'apprentissage automatique est de gérer des problèmes complexes qui sont trop difficiles à résoudre avec des programmes dont le raisonnement serait entièrement détaillé par des opérateurs humains. Un exemple de tâche

qui illustre bien l'intérêt des méthodes d'apprentissage automatique est la détection de visages. Il s'agit d'une tâche que l'être humain est capable de faire dès le plus jeune âge (au point même où il devient capable de voir des visages n'importe où). Pourtant, pour un programme informatique, reconnaître un visage dans une image, c'est-à-dire des tableaux gigantesques de valeurs, est une tâche ardue. Il est vain de décrire tous les cas particuliers qui caractérisent un visage dans une image. Les approches d'apprentissage sont alors utilisées.

Voici une liste non exhaustive de tâches classiques en apprentissage automatique :

Classification La classification consiste à trouver à quelles classes correspondent les entrées du système. Le système a connaissance d'un nombre fini de classes. Par exemple, le système embarqué d'une caméra peut détecter quand un visage est visible.

Régression Le programme doit estimer une valeur numérique en fonction des entrées qui lui sont données. Par exemple, on peut demander à un robot d'estimer la distance qui le sépare de l'obstacle le plus proche à partir de ses capteurs.

Transcription Le programme reçoit une entrée dans un certain format et doit le convertir dans une autre structure. Cela peut être utilisé par un outil de traduction automatique ou un système de reconnaissance vocale qui convertit ce que l'opérateur prononce dans un format texte.

Détection d'anomalie Le programme observe une série d'événements ou de données et doit être capable de détecter lorsque l'une de ces entrées est inhabituelle ou atypique. Ce type de tâche peut être utilisé pour détecter des opérations frauduleuses sur une carte bancaire ou pour le suivi qualité d'une usine de production.

2.3.2 Types d'apprentissage

On distingue différents types d'apprentissage en fonction de l'expérience autorisée pendant le processus d'apprentissage. On distingue en particulier les méthodes supervisées des méthodes non supervisées. On peut également distinguer une dernière forme d'apprentissage qui ne sera pas développée en profondeur dans ce manuscrit que l'on appelle apprentissage par renforcement.

2.3.2.1 Apprentissage non supervisé

L'apprentissage non supervisé consiste à utiliser un jeu de données et à apprendre les propriétés de la structure de ce jeu de données. Lors de l'apprentissage, l'algorithme ne dispose pas d'information sur la sortie attendue. Un exemple de tâche est le regroupement (*clustering* en anglais) qui consiste à créer des groupes à partir d'exemples similaires du jeu de données. Un autre exemple de tâche non supervisée consiste à retrouver l'observation d'origine à partir d'une observation altérée. Nous reviendrons sur ce type d'usage dans la section 2.3.2.6.

2.3.2.2 Apprentissage supervisé

L'apprentissage supervisé nécessite un jeu de données plus complet. En plus des exemples qui la composent, qui sont les entrées de notre système (image, variable

temporelle, etc), chaque exemple est associé à une valeur cible, ou étiquette (*label* en anglais). Cette étiquette correspond à la prédiction attendue de la part de notre algorithme d'apprentissage pour l'exemple sélectionné. Dans le cadre de la classification d'images, l'étiquette correspond à la classe d'objet qui correspond à l'image. L'apprentissage supervisé consiste à inciter l'algorithme à reproduire les étiquettes du jeu de données.

2.3.2.3 Apprentissage par renforcement

Une autre variante de type d'apprentissage est l'apprentissage par renforcement. Plutôt que de disposer d'un jeu de données préétabli, cet apprentissage consiste à interagir avec son environnement pour compléter son apprentissage. C'est une approche très utilisée dans le cadre du pilotage de robots autonomes qui, par exemple, doivent atteindre une cible en un temps donné sur un parcours inconnu. Ce sont les différents essais que va réaliser le robot qui vont lui permettre de construire un jeu de données de son environnement pour ainsi établir quelles sont les stratégies gagnantes ou non. Ce type d'apprentissage ne sera pas exploité dans nos travaux.

2.3.2.4 Apprentissage par transfert

L'apprentissage par transfert, *transfer learning* en anglais, est un domaine de recherche de l'apprentissage statistique qui vise à transférer des connaissances d'une ou plusieurs tâches sources vers une ou plusieurs tâches cibles.

On suppose qu'un premier entraînement à été réalisé avec un premier jeu de données qui se caractérise par une première distribution de probabilité. On suppose qu'un certain nombre de facteurs qui expliquent les variations du premier jeu de données sont pertinents pour expliquer les variations d'un nouveau jeu de données similaire ou pour répondre à une tâche similaire. Dans le cadre de la vision par ordinateur, de nombreuses catégories visuelles partagent des notions de bas niveau de bords et de formes visuelles. Par exemple, un premier jeu de données peut permettre d'entraîner un modèle pour reconnaître des voitures. Les caractéristiques retenues par le modèle peuvent être de bonnes caractéristiques pour reconnaître des camions par exemple.

Concrètement, les poids du réseau du premier entraînement sont utilisés pour initialiser les poids du réseau sur la seconde tâche. On appelle cela le pré-entraînement d'un modèle. Il est très fréquent de pré-entraîner le réseau avec un jeu de données conséquent comme celui d'ImageNet [29] plutôt que d'utiliser un réseau initialisé aléatoirement. L'apprentissage par transfert est particulièrement utile quand on dispose d'un jeu de données limité pour la seconde tâche.

2.3.2.5 Adaptation de domaine

L'adaptation de domaine est un concept proche de l'apprentissage par transfert. Dans ce cas, la tâche entre le premier et le second entraînement est très proche. La différence vient cette fois-ci de la distribution des entrées entre les deux entraînements. Pour résoudre ce problème, on étudie les différences entre les distributions de chaque jeu de données pour trouver une transformation qui rapproche les deux distributions. Par exemple, un premier réseau de neurones a été entraîné à une tâche de classification sur des images issues d'un appareil photo standard et on veut entraîner

un nouveau modèle pour faire la même tâche sur des images d'une caméra grand angle. Les deux appareils déforment de manière différente les images et présentent une distribution des couleurs différente. Il convient alors de trouver une transformation qui permet de rapprocher les variations du premier jeu de données de celles du second jeu de données.

2.3.2.6 Auto-apprentissage et tâche prétexte

La difficulté pour adapter une solution d'apprentissage profond peut être de devoir dédier une grande quantité de son travail à l'annotation et la collecte de nouveaux exemples pour constituer un jeu de données dédié conséquent. La question qui revient régulièrement est donc quelle est la quantité de données nécessaire pour répondre correctement à un problème donné. C'est une question à laquelle nous essaierons de répondre dans notre cas spécifique dans l'annexe A.

Une méthode complémentaire a été proposée pour permettre au réseau d'adapter le réseau au domaine du nouveau jeu de données ou d'initialiser les poids du réseau. On appelle cette approche l'auto-apprentissage. Cette approche ne permet pas de se débarrasser de l'annotation mais sert d'étape préliminaire à un apprentissage supervisé classique. Cette approche est très proche des techniques d'apprentissage non supervisé.

L'idée est d'exploiter la structure des données elle-même pour permettre au réseau de se "familiariser" avec le nouveau domaine du jeu de données. Pour cela, nous allons définir une tâche prétexte qui va nous permettre d'adapter le réseau au domaine du jeu de données puis nous utiliserons ce réseau pré-entraîné sur la tâche réelle visée.

Un premier type de tâche prétexte consiste à exploiter la structure des données. Sur des images naturelles, on peut facilement, en tant qu'être humain, comprendre la position relative de deux parties d'un objet. Pour apprendre au réseau cette structure, on peut ainsi extraire de manière aléatoire deux patchs (images rognées extraites de l'image) et lui demander de les positionner relativement l'une par rapport à l'autre. C'est l'idée proposée par Doersch dans [31]. Les données permettant d'entraîner cette tâche ne nécessitent aucune annotation car on connaît déjà la position relative des deux patchs lors de leur création.

Une fois le réseau entraîné sur cette tâche, on récupère les poids de la partie du réseau qui extrait les caractéristiques des patchs pour initialiser les poids du réseau pour la tâche réelle.

Un autre type de tâche consiste à dégrader les données d'entrées et demander au réseau de retrouver l'état d'origine de ces données. Des exemples de détérioration sont : transformer une image couleur en niveau de gris [64, 129], effacer une zone portion de l'image [93], sous-échantillonner une image.

2.3.3 Type de réseaux de neurones

La notion de réseau formel est tirée de la modélisation simplifiée d'un neurone biologique. La métaphore de réseaux de neurones reprend l'idée que les différents neurones sont organisés par un réseau de connexions qui sont modifiées au cours de la phase d'apprentissage et que l'on peut représenter par un graphe.

Certains algorithmes précurseurs d'apprentissage avaient l'intention de reproduire une forme d'apprentissage du point de vue biologique. Ces modèles ont ainsi

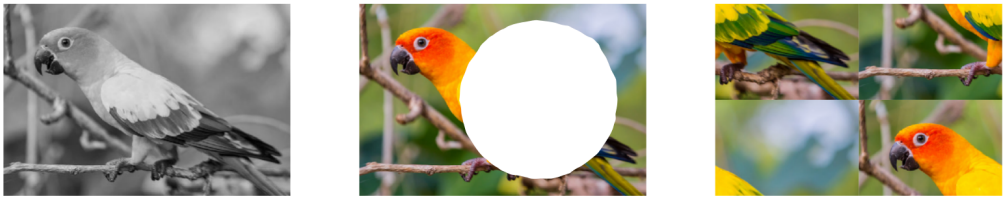


FIGURE 2.2 – Exemples de tâche prétexte en détériorant l'image d'entrée. (Gauche) L'image est transformée en niveau de gris, le réseau doit retrouver les couleurs originales de l'image. (Milieu) Une partie de l'image a été effacée, le réseau doit estimer la partie effacée. (Droite) L'image originale a été séparée en plusieurs morceaux, le réseau doit replacer correctement chaque patch (Image extraite de la présentation de Andrea Vevaldi à MISS2018 [114]).

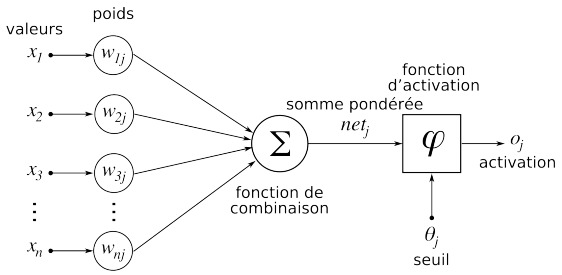


FIGURE 2.3 – Modélisation d'un neurone formel. Le neurone calcule la somme pondérée par les poids w_i des entrées x_i. Cette valeur passe ensuite par une fonction d'activation pour produire une sortie qui est transmise à l'entrée du neurone suivant (©Wikipédia).

exploré certains aspects de neurosciences et biologie pour tenter d'imiter des processus du cerveau ou de la biologie animale. Ce parallèle s'arrête là dans le sens où il est faux de manière générale de dire que ces méthodes sont des modélisations réalistes de fonctions biologiques.

2.3.3.1 Réseau de neurones classique

On appelle réseau de neurones une combinaison de couches de neurones. Une couche est un ensemble d'entités qui forme un bloc autonome et qui sont connectées aux mêmes entrées et qui sont dirigées vers les mêmes sorties.

Pour définir l'architecture d'un réseau de neurones classiques, il faut définir combien de couches comporte le réseau, combien de neurones par couche sont présents et comment ces différentes couches sont reliées entre elles. On appelle les couches intermédiaires, situées entre la couche d'entrée et la couche de sortie, les couches cachées, *hidden layers* en anglais.

Pour chaque neurone, la sortie est obtenue en faisant la somme des entrées, pondérées par leur poids respectif (voir figure 2.3). Le résultat final de la sortie passe alors par une fonction d'activation. Cette fonction d'activation est une fonction non-linéaire qui sert généralement de seuil. Une fonction d'activation très largement utilisée est la fonction ReLU, *REctified Linear Unit*, qui est formulée par la fonction $g(z) = \max(0, z)$. L'ajout de ces non-linéarités est un élément clé pour le fonctionnement des réseaux de neurones.

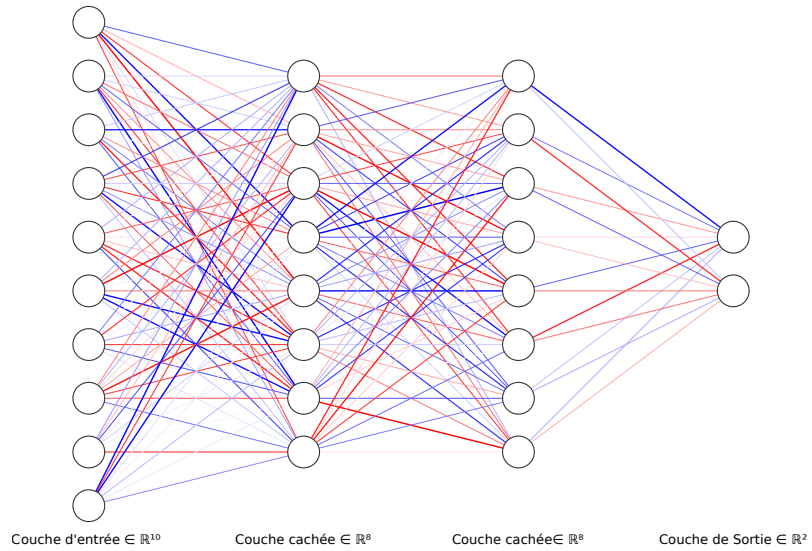


FIGURE 2.4 – Exemple d'un réseau de neurones. Ce réseau contient une couche d'entrée, deux couches cachées et une couche de sortie, représentées par des colonnes de cercles. Chaque cercle représente un neurone formel. Les liens entre deux neurones représente le poids de la connexion entre ces deux neurones. L'épaisseur des connexions traduit la valeur, la couleur traduit si le poids est négatif (bleu) ou positif (rouge). La valeur des poids des connexions a été établie lors de la phase d'apprentissage.

On peut représenter le passage d'une couche i à la couche $i+1$ par le produit matriciel définit par :

$$\mathbf{x}_{i+1} = \mathsf{W}\mathbf{x}_i = \begin{pmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,m} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,m} \\ \vdots & \vdots & & \vdots \\ w_{n,1} & w_{n,2} & \cdots & w_{n,m} \end{pmatrix} \cdot \begin{pmatrix} x_{i,1} \\ x_{i,2} \\ \vdots \\ x_{i,m} \end{pmatrix}. \tag{2.17}$$

Une couche d'un réseau de neurones traditionnel multiplie les entrées par une matrice de paramètres pour obtenir des sorties. Chaque connexion représente un élément de cette matrice de paramètres. On appelle ce type de couche, une couche complètement connectée, *fully connected layer* en anglais, puisque tous les neurones sont connectés à l'ensemble des entrées et des sorties.

Une des limites de ce modèle est que le nombre de paramètres augmentent considérablement avec le nombre de neurones utilisés. Cette approche est ainsi particulièrement inefficace pour le traitement d'images. En effet, en utilisant un neurone par pixel de l'image, on obtient très rapidement un nombre gigantesque de poids à optimiser.

2.3.3.2 Réseaux convolutifs

Un réseau convolutif, *Convolutional Neural Network* (CNNs) en anglais, est un type de réseaux de neurones spécialisé pour traiter des données sous formes de grilles. On peut penser à une série de données temporelles, que l'on peut interpréter comme une grille 1D ou une image que l'on représente par une grille 2D de pixels.

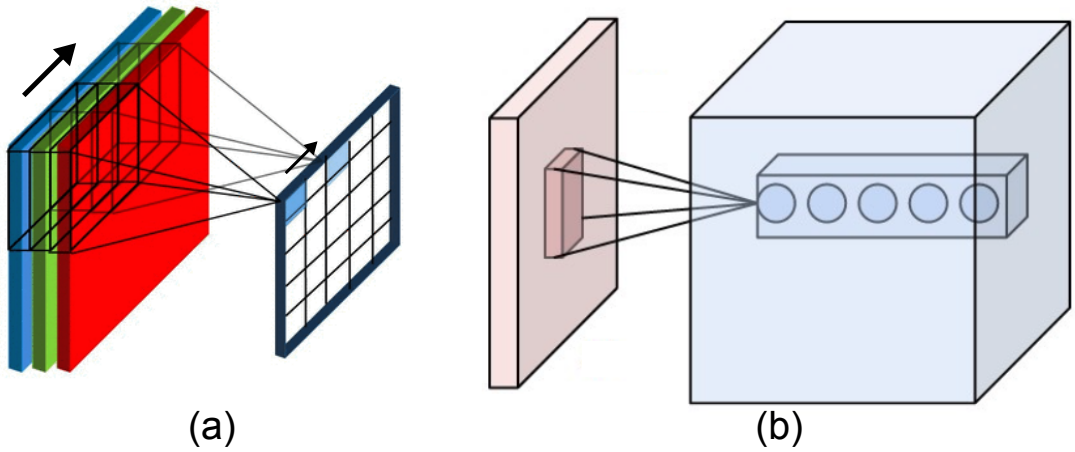


FIGURE 2.5 – Description de la couche convolutive. (a) On applique sur une portion de l'image d'entrée, représentée par 3 couches (représentant les canaux rouge, vert, bleu), le filtre de convolution (cadre noir). Le résultat de cette opération est stockée sur l'emplacement bleu sur la grille à droite. Cette opération est répétée en déplaçant le filtre sur toute l'image selon le pas utilisé. La position de l'application du filtre après 2 déplacements est affichée de manière plus claire. La taille du pas définit la taille de la grille obtenue. Un petit pas donne une grande grille et un grand pas, une petite grille. (b) La structure de gauche représente l'entrée avec la position actuelle du filtre de convolution comme illustrée dans (a). Le pavé bleu à droite représente le volume de sortie. Chaque cercle traduit le résultat d'un filtre. Le nombre de cercles représente ainsi le paramètre de profondeur de la couche convolutive, ici 5 (©Wikipédia).

L'opération de convolution est une opération déjà connue avant les méthodes d'apprentissage profond. La fonction de convolution entre le vecteur $\mathbf{x} \in \mathbb{R}^n$ et le vecteur $\mathbf{w} \in \mathbb{R}^m$ est donnée par :

$$\mathbf{y}_i = (\mathbf{x} * \mathbf{w})_i = \sum_j w_j x_{i-j}, \tag{2.18}$$

pour tout i tel que $\mathbf{x}_{i-j}$ soit défini. On appelle $\mathbf{w}$ le noyau de la convolution.

En 2D, la convolution est notamment utilisée pour appliquer des filtres sur une image, pour la détection de contours par exemple. L'idée est d'utiliser cette opération pour remplacer une couche de neurones. Les poids nécessaires pour cette couche se résument ainsi à la taille du noyau qui est partagé pour toutes les entrées. Les dimensions de la sortie d'une couche convolutive sont définies par trois hyperparamètres : la profondeur, le pas et la marge. La profondeur définit le nombre de noyaux utilisés. La sortie contient ainsi le résultat de la convolution avec chacun de ces noyaux. Le pas contrôle le chevauchement des champs récepteurs. Plus le pas est petit, plus les champs récepteurs se chevauchent et plus le volume de sortie sera grand. La marge, *padding* en anglais, consiste à compléter les frontières de la grille d'entrée par des pixels noirs (qui valent 0). Toutes ces options permettent de contrôler les dimensions du volume de sortie de la couche convolutive.

Un réseau convolutif est par définition un réseau qui possède des couches convolutives, *convolution layers* en anglais. En pratique, une couche convolutive est combinée avec une couche d'activation, comme dans un réseau classique, et une couche de

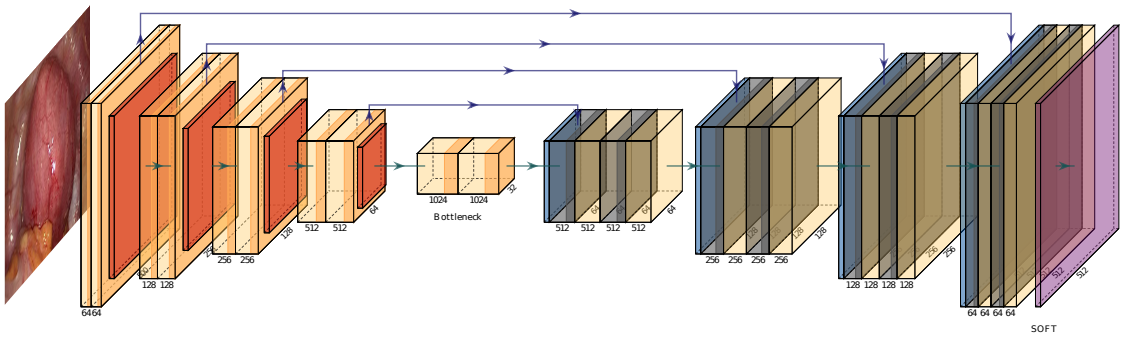


FIGURE 2.6 – Illustration du réseau U-Net [101] qui est une figure emblématique des architectures encodeur-décodeur, ou en forme de U. L'illustration a été réalisée avec PlotNeuralNetwork. La partie orange du réseau est l'encodeur qui reçoit l'image d'entrée à gauche, la zone bleue est le décodeur qui produit la sortie à droite, avec la même résolution que l'entrée. Les flèches qui partent de l'un vers l'autre représentent les *skip-connections*.

regroupement (*pooling* en anglais). La couche de regroupement consiste à réduire les dimensions de l'image en réalisant une forme de sous-échantillonnage. Une méthode de regroupement très largement utilisée est le *max pooling* qui consiste à conserver le maximum dans un voisinage rectangulaire. Cette opération permet de rendre la représentation de l'image invariante à de petites translations. L'ajout de cette étape est particulièrement important lorsqu'on cherche plus à vérifier si une caractéristique (*feature*) est présente plutôt que de savoir exactement où elle se trouve. Cette étape réduit la quantité de paramètres et de calcul dans le réseau ce qui rend le réseau plus facile à entraîner.

Les réseaux totalement convolutifs, c'est-à-dire des réseaux qui n'utilisent pas de couche totalement connectée, ont aujourd'hui montré leurs avantages pour traiter des images.

Un nouveau type de d'architecture a été proposé pour obtenir en sortie une grille de la même résolution que l'image d'entrée. Cette approche a trouvé son intérêt pour des tâches qui s'intéressent à chaque pixel comme la segmentation sémantique [103, 101]. On appelle ce type de structure encodeur-décodeur. La première partie du réseau correspond à la partie usuelle d'un réseau qui extrait les caractéristiques de l'image d'entrée. La seconde partie du réseau est construite en miroir de la partie encodeur et vise à retrouver la résolution initiale de l'image. Cette seconde partie réalise donc une forme de sur-échantillonnage. Renverser simplement la structure de départ de l'encodeur ne suffit pas. En effet, les détails de l'image originale sont perdus à mesure que l'on progresse dans les couches du réseau. Pour cela, il est nécessaire de connecter les différentes couches du décodeur avec leurs couches correspondantes de la partie encodeur. Ces connexions, appelées *skip-connections* en anglais, permettent notamment d'inverser l'étape de regroupement réalisée dans les couches convolutives.

Le réseau U-Net [101] est un exemple emblématique d'architecture encodeur-décodeur particulièrement utilisée pour des tâches de segmentation sémantique.

2.3.3.3 Réseaux récurrents

Les réseaux de neurones récurrents (RNNs) sont une famille de réseaux de neurones pour traiter des données séquentielles. De la même manière que les CNNs sont adaptés pour traiter des données sous forme de grille comme les images, ces réseaux sont spécialisés pour gérer des séquences de valeurs. L'idée est que ce type de réseau va prendre en compte les entrées précédemment utilisées et éventuellement leurs prédictions. Ce type de réseau est très largement utilisé pour le traitement de texte, par exemple pour faire de la traduction automatique. On peut également l'utiliser pour le traitement de vidéos, pour de la reconnaissance d'actions par exemple.

Ce type de réseaux ne sera pas utilisé dans le cadre de nos travaux même si ceux-ci ont été envisagés pour traiter le problème de la détection d'incision qui nécessite une détection à la fois spatiale (localiser les pixels sont associés à la classe qui correspond à l'incision) et temporelle (localiser à quel moment l'incision apparaît dans l'image).

2.3.4 Fonctionnement de la phase d'apprentissage

2.3.4.1 Jeux d'entraînement, de validation et de test

Nous avons parlé précédemment de l'importance du jeu de données pour réaliser l'entraînement d'un modèle d'apprentissage. Il est toutefois nécessaire de séparer ce jeu de données en plusieurs catégories. La première catégorie est le jeu d'entraînement ou d'apprentissage. C'est sur ces exemples que l'algorithme va tester ses prédictions et évaluer son erreur pour mettre à jour ses paramètres.

La deuxième catégorie est le jeu de validation. Les exemples de ce jeu de données vont être utilisés pour évaluer l'erreur de l'algorithme sur des images qui ne sont pas utilisées pour optimiser ses paramètres. De manière régulière, on mesure ainsi l'erreur moyenne obtenue sur le jeu de validation. Cette erreur est bien souvent supérieure à l'erreur moyenne du jeu d'entraînement puisque ce n'est pas sur ces exemples que l'algorithme minimise son erreur. L'intérêt est de comparer l'évolution de cette erreur par rapport à l'évolution de l'erreur d'entraînement. Si ces deux erreurs diminuent, cela signifie que les améliorations de l'algorithme sur le jeu d'entraînement se généralisent au jeu de validation. Si l'erreur de validation augmente tandis que l'erreur d'entraînement diminue, cela signifie que les améliorations ne sont valables que sur les exemples du jeu d'entraînement. On parle alors de sur-apprentissage, ou *overfitting* en anglais. Lorsque ce phénomène apparaît, il est coutume de stopper la phase d'entraînement.

La dernière catégorie est le jeu de test. C'est un jeu de données utilisé pour fournir une évaluation impartiale de la méthode d'apprentissage. Ce jeu permettra de donner les résultats de l'entraînement sur de nouvelles images.

La répartition des exemples entre ces différents jeux de données est une question délicate. Avec moins de données d'entraînement, les paramètres optimaux théoriques du modèle d'apprentissage ont moins de chances d'être obtenus. Avec moins de données de test et de validation, il devient difficile de savoir si les performances de le jeu d'entraînement se généralise correctement. Il n'y a pas de règle optimale pour définir la répartition des données. Il est d'usage d'utiliser une distribution proche de 80%, 10%, 10% respectivement pour le jeu d'entraînement, de validation et de test.

En fonction des utilisations, il peut être utile de vérifier que les jeux de données

ont une distribution homogène des propriétés. Par exemple, il est important de vérifier que la répartition des classes entre le jeu d'entraînement n'est pas trop éloignée de celle des autres jeux.

2.3.4.2 Descente de Gradient Stochastique

Nous avons déjà présenté la méthode de descente de gradient précédemment. Pour entraîner un réseau de neurones, nous allons utiliser une méthode dérivée.

Appliquer l'approche classique de la descente de gradient nécessite de réaliser la prédiction de l'ensemble des observations du jeu d'apprentissage, calculer la moyenne des résidus avec la fonction de coût puis estimer le gradient de la moyenne des résidus. En pratique, cela n'est pas réalisable.

L'astuce qui a été trouvée est de réaliser la prédiction sur un sous-ensemble du jeu de données sélectionné aléatoirement, appelé lot, *batch* en anglais, de calculer le résidu et estimer le gradient sur ce sous-ensemble. On utilise le gradient calculé sur ce sous-ensemble pour modifier les poids du réseau. On répète cette étape sur d'autres lots jusqu'à avoir vu toutes les observations du jeu d'entraînement. On appelle le fait d'avoir fait l'itération des lots sur l'ensemble du jeu d'entraînement une époque.

2.3.4.3 Fonction de coût

La fonction de coût, *loss function* en anglais, est la fonction qui permet d'évaluer l'erreur entre les prédictions du réseau et les labels. Pendant l'entraînement, cette fonction donne une valeur de résidu dont le suivi va nous permettre de vérifier que le réseau améliore ses prédictions. Au cours de l'entraînement on minimise cette fonction sur les exemples du jeu d'entraînement. C'est de cette fonction que va être calculé le gradient qui va être rétro-propagé dans chaque couche du réseau pour mettre à jour les poids correspondants.

La fonction de coût par défaut dans la plupart des solutions d'apprentissage profond est l'Entropie Croisée (*Cross Entropy* en anglais). Sa formule est donnée par :

$$\mathcal{L}_{EC}(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{i \in \{1,n\}} \mathbf{y}_i \log(\hat{\mathbf{y}}_i), \tag{2.19}$$

avec n le nombre de classe ; $\mathbf{y} \in \{0,1\}^n$ le vecteur label pour lequel $y_i = 1$ quand i correspond à la classe à détecter et 0 ailleurs ; $\hat{\mathbf{y}} \in \mathbb{R}^n$ le vecteur de sortie du réseau.

L'autre fonction de coût très utilisée est l'Entropie Croisée Binaire (*Binary Cross-Entropy* en anglais) dont la formule est :

$$\mathcal{L}_{ECB}(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{i \in \{1,n\}} \mathbf{y}_i \log(\hat{\mathbf{y}}_i) + (1 - \mathbf{y}_i) \log(1 - \hat{\mathbf{y}}_i). \tag{2.20}$$

Pour ces deux fonctions de coût, on peut constater qu'elles ne sont définies que pour $\hat{y}_i \in]0,1[$. En pratique, la dernière étape du réseau de neurones convertit sa sortie pour avoir une sortie qui correspond au format de la réponse attendue. Dans notre cas, on voudrait avoir une sortie de réseau qui puisse être interprétée comme la probabilité de chaque classe prédite par le réseau.

On utilise pour cela la fonction *softmax* qui est donnée par :

$$h(\mathbf{z})_i = \frac{\exp(\mathbf{z}_i)}{\sum_j \exp(\mathbf{z}_j)}, \tag{2.21}$$

ou la fonction *sigmoïde* donnée par :

$$\sigma(\mathbf{z}_i) = \frac{1}{1 + \exp(-\mathbf{z}_i)}, \tag{2.22}$$

avec $\mathbf{z}$ le vecteur sortie du réseau.

On utilise plus souvent la fonction sigmoïde pour un problème de classification binaire où la sortie du réseau est un scalaire. La fonction *softmax* est utilisée dans le cas d'une classification multi-classes.

L'entropie croisée simple prend en compte seulement à quelle point la probabilité de la classe à prédire est différente de 1. Tandis que l'entropie croisée binaire mesure aussi à quel point les autres classes sont différentes de 0. La première est utilisée pour la classification multi-classes en combinaison avec la fonction *softmax* et la seconde pour la classification binaire avec la fonction sigmoïde.

2.3.5 Régularisation

Un enjeu majeur des méthodes d'apprentissage statistique est de proposer un modèle qui fonctionne correctement sur des nouvelles entrées, c'est-à-dire sur lesquelles il n'a pas été spécifiquement entraîné. On appelle cette propriété la généralisation. La régularisation décrit les méthodes qui permettent d'améliorer la généralisation d'un algorithme d'apprentissage. Ces contraintes peuvent être conçues pour encoder un a priori. Ces méthodes peuvent également consister à préférer le modèle le plus simple. On appelle la capacité d'un modèle sa faculté à s'adapter à un grand nombre de fonctions. Les modèles avec une capacité trop faible peuvent avoir du mal à s'adapter au jeu d'entraînement. Les modèles avec une grande capacité vont avoir tendance à retenir des propriétés supplémentaires spécifiques au jeu d'entraînement qui ne seront pas utiles sur de nouvelles données. Par la suite, nous parlerons en particulier de méthodes utilisées pour l'apprentissage profond.

2.3.5.1 Pénalité sur la norme des poids

Une approche très utilisée pour limiter la capacité du modèle consiste à ajouter une pénalité sur les paramètres pour compléter la fonction de coût. On note ce terme de régularisation $\Omega(\boldsymbol{\theta})$ qui prend en entrée l'ensemble des paramètres du réseau noté $\boldsymbol{\theta}$. L'impact de ce terme de régularisation est contrôlé par un coefficient α. Pour les approches avec réseaux de neurones, cette régularisation est souvent appliquée aux poids du réseau $\mathbf{w}$ uniquement et non au biais. En pratique, la régularisation sur les biais contraint trop l'apprentissage.

La régularisation des paramètres avec la norme l2, appelée décroissance des poids, *weight decay* en anglais, consiste à ajouter le terme de régularisation $\Omega(\boldsymbol{\theta}) = \frac{1}{2}\|\mathbf{w}\|^2$ à la fonction de coût. Lors de l'apprentissage, cette contrainte va retrancher le produit du pas de la descente du gradient avec le coefficient α à la norme des poids du réseau. En pratique, on choisit α faible.

2.3.5.2 Augmentation du jeu de données

L'augmentation du jeu de données, *dataset augmentation* en anglais, est une méthode pour augmenter virtuellement la taille du jeu d'entraînement pour améliorer

l'apprentissage, notamment vis-à-vis de la capacité de généralisation. L'idée est d'altérer les données d'entrées, et éventuellement les labels si besoin, en appliquant des transformations. On crée virtuellement de nouveaux exemples pour l'entraînement. Cette approche rejoint le problème d'adaptation de domaine.

Dans le cas des images, on a souvent tendance à appliquer des rotations, des translations, des rognages aléatoires pour s'assurer que l'apprentissage n'est pas dépendant du cadrage des exemples. On peut également appliquer des filtres comme le flou gaussien, ou du bruit blanc.

Cette approche est toutefois limitée : d'une part, les altérations ne doivent pas être trop importantes au risque de rendre l'image d'entrée incompréhensible. D'autre part, ces exemples dégénérés ne remplacent pas des nouveaux exemples radicalement différents. Le domaine des augmentations n'est pas le même que le domaine réel des données d'entrées.

2.3.5.3 Dropout

Le dropout [107] est une méthode puissante pour régulariser de nombreux types de modèles d'apprentissage tout en étant peu coûteuse en termes de ressources de calcul. Rappelons qu'un réseau est un graphe d'unités interconnectées les unes aux autres. Le dropout consiste à temporairement retirer une unité, autre que celle de la couche de sortie, du réseau pendant une étape de l'apprentissage. Pour chaque itération, les unités ont une certaine probabilité d'être retirées temporairement du réseau. Concrètement, l'architecture du réseau est légèrement différente à chaque itération. Le dropout force le réseau à développer son raisonnement sur plusieurs parties de son architecture ce qui est bénéfique pour la généralisation.

2.4 Description du logiciel Uteraug

Le sujet de thèse a été formulé autour du développement du logiciel de réalité augmentée développé au sein de l'équipe EnCoV. Ce logiciel, appelé Uteraug, permet de visualiser des tumeurs par effet virtuel de transparence pendant une chirurgie. Le logiciel réalise notamment le recalage non rigide entre les données préopératoires et les images cœlioscopiques. Ce logiciel a été le sujet de plusieurs publications scientifiques et cliniques [24, 25, 23, 26] décrivant ces différents perfectionnements.

Comme les sujets abordés durant nos travaux concernent l'amélioration de plusieurs limites de ce logiciel, il paraît nécessaire de décrire le fonctionnement de ses étapes clés. Les explications actuelles peuvent ne pas refléter exactement l'état actuel du logiciel.

L'ensemble des étapes du logiciel est illustré sur la figure 2.7.

2.4.1 Phase préparatoire

Avant le jour de l'opération, il est nécessaire pour le logiciel de disposer des données préopératoires qui permettent de décrire l'organe et la localisation des tumeurs. Dans le protocole existant, une image volumique a été obtenue par Tomodensitométrie (TDM) ou Imagerie par Résonnance Magnétique (IRM) dans le but de faire un diagnostic et d'évaluer l'intérêt d'une intervention. La prise d'image par TDM

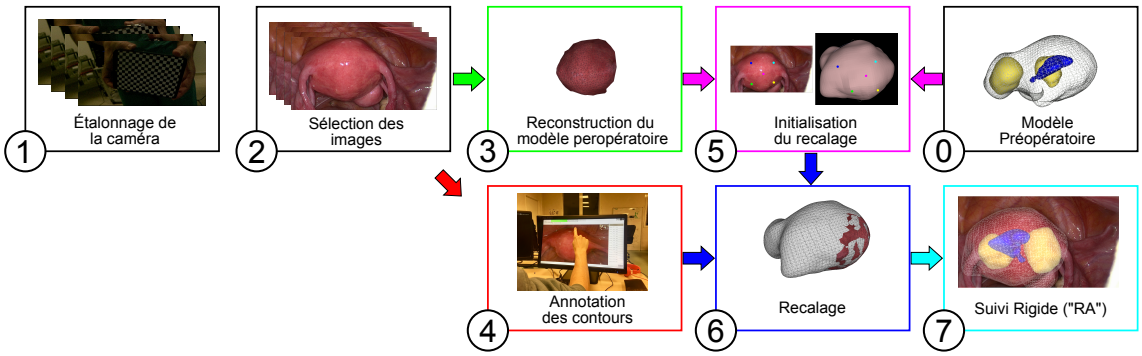


FIGURE 2.7 – Articulation des étapes du logiciel Uteraug.

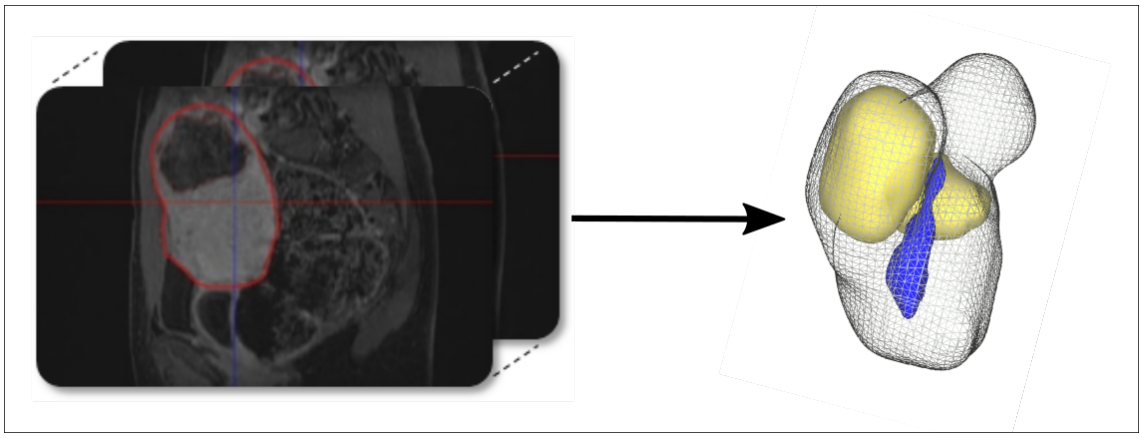


FIGURE 2.8 – Illustration de la conversion d'un fichier DICOM vers des modèles 3D segmentés. (à gauche) Le contour de l'organe est surligné en rouge, la zone sombre dans cette région surlignée est une tumeur. Les traits hachurés traduisent la multitude des couches (*slice* en anglais) de l'IRM. (à droite), la conversion en modèles 3D des éléments segmentés sur les couches de l'IRM. Les tumeurs sont représentées en jaune, la cavité utérine en bleu et la surface extérieure de l'organe en gris.

consiste à mesurer l'absorption des rayons X par les différents tissus. L'IRM utilise un champ magnétique puissant et mesure la réaction des tissus.

Cette image volumique décrit l'ensemble des éléments visibles dans un certain volume. Il est nécessaire de localiser les structures d'intérêt, dans notre cas, l'utérus et les myomes qui se trouvent à l'intérieur, et de les segmenter. Cette étape est réalisée en pratique, dans le cadre de nos recherches, grâce au logiciel MITK [1]. Ce logiciel open-source permet de segmenter de manière semi-automatique.

Cette étape reste laborieuse et en pratique, peu de personnes peuvent réaliser ce travail à la place des experts médicaux, chirurgiens ou radiologues. Des solutions basées sur l'apprentissage profond sont en plein essor depuis plusieurs années pour réaliser la segmentation de manière complètement autonome. Les travaux de cette thèse ne portent toutefois pas sur ce problème.

1. http://mitk.org

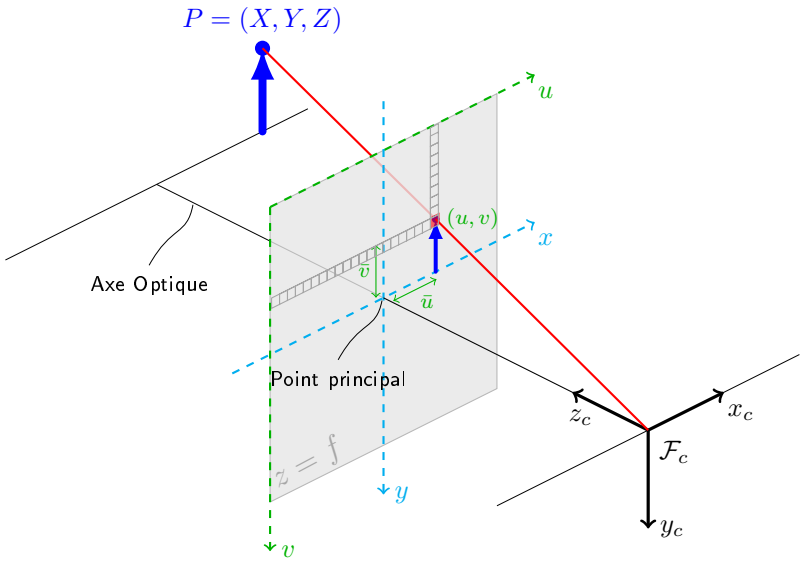


FIGURE 2.9 – Projection perspective. Cette illustration présente les différentes transformations de changement de repères qui permet de convertir les coordonnées 3D dans le repère monde vers les coordonnées 2D de l'image capturée par la caméra.

2.4.2 Étalonnage de la caméra

L'étalonnage de la caméra, *camera calibration* en anglais (l'anglicisme calibration ou calibrage est parfois utilisé), permet de définir les paramètres intrinsèques de la caméra. C'est un élément essentiel pour établir le modèle qui permet de transformer le point d'un objet dans le monde physique en trois dimensions, appelé point objet, vers le point correspondant dans l'image qui a été capturé par une caméra, appelé point image. Pour comprendre l'intérêt de cette étape, nous allons présenter le modèle qui permet de réaliser cette transformation et également le modèle de correction qui permet de prendre en compte la déformation de la lentille de la caméra utilisée.

2.4.2.1 Caméra perspective

Dans le cadre de nos travaux, nous utilisons le modèle de caméra perspective en utilisant la convention du modèle sténopé (*pinhole* en anglais). Cette convention suppose que l'ensemble des rayons lumineux passe par un unique point avant d'atteindre le capteur. La figure 2.9 illustre les différents repères et transformations associées. On appelle rayon de projection d'un point 3D, la droite qui relie le point 3D et le centre de la caméra. On appelle projection perspective l'ensemble des changements de repère avec un jeu de paramètres associés qui permet de faire correspondre un point de l'espace $\mathbf{P}$ avec son image $\mathbf{p}$ dans le repère image.

Les différentes étapes de la projection perspective sont :

— Le point de l'espace $\mathbf{P}$ est exprimé dans les coordonnées liées au repère de la caméra. Ce changement de repère est défini par les paramètres extrinsèques de la caméra.

— La projection centrale : cette transformation permet de traduire le point 3D, exprimé dans le repère caméra, au point d'intersection du rayon de projection sur le plan du capteur.

— La dernière étape traduit le changement du repère de la rétine du capteur

(coordonnées en mm) vers l'image (en pixels). Cette transformation est définie par les paramètres intrinsèques de la caméra.

La projection perspective est décrite par une fonction $\mathbb{R}^3 \mapsto \mathbb{R}^2$ que l'on représente par une matrice F de dimension (3×4). La projection d'un point du repère monde vers le repère image s'écrit sous la forme :

$$\mathbf{p} = \mathsf{F} \begin{pmatrix} X & Y & Z & 1 \end{pmatrix}^\top, \tag{2.23}$$

avec

$$\mathsf{F} = \mathsf{K} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \mathsf{R} & \mathbf{t} \\ 0_{(1\times3)} & 1 \end{pmatrix}, \tag{2.24}$$

avec K la matrice d'étalonnage de la caméra (matrice (3×3)), $\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$ la matrice de projection centrale, et $\begin{pmatrix} \mathsf{R} & \mathbf{t} \\ 0_{(1\times3)} & 1 \end{pmatrix}$ la matrice de pose de la caméra.

2.4.2.2 Paramètres extrinsèques.

Les paramètres extrinsèques définissent la matrice de pose de la caméra dans le repère monde. La pose de la caméra possède six degrés de liberté exprimés par un vecteur de rotation $\mathbf{r} = (\psi, \theta, \phi)^\top$ et un vecteur de translation $\mathbf{t} = (t_1, t_2, t_3)^\top$. Le vecteur $\mathbf{t}$ représente la position 3D du centre optique et le vecteur de rotation $\mathbf{r}$ l'orientation de la caméra depuis le centre optique, décomposé en 3 angles nommés roulis ou précession, tangage ou nutation et lacet ou rotation propre. En pratique, on utilise la matrice de rotation R obtenue à partir de $\mathbf{r}$ par :

$$R = \begin{pmatrix} \cos\psi\cos\phi - \sin\psi\cos\theta\sin\phi & -\cos\psi\sin\phi - \sin\psi\cos\theta\cos\phi & \sin\psi\sin\theta \\ \sin\psi\cos\phi + \cos\psi\cos\theta\sin\phi & -\sin\psi\sin\phi + \cos\psi\cos\theta\cos\phi & -\cos\psi\sin\theta \\ \sin\theta\sin\phi & \sin\theta\cos\phi & \cos\theta \end{pmatrix}. \tag{2.25}$$

Ces paramètres permettent d'établir les changements de repères monde/caméra, exprimés par la formule :

$$\mathbf{P}_\mathcal{W} = \begin{pmatrix} \mathsf{R} & \mathbf{t} \\ \mathbf{0_{1\times3}} & 1 \end{pmatrix} \mathbf{P}_\mathcal{C} \tag{2.26}$$

$$\mathbf{P}_\mathcal{C} = \begin{pmatrix} \mathsf{R}^\top & -\mathsf{R}^\top\mathbf{t} \\ \mathbf{0_{1\times3}} & 1 \end{pmatrix} \mathbf{P}_\mathcal{W}, \tag{2.27}$$

avec $\mathbf{P}_\mathcal{W}$ un point 3D dans le repère monde et $\mathbf{P}_\mathcal{C}$ le point correspondant exprimé dans le repère caméra.

La rétroprojection peut être vue comme l'opération inverse de la projection. Le but est de donner la position 3D dans le repère monde d'un point 2D dans le repère image. Seulement l'opération inverse de la projection ne permet que de trouver le rayon optique et par le point image $\mathbf{p}$. La position du point 3D correspond au point $\mathbf{p}$ est donc définie à un facteur λ près, qui traduit la profondeur du point :

$$\mathbf{P}(\lambda) = \lambda \mathsf{F}^\dagger \mathbf{p}, \tag{2.28}$$

avec $\mathsf{F}^\dagger$ la pseudo-inverse de la matrice F

FIGURE 2.10 – Illustration du phénomène de distorsion (©opencv.org).

2.4.2.3 Paramètres intrinsèques.

Les paramètres intrinsèques définissent les propriétés géométriques de l'image de la caméra. La matrice d'étalonnage K peut alors s'exprimer sous la forme :

$$\mathsf{K} = \begin{pmatrix} f_x & s_{uv} & c_u \\ 0 & f_y & c_v \\ 0 & 0 & 1 \end{pmatrix}, \tag{2.29}$$

avec f_x et f_y correspondent à la distance focale de la caméra exprimée en largeur et hauteur de pixels, c_u et c_v correspondent au centre optique de la caméra sur le plan image. s_{uv} représente l'obliquité (*skew* en anglais) qui permet de modéliser les situations où le capteur ne forme pas un rectangle exact. On suppose que les pixels sont carrés donc que s_{uv} est nul. Cette hypothèse est réaliste avec les caméras modernes.

2.4.2.4 Distorsion optique

En optique, la distorsion est une aberration géométrique lorsque les conditions menant à l'approximation de Gauss ne sont plus respectées.

Les conditions de Gauss sont :

— Les angles d'incidence des rayons par rapport à l'axe optique sont faibles.
— Le point d'incidence est proche de l'axe optique.

L'approximation de Gauss permet alors de simplifier le calcul des angles.

Visuellement, cette aberration se concrétise par la déformation des lignes droites. On distingue la distorsion radiale de la distorsion tangentielle (voir figure 2.10). La distorsion radiale transforme les lignes droites en courbes et devient de plus en importante lorsqu'on s'éloigne du centre optique. Cet effet est particulièrement important sur les caméras grand-angle dite *fish eye*. La distorsion tangentielle est provoquée par un mauvais parallélisme entre la lentille et le plan de l'image. Elle provoque l'impression que certaines zones de l'image sont plus près qu'en réalité. Le

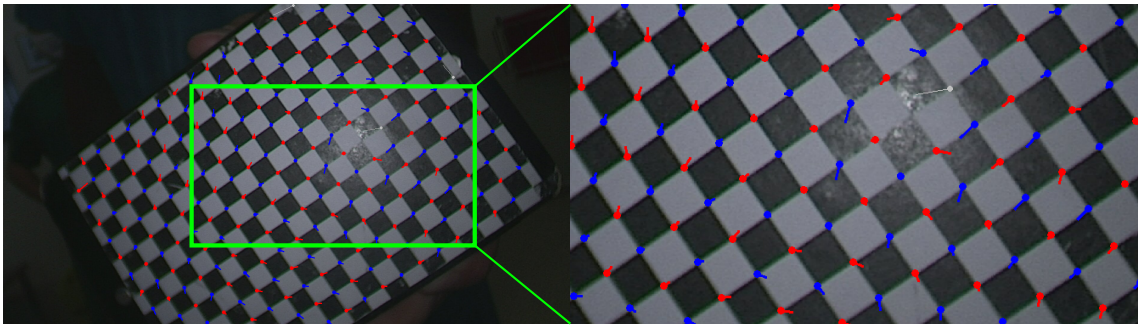


FIGURE 2.11 – Exemple d'une prise de vue de mire d'étalonnage. Les points de couleurs représentent les coins repérés par le système d'étalonnage. Les lignes (bleu et rouge) correspondent au décalage entre la position capturée et la position corrigée. Dans ce cas, la correction est faible puisque les lignes sont courtes.

modèle de Brown-Conrady a été proposé pour corriger ces distorsions :

$$x_u = x_d + (x_d - c_u)(k_1 r^2 + k_2 r^4 + \cdots) + (p_1(r^2 + 2(x_d - c_u)^2 + 2p_2(x_d - c_u)(y_d - c_v))(1 + p_3 r^2 + p_4 r^4 + \cdots) \tag{2.30}$$

$$x_u = x_d + (x_d - c_u)(k_1 r^2 + k_2 r^4 + \cdots) + (2p_1(x_d - c_u)(y_d - c_v) + p_2(r^2 + 2(y_d - c_v)^2))(1 + p_3 r^2 + p_4 r^4 + \cdots), \tag{2.31}$$

avec (x_d, y_d) les coordonnées du point dans l'image déformée, (x_u, y_u) les coordonnées du point dans l'image tel qu'il serait projeté avec un modèle sténopé parfait (sans distorsion), k_i le $i^{\text{ème}}$ coefficient de distorsion radiale, p_i le $i^{\text{ème}}$ coefficient de distorsion tangentielle, $r = \sqrt{(x - c_u)^2 + (y - c_v)^2}$.

2.4.2.5 Étalonnage de la caméra

La première étape du logiciel est d'étalonner la caméra, c'est-à-dire d'estimer les paramètres intrinsèques et les coefficients de distorsions. Pour ce faire, nous allons capturer plusieurs images d'une mire d'étalonnage, ou échiquier. C'est un objet parfaitement plat sur lequel est imprimé un pavage régulier de carrés noirs et blancs. Le système d'étalonnage repère les coins de ces motifs pour estimer les paramètres de distorsions et de la caméra (voir figure 2.11).

Cette étalonnage est réalisée grâce à la librairie d'OpenCV [2] Une fois cette estimation obtenue, on peut alors corriger la déformation de la caméra sur les images capturées pour la suite du logiciel.

2.4.3 Sélection d'images-clés

Cette étape consiste à capturer différents points de vue de l'organe. L'avantage de l'utérus est qu'il peut être mobilisé via une canule que l'on passe par le vagin. Cela permet d'obtenir des points de vue différents de la vue de face plus facilement.

Ces images-clés vont être utilisées d'une part pour reconstruire la partie visible de l'organe et d'autre part pour annoter les contours occultants de l'organe.

2. https://opencv.org/

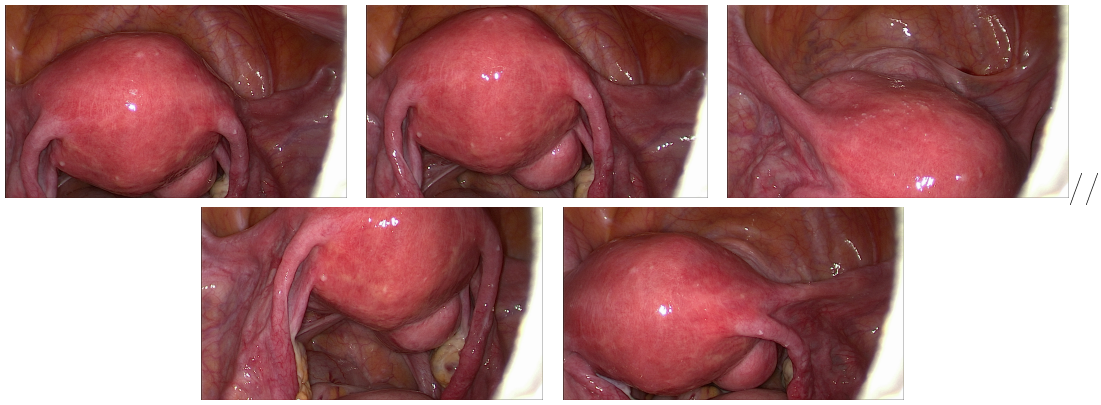


FIGURE 2.12 – Sélection d'images-clés de différents points de vues de l'utérus.

Cette étape est réalisée manuellement pour garantir que l'image soit nette et que les points de vues capturés soient différents les uns des autres. Pour chaque image-clé, un masque est annoté manuellement pour segmenter la partie visible de l'organe. Ce masque est utilisé pour l'étape de reconstruction 3D et pour la détection de points-clés.

Pour chaque image-clé, on récupère un ensemble de points-clés SIFT [76] qui sera utilisé pour la reconstruction 3D et le suivi de l'organe. SIFT, qui signifie *Scale-Invariant Feature Transform*, est un algorithme qui permet de détecter et d'identifier des éléments similaires entre deux images. La première étape de cet algorithme consiste à calculer des descripteurs SIFT pour les images à mettre en relation. Ces descripteurs caractérisent une zone d'intérêt de l'image. Ces descripteurs sont robustes aux changements d'échelle, de translation et de rotation. Chaque descripteur est associé aux coordonnées de cette zone pour former ce qu'on appelle un point-clé.

Pour comparer deux images, on va alors comparer l'ensemble de leurs descripteurs. Si deux descripteurs sont suffisamment similaires, ils sont alors associés. On appelle cette étape la mise en correspondance.

Il existe quelques stratégies pour rendre cette mise en correspondance plus robuste. On peut garder pour chaque point-clé d'une image les deux meilleures correspondances. On compare alors les distances respectives de ces correspondances avec le point-clé considéré. Si le rapport entre ces deux distances est trop faible alors on retire ce point-clé car il n'est pas assez discriminant. Cette méthode s'appelle le test de ratio de Lowe, *Lowe's Ratio Test* [77]. On peut également utiliser un critère géométrique en vérifiant que la transformation qui permet de déplacer le point-clé de la première image vers la seconde est cohérente avec les transformations des autres points-clés. Avoir un a priori sur la transformation entre les deux images est alors un avantage pour utiliser ce type de critère.

Dans le cadre du logiciel, ces points-clés seront utilisés d'une part pour l'étape de reconstruction 3D du modèle 3D peropératoire mais aussi lors de l'étape de suivi.

2.4.4 Annotation des contours

Pour contraindre l'étape de recalage (qui sera expliquée plus tard), nous allons utiliser les contours occultants de l'organe repérés sur les images-clés.

Un contour occultant est une frontière sur l'image entre la surface de l'objet considéré au premier plan et la surface cachée, ou occultée, derrière l'objet considéré.

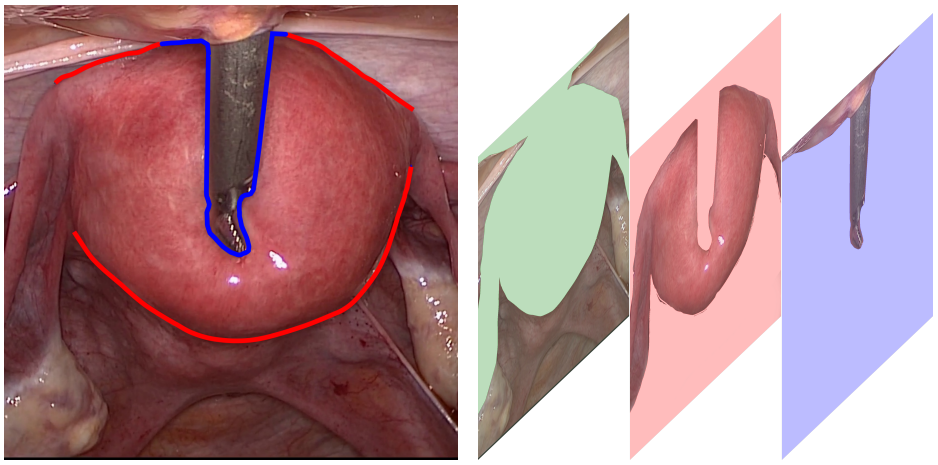


FIGURE 2.13 – (Gauche) Contours occultants de l'utérus (bleu) et contours d'occultation de l'utérus (vert) par un outil. (Droite) Illustration des différents plans d'occultation de la scène : l'outil (zone bleue) occulte l'utérus (zone rouge), qui a son tour occulte le reste de la cavité abdominale (zone verte).

En d'autres mots, ce sont les pixels visibles de la silhouette d'un objet (voir la figure 2.13).

Les contours occultants sont une donnée précieuse car ils matérialisent la limite de la forme tridimensionnelle de l'organe.

Ces contours sont actuellement annotés manuellement pendant l'opération grâce à un écran tactile. Cela nécessite une personne dédiée à cette tâche dans la salle d'opération. C'est l'étape manuelle la plus chronophage de la mise en place de la réalité augmentée. Il est donc crucial pour l'opérateur d'annoter à la fois rapidement et précisément.

C'est une limite majeure du logiciel pour son acceptabilité dans les hôpitaux. C'est une limite que nous avons décidé de lever grâce à nos travaux sur la détection automatique de contours occultants présentés dans le chapitre 3.

2.4.5 Reconstruction du modèle 3D peropératoire

Cette étape consiste à reconstruire un modèle 3D de la partie visible de l'organe. Nous allons utiliser une méthode appelée *Structure-from-Motion* (SfM) [109]. Cette méthode suppose que la scène visible dans les vues-clés est fixe et que la caméra a été étalonnée. Comme évoquée dans la section précédente, la scène n'est pas fixe puisque l'utérus est mobilisé pour observer des points de vue différents. En revanche, l'utérus est un organe relativement rigide et en particulier dans cette situation nous pouvons considérer que l'utérus ne se déforme pas.

Cette méthode consiste à déduire la structure 3D observée par les différentes images-clés proposées. A partir de deux images, on peut trouver les points images communs grâce à un détecteur de caractéristiques comme SIFT et les mettre en correspondance. Si suffisamment de points communs sont trouvés, on peut chercher la pose relative qui permet d'expliquer la transformation des points images d'une caméra vers l'autre. En reproduisant cette estimation sur l'ensemble des paires possibles, on obtient la pose relative de l'ensemble des caméras entre elles. On dit que les caméras sont alignées. On peut alors reconstruire la surface observée à partir des

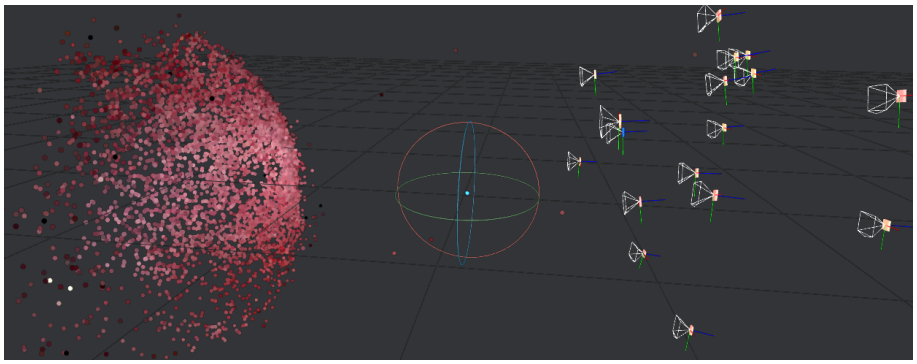


FIGURE 2.14 – Exemple de reconstruction Structure-from-Motion obtenue avec Meshroom [49]. Les poses des différents points de vue des différentes images sont représentées par les pyramides blanches sur la droite. Ici, seuls les points reconstruits sont représentés (à gauche) avant l'étape de création de maillage.

points objets communs entre les différentes prises de vues (voir la figure 2.14).

Dans un premier temps, l'algorithme fait une mise en correspondances des points clés calculés précédemment. Une triangulation est estimée entre chacune de ces correspondances pour trouver la pose relative des différents points de vue représentés par les différentes images-clés. Une fois les points de vues alignés entre eux, on peut densifier le nombre de correspondances en exploitant la géométrie des premières correspondances trouvées. Ensuite, on déduit la surface de cet ensemble dense de points pour définir un maillage 3D. Finalement, on ajoute une texture sur cette surface à partir des images-clés utilisées. En pratique, il peut être nécessaire de rogner manuellement le modèle 3D obtenu pour retirer les éléments proches de l'objet qui ont été reconstruits.

La reconstruction SfM est ainsi constituée des poses relatives entre les différentes images-clés, et du maillage 3D qui représente la surface visible depuis ces images-clés qui constitue le modèle 3D peropératoire.

Une fois le modèle 3D reconstruit, on associe les poses des caméras avec les points-clé SIFT acquis plus tôt. Pour chaque image-clé, on peut ainsi associer les points-clés aux points objets sur le modèle 3D reconstruit. Ce lien entre les points-clés sur l'image et les points objets sera exploité dans la dernière étape, pour le suivi de l'organe.

Uteraug utilise l'outil MeshRoom [49] et dans le cadre de cette thèse nous serons amenés à utiliser le logiciel Photoscan [3]. Photoscan est un logiciel propriétaire mais permet en pratique d'obtenir des reconstructions plus denses et de meilleure précision.

2.4.6 Recalage et suivi

Le recalage du modèle 3D préopératoire sur les images peropératoires est réalisé en deux temps : Dans un premier temps, un recalage non rigide est réalisé pour déformer le modèle 3D préopératoire et les structures internes repérées (tumeurs, cavité utérine, etc) en fonction des images peropératoires. On appelle cette étape le recalage initial. La seconde étape consiste à mettre à jour la pose du modèle 3D

3. https://www.agisoft.com

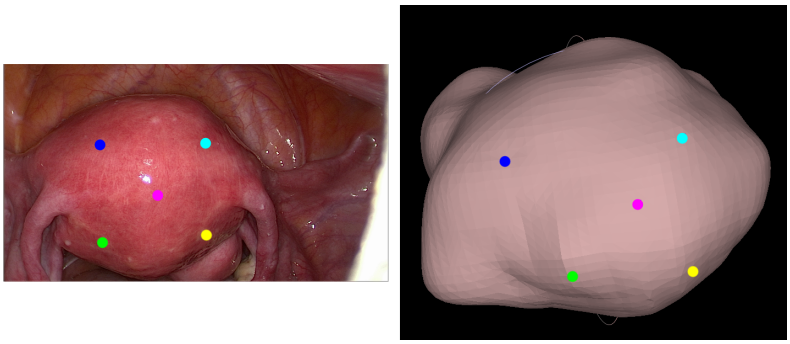


FIGURE 2.15 – Initialisation du recalage. Cette approche consiste à faire correspondre n points sur les modèles 3D préopératoire et peropératoire pour estimer une transformation rigide. Ici, nous avons sélectionné 5 points sur une image-clé et leurs points correspondants sur le modèle 3D préopératoire.

recalé dans la première étape pour correspondre à l'image actuelle. On appelle cette étape le suivi (*tracking* en anglais).

2.4.6.1 Initialisation du recalage

L'étape de recalage initiale nécessite une étape d'initialisation. Cette étape doit proposer une solution approchée pour permettre à l'optimisation de fonctionner correctement. On utilise une approche semi-automatique basée sur le problème *Perspective-n-Point* (PnP) qui consiste à trouver la pose d'une caméra 3D calibrée à partir d'un jeu de points objets et de leurs points images correspondants. Pour $n = \{2, 3\}$, il existe plusieurs solutions qui rendent le problème ambigu. Pour $n \geq 4$, on utilisera la version *Efficient PnP* [68] (EPnP).

Dans cette étape du logiciel, l'utilisateur doit marquer 5 correspondances d'une part sur une image-clé et d'autre part sur le modèle 3D préopératoire. On rappelle que grâce à la reconstruction 3D, on connaît la transformation entre chaque image-clé et le modèle 3D peropératoire. On obtient ainsi une transformation permettant de rapprocher le modèle 3D préopératoire du modèle 3D peropératoire.

2.4.6.2 Recalage non rigide 3D-3D

Cette étape est coûteuse en termes de ressources et prend un certain temps, de l'ordre de quelques minutes. Cette étape est réalisée une fois pour toute.

Le recalage est obtenu en minimisant une fonction d'énergie qui permet de rapprocher le modèle 3D préopératoire, un modèle 3D fermé sans texture représentant l'ensemble de la forme de l'utérus, du modèle 3D peropératoire, un modèle 3D qui représente la surface texturée visible depuis la cœlioscope. Le modèle 3D déformé est le modèle 3D préopératoire tandis que le modèle 3D peropératoire est constant. Cette fonction d'énergie est composée de plusieurs termes d'attaches aux données et d'un terme de régularisation. Le premier terme d'attache aux données est la distance entre les surfaces des deux modèles 3D. Le second terme d'attache aux données évalue la distance entre les contours occultants annotés et la silhouette du modèle 3D préopératoire déformé reprojeté dans les différentes vues utilisées pour reconstruire le modèle 3D peropératoire. Finalement, un terme de régularisation permet

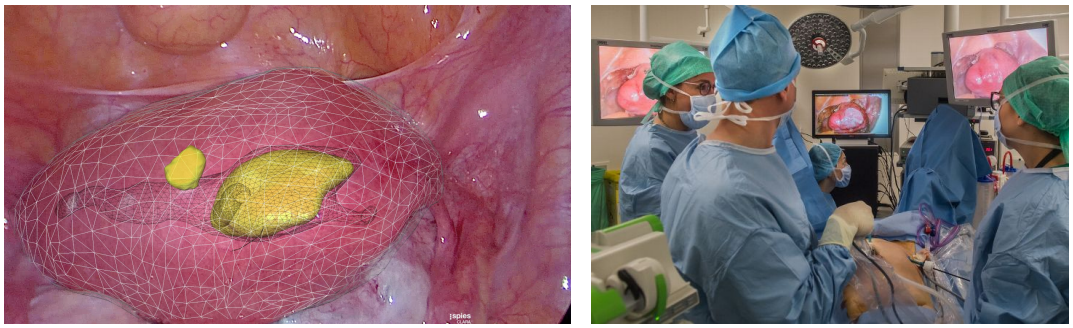


FIGURE 2.16 – (Gauche) Rendu de la réalité augmentée sur un utérus réalisée avec le logiciel Uteraug. Les tumeurs sont représentés en jaune, la grille blanche représente la surface extérieure du modèle 3D virtuel suivi. (Droite) Utilisation de la Réalité Augmentée dans les conditions réelles d'une salle d'opération (©Richard BRUNEL, La Montagne).

de contrôler cette déformation. Ce terme évalue l'énergie mécanique de déformation du modèle 3D pour limiter les déformations trop importantes du modèle 3D.

2.4.6.3 Suivi de l'organe

La seconde partie concerne le suivi de l'organe. Cette étape doit être rapide pour permettre un suivi en temps réel. Dans cette étape, l'utérus est à nouveau considéré comme rigide. Le suivi va donc se contenter de mettre à jour la pose de la caméra. Pour cela, l'image courante est mise en correspondance avec les image-clés stockées précédemment. On retient l'image-clé pour laquelle le plus grand nombre de points communs sont trouvés avec l'image courante. On estime la pose de la caméra qui explique le mieux la mise en correspondance avec RANSAC [36]. Si le nombre de points conservé après RANSAC [36] est inférieur à 8, on considère que l'image est trop différente des images-clés et le calcul de pose n'est pas mis à jour.

Finalement, la pose estimée est passée dans un filtre de Kalman pour lisser le suivi et éviter des à-coups. Un exemple de rendu final est présenté sur la figure 2.16.

L'utilisation de la réalité augmentée se limite à la phase préparatoire du chirurgien pendant laquelle l'hypothèse de rigidité de l'utérus est vérifiée. Durant cette phase, il peut repérer les structures internes avec la RA et prévoir la position de l'incision à réaliser.

Une fois l'incision débutée, l'organe se déforme jusqu'à rendre impossible la mise à jour de la pose, ou rendre le résultat inexploitable pour le chirurgien. L'hypothèse de rigidité de l'organe n'est plus vérifiée. C'est une limite du logiciel qui sera explorée dans le chapitre 4.

Chapitre 3

Détection de contours occultants de l'utérus

Dans ce chapitre, nous abordons le problème de la détection automatique de contours occultants de l'utérus. Les contours occultants correspondent aux pixels de la silhouette visible d'un objet. L'objectif de nos travaux est double : définir les propriétés que doivent vérifier un score dédié à l'évaluation d'un détecteur de contours et un algorithme de détection des contours occultants. À terme, nous voulons tester la capacité de notre détecteur de contours à remplacer l'étape d'annotation manuelle des contours occultants de l'utérus du logiciel Uteraug. Ce chapitre est basé sur notre contribution [39] publiée dans IJCARS 2020, pour le congrès international IPCAI 2020.

3.1 Introduction

La coelioscopie monoculaire augmentée nécessite le recalage d'un modèle 3D préopératoire avec les images de cœlioscopie. Comme le montre la figure 3.1, les systèmes de recalage de l'état de l'art [25, 5, 62, 91] s'appuient sur des repères visuels extraits d'images de cœlioscopie, en particulier les contours occultants de l'organe. Pour un objet donné dans une image, un contour occultant correspond à tout fragment de la frontière de l'objet où celui-ci occulte le reste de la scène, et fait donc partie de la silhouette de l'objet. Les contours occultants sont essentiels pour contraindre le recalage d'un modèle biomécanique déformable, comme celui de l'utérus [25, 23] et du foie [91, 62, 5]. Ces systèmes sont bien avancés en termes de calcul de recalage. Toutefois, ils exigent que le chirurgien marque manuellement les contours occultants sur les images de cœlioscopie pendant l'opération. Cela réduit considérablement l'acceptation et l'accessibilité de la cœlioscopie augmentée, car le concept d'occultation d'un contour n'est pas trivial et leur marquage demande au chirurgien d'y consacrer un temps précieux pendant l'opération. Nous proposons de détecter les contours occultants de l'organe automatiquement afin d'automatiser les systèmes de réalité augmentée existants. Nous abordons le problème général, défini sous le terme *détection de contours occultants de classe d'objet* (en anglais *Object-Class Occluding Contour Detection*, OC2D) et nous spécialisons notre détecteur au cas l'utérus. Ce problème est à l'intersection entre la détection de contours sémantiques et la détection de contours d'occultation. La détection de contours sémantiques consiste à repérer les contours spécifiques d'une classe d'objet et peut être considérée comme le

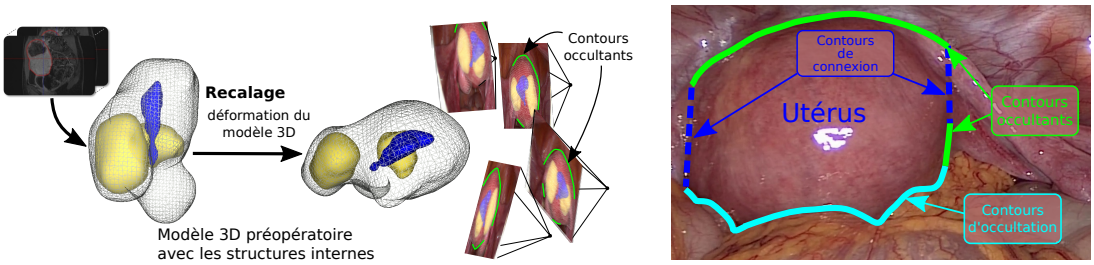


FIGURE 3.1 – À gauche : en cœlioscopie augmentée, le modèle 3D préopératoire est recalé en ajustant les contours occultants de l'organe sur les images cœlioscopiques. Les systèmes actuels exigent que le chirurgien marque ces contours manuellement pendant l'opération. À droite : un contour occultant correspond à une limite de l'organe où l'organe occulte une autre structure, par opposition à un contour d'occultation où l'organe est occulté par une autre structure. Il est possible que la frontière d'un organe ne corresponde ni à un contour occultant, ni à un contour d'occultation. Dans ce cas, il s'agit de contours dits de connexions qui correspondent aux transitions anatomiques entre l'utérus et ses jonctions avec les trompes de Fallope. L'ensemble de ces contours forme la silhouette. OC2D est la tâche de détection des contours d'occultation pour un objet spécifique, ici l'utérus. Il s'agit d'une tâche de détection de contours sémantiques.

problème dual de la segmentation sémantique. Les premières méthodes sont basées sur la détection d'arêtes. Les arêtes correspondent à des changements de luminance abrupts. Cependant, les frontières d'un objet ne sont pas toujours caractérisées par des arêtes, en particulier lorsque l'objet et le reste de l'image sont de la même couleur. Les méthodes récentes basées sur des CNNs combinent des caractéristiques haut-niveau avec la forme apprise et les a priori (*priors*) d'une classe d'objet. La détection des contours d'occultation repère l'ensemble des frontières des objets de la scène capturée et les classe en fonction de la relation d'occultation. Cette classification rend la tâche plus difficile que la détection de contours sémantiques. Des méthodes basées sur des CNNs ont également montré de bons résultats pour un grand nombre de classes d'objets dans des images naturelles. La tâche de détection de contours occultants spécifiques à une classe d'objet combine les difficultés d'une classe d'objet et de la relation d'occultation. Son application à l'utérus en cœlioscopie est encore plus difficile car les couleurs sont peu discriminantes dans cet environnement d'acquisition. À notre connaissance, il n'existe pas dans la littérature de solution spécifique au problème de la détection de contours occultants de classes d'objets.

Dans ce qui suit, nous nommerons vrai contour, tout élément labellisé comme étant le contour à détecter (vérité terrain) et réponse, tout élément prédit comme contour par le détecteur utilisé.

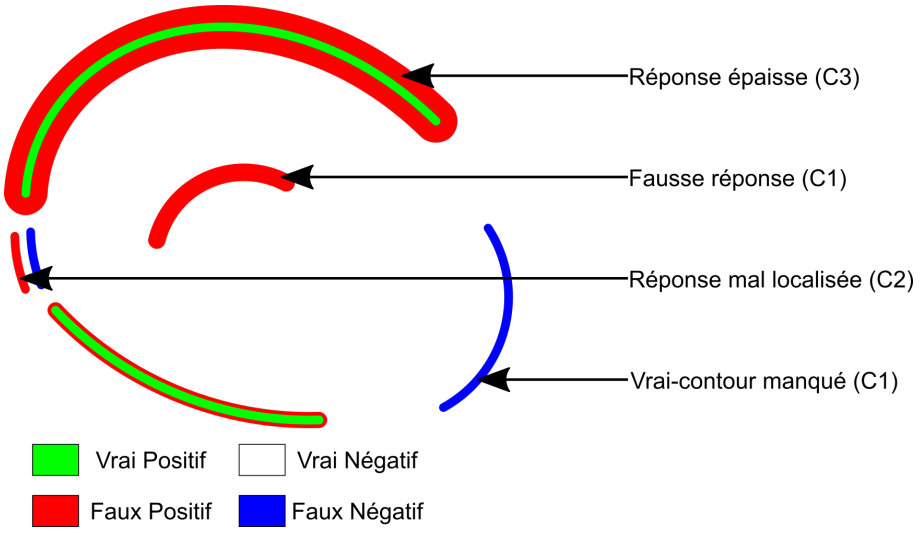


FIGURE 3.2 – Illustration de la classification des contours. Les différents types d'erreurs sont explicités avec le critère de Canny associé.

3.2 État de l'art

3.2.1 Évaluation des contours

3.2.1.1 Motivations

La détection de contours et en particulier la détection de contours occultants de classes d'objets ne possède pas de score d'évaluation conforme aux critères de performance définis par Canny [18].

3.2.1.2 Méthodes existantes

Plusieurs techniques d'évaluation de détecteur de contours existent dans la littérature. John Canny, qui a proposé un détecteur d'arêtes basé sur une hystérésis entre 2 seuils du gradient, a également proposé 3 critères principaux pour évaluer la qualité d'un détecteur d'arêtes [18]. Ces critères se transposent parfaitement à la détection de contours. Les trois critères définis par Canny sont les suivants :

C1 les contours réels ne doivent pas être manqués et les réponses ne doivent pas être fausses ;

C2 les réponses doivent être proches des contours réels ;

C3 chaque contour réel ne doit produire qu'une seule réponse.

On appelle fausse réponse, la prédiction d'un élément qui n'est ni un vrai contour, ni proche d'un vrai contour. On dira que la réponse est mal localisée quand la prédiction d'un élément est proche d'un vrai contour mais pas parfaitement alignée. On parlera de réponse épaisse lorsque la prédiction propose un grand nombre de réponses autour d'un vrai contour. Nous utiliserons ces critères pour évaluer les méthodes de détection de contours. En fonction des applications, chacun des critères proposés est plus ou moins critique. Dans notre application par exemple, l'impact des fausses réponses reste limité par l'utilisation d'un M-estimateur dans l'algorithme de recalage. En revanche, il est impossible de pallier l'absence de détection de vrais contours

puisqu'il s'agit d'une perte d'information nécessaire à la résolution du problème du recalage.

Une première méthode pour évaluer un détecteur de contours est d'utiliser la classification Vrai/Faux Positif/Négatif de la prédiction par rapport à la vérité terrain. On évalue chaque pixel en comparant la classe proposée par le système de classification et la classe réelle. Dans notre contexte, un positif correspond à un pixel contour et un négatif à un pixel non-contour. Par exemple, un faux positif correspond alors à un contour prédit mais qui en réalité n'en est pas un. Un système de classification idéal ne propose aucun Faux Positif (FP) ni aucun Faux Négatif (FN) mais seulement des Vrais Positifs (VP) et Vrai Négatifs (VN). Nous appellerons dans ce qui suit *mesures d'erreurs statistiques* l'ensemble des scores reposant sur ces éléments conformément à la terminologie utilisée dans la littérature. Ces mesures sont très utilisées pour évaluer des solutions de classification.

Ainsi, on peut définir la précision (p) comme le rapport entre le nombre de VP et l'ensemble des positifs détectés (VP + FP). Le rappel (r) est le rapport entre le nombre de VP et le nombre de positifs de la vérité terrain (VP + FN). Un haut rappel signifie que la plupart des éléments positifs à détecter ont été détectés. Ces mesures peuvent être combinées pour obtenir des compromis entre la précision et le rappel comme par exemple via le coefficient Dice, aussi appelé score F1, ou *Intersection-over-Union* (IoU). Elles s'écrivent sous la forme :

$$p = \frac{|VP|}{|FP| + |VP|}; \qquad r = \frac{|VP|}{|FN| + |VP|}; \tag{3.1}$$

$$f_1 = \frac{2 \times |VP|}{2 \times |VP| + |FP| + |FN|}; \qquad IoU = \frac{|VP|}{|FP| + |FN| + |VP|}, \tag{3.2}$$

avec $|VP|, |FP|, |FN|$ le nombre de vrai positifs, de faux positifs et de faux négatifs respectivement. On remarque que le nombre de vrais négatifs n'est pas impliqué dans ces différentes métriques. C'est un avantage dans le cadre de la détection des contours où le nombre de négatifs (pixels non-contour) est bien plus important que le nombre de positifs.

Cette approche est très largement utilisée en segmentation sémantique et pour la détection d'objets. La limite de ces mesures est qu'elles ne donnent pas d'information sur la distance entre la prédiction des contours par rapport à la vérité terrain. Dès lors que la prédiction est décalée par rapport à la vérité terrain, cette prédiction est considérée comme fausse, quelle que soit la valeur du décalage. On ne peut ainsi pas distinguer une réponse trop épaisse d'une fausse réponse (voir figure 3.2).

Pour répondre à cette limite, des mesures faisant intervenir une zone de tolérance ont été proposées [85]. La zone de tolérance permet de considérer les réponses légèrement mal localisées comme des vrais positifs. À partir de ce sous-ensemble entre les prédictions et la vérité terrain, une mise en correspondance est calculée grâce à une minimisation de flux pour un graphe biparti. En pratique, seules des solutions approchées peuvent être trouvées. Néanmoins, cette solution ne vérifie pas le critère C3. Cette métrique sera évaluée et comparée à celle proposée dans ce travail. Comme il s'agit d'une mesure de similarité et que les autres sont des mesures d'erreur, nous utiliserons le complémentaire de cette métrique noté $1 - MF$ (MF est définie dans le paragraphe suivant).

Un autre score permettant d'évaluer les performances d'un détecteur de contour est la courbe précision/rappel. Cette courbe décrit l'évolution de la précision en

fonction du rappel. Dans notre exemple, les détecteurs de contours proposent souvent une carte de probabilité qu'il faut alors binariser pour avoir une prédiction. On peut alors faire varier le seuil de binarisation pour tracer cette courbe. Cette courbe permet de déterminer le seuil qui donne le bon compromis entre précision et rappel. On associe souvent cette courbe aux deux métriques AP et MF, pour *Average Precision* et *Maximum F* (score). La première est obtenue en faisant la moyenne des précisions obtenues pour l'ensemble des seuils testés, la seconde correspond au score f_1 maximum.

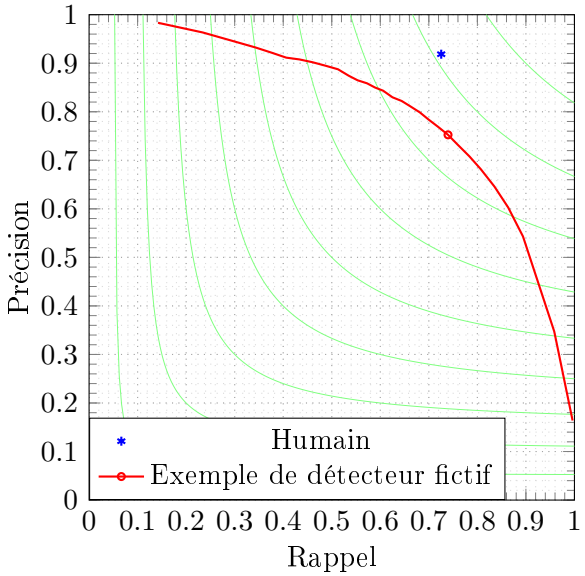


FIGURE 3.3 – Exemple de courbe Précision/Rappel. Pour le détecteur fictif (rouge), on peut tracer la courbe en faisant varier le seuil de décision. Le point de cette courbe correspond au MF. Généralement, la comparaison avec l'annotation d'opérateur humain est proposée à titre de comparaison. Dans cet exemple, $AP = 0,78$ et $MF = 0,75$. Les courbes vertes correspondent à des lignes où le score F1 est constant. Plus un point est proche du coin en haut à droite (1 ;1), meilleur est la prédiction.

Dans nos expériences, nous n'utilisons pas directement de seuil pour obtenir la prédictions des contours. La prédiction est obtenue de manière classique dans un problème de classification en récupérant la classe qui obtient la probabilité la plus grande. Dans plusieurs travaux qui utilisent beaucoup de classes, les cartes de probabilités de chaque classe sont traitées indépendamment. Ces approches utilisent donc un seuil de binarisation et ont par conséquent tendance à présenter leurs résultats via des courbes précision/rappel.

En opposition aux mesures d'erreurs statistiques, des scores basés sur les distances ont été proposés comme par exemple la distance symétrique. Ces méthodes évaluent la distribution des distances entre les éléments de la prédiction et les éléments de la vérité terrain. Prenons l'exemple de la distance symétrique SD_1 [33]. Cette mesure d'erreur fait la moyenne des distances de chaque pixel de la prédiction vers le pixel de la vérité terrain le plus proche d'une part et d'autre part la moyenne des distances de chaque pixel de la vérité terrain vers le pixel de la prédiction le plus proche. Le problème des mesures basées sur les distances est que le moindre faux positif loin du vrai contour va considérablement augmenter le score ce qui rend difficile leur usage en pratique. L'utilisation d'une fonction de distance tronquée,

c'est-à-dire $d_M(a,b) = \min(d(a,b), M)$ avec $M \in \mathbb{R}^+$ et d la distance euclidienne, apporte un intérêt non négligeable. En effet, cela permet de limiter l'impact d'un pixel faux positif solitaire très loin de la vérité terrain et donne également une borne supérieur théorique à la mesure considérée.

Les travaux de Magnier et al., Lopez-Molina et al. [80, 75] ont proposé une comparaison de ces scores. Pour comparer les différents scores, Lopez-Molina et al. ont généré des réponses à partir d'altérations artificielles sur la vérité terrain [75]. L'idée est d'observer l'évolution de la valeur de chaque score avec l'application successive d'altérations comme l'ajout de faux négatifs ou faux positifs en partant d'une réponse qui correspond exactement à la vérité terrain. Le comportement idéal d'une métrique est un profil monotone (croissante pour une mesure d'erreur, décroissante pour une mesure de similarité) avec une pente relativement stable en fonction de l'amplitude de l'altération appliquée. On peut reprendre la formule de Liu et al. qui affirme que : *un score de qualité est précis si de petits changements dans la sortie du détecteur sont traduits par de petits changements dans sa valeur* [71]. En fonction de ces observations, choisir la métrique la plus appropriée dans l'absolu n'est pas possible. En effet, aucun score ne propose un comportement idéal pour chaque altération. Un choix de score doit être fait en fonction de l'application de cette mesure et du type d'erreur que l'on veut pénaliser.

3.2.2 Détection de contours

3.2.2.1 Motivations

La détection de contours et d'arêtes est un problème relativement ancien. En effet, dans les modèles théoriques du traitement de la vision dans le système de perception humain, il est affirmé que la détection d'arêtes est la première étape pour établir un croquis de la scène observée. Ce croquis est ensuite utilisé pour définir les zones d'intérêt pour lesquelles un maximum d'informations doit être récolté [78]. La détection d'arêtes a notamment été utilisée pour fournir des indices visuels pour la reconnaissance de formes et le suivi d'objets. Les récentes améliorations apportées par l'usage de techniques d'apprentissage profond et en particulier de l'usage de réseaux convolutifs (CNNs) ont permis de s'intéresser à des sous-problèmes encore plus spécifiques. Tandis que la détection de contours agnostiques traite du problème de détection de contours sans distinction spécifique à une classe d'objet en particulier, la détection de contours sémantiques associe cette détection à une classe, par exemple une classe d'appartenance à un objet. La détection de contours sémantiques, par sa spécialisation à une classe d'objets, offre ainsi de nouvelles possibilités d'application comparé à la détection d'arêtes. Elle est par exemple d'un très grand intérêt pour des applications nécessitant un traitement spécifique en présence d'occultations ou encore lors de la détection de contours d'objets spécifiques Les méthodes de segmentations sémantiques récentes commettent la grande majorité de leurs erreurs sur les régions de l'image situées à la frontière des objets. Pour cette raison, de nombreux travaux ont été portés sur la caractérisation des frontières de classe d'objets afin d'améliorer les résultats de segmentation.

3.2.2.2 Méthodes classiques de détection de contours agnostiques

La détection d'arêtes et de contours est un problème ouvert qui, comme beaucoup de problèmes de vision par ordinateur, a observé un saut de performances grâce aux techniques d'apprentissage profond. Traditionnellement, un détecteur de contour comprend 2 étapes : l'extraction de caractéristiques basées sur le gradient de l'intensité de l'image et la classification binaire permettant de déterminer si un pixel est traversé ou non par un contour. Certains travaux utilisant des filtres basées sur des convolutions tels que Sobel et Prewitt ont permis de détecter les arêtes d'une image en s'intéressant aux zones où la norme du gradient est élevée.

Cependant, une telle approche ne permet pas de distinguer la frontière d'un objet d'un changement brusque d'intensité. En effet, la notion d'objet est difficile à appréhender à une échelle très locale. Des travaux ont été menés pour comprendre ce que l'humain considérait comme un contour pertinent, c'est-à-dire permettant de segmenter l'image en des régions qui avaient du sens. C'est notamment une des pistes de recherche ayant motivée la création des jeux de données BDS300 [84] et BDS500 [8]. Les jeux de données ont été annotés par des personnes à qui on a volontairement donné une consigne vague : "Définir des régions d'importance relativement équivalente pour chaque image. L'idéal serait d'avoir entre 2 et 20 régions.". Le but de cette opération était de vérifier si l'annotation était cohérente d'un annotateur à un autre et ainsi trouver des algorithmes pour extraire ces contours définis par l'humain.

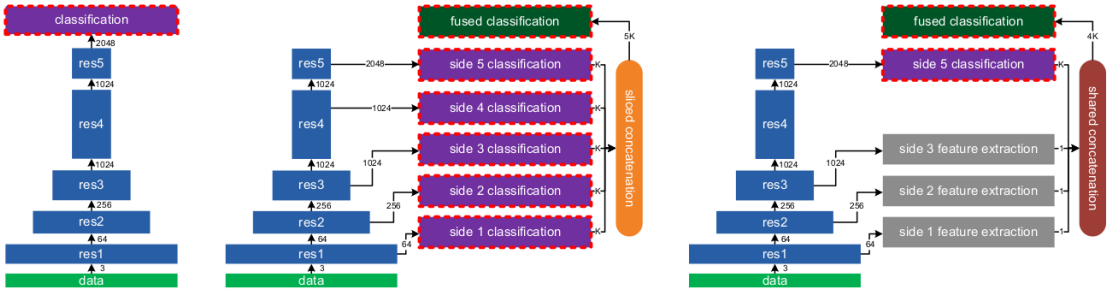
Avant l'arrivée de l'apprentissage profond, des premiers modèles basées sur les données, *data-driven* en anglais, ont permis de premières percées : l'extraction de caractéristiques à partir d'une probabilité à posteriori gPb (*globalized probability of boundary*) [8], l'utilisation de croquis de caractéristiques de contours [69], ou de caractéristiques entraînées à partir de forêt d'arbres décisionnels [32] (*random forests* en anglais).

3.2.2.3 Méthodes avec apprentissage profond pour la détection de contours agnostiques

L'utilisation de CNNs a permis d'exploiter d'une part les informations locales, exploitées par les méthodes classiques, pour détecter ou non ces frontières mais également d'interpréter le contexte de l'image pour distinguer la frontière d'un objet d'un changement brusque d'intensité. Des solutions utilisant des architectures proches de celles utilisées pour la segmentation sémantique ont ainsi été proposées [104, 123, 121, 72, 14, 61, 97] pour la détection de contours. Les principales innovations ont consisté à utiliser différents étages du réseau de neurones pour prédire les contours à différentes échelles et les combiner astucieusement. Pour entraîner ce type de réseaux, le choix de l'entropie croisée pondérée, *weighted cross entropy* en anglais, est très largement utilisée pour compenser la distribution très inégale entre les pixels contours et les pixels non-contours dans une image. On a généralement au moins 90% des pixels de l'image qui ne sont pas des contours ce qui crée un biais très fort qui a tendance à favoriser la sur-prédiction de la classe sur-représentée. Pour compenser ce biais, on ajoute un poids sur les pixels des classes sous-représentées pour augmenter leur impact dans le calcul de la fonction de coût.

3.2.2.4 Détection de contours sémantiques

Les travaux de références qui ont définis le terme de détection de contours sémantiques, *Semantic Edge Detection* en anglais (SED) sont ceux de Hariharan et al. [53]. C'est un problème qui requiert à la fois des informations bas-niveau de l'image pour localiser précisément la position du contour, mais aussi les informations haut-niveau pour classer correctement le type d'objet représenté. Hariharan et al. ont combiné un détecteur de contours (approche bottom-up) avec un détecteur d'objets (approche top-down) pour extraire les contours sémantiques de classes d'objets, qu'ils présentent comme le problème dual de la segmentation sémantique. Ces travaux vont proposer le jeu de données *Semantic Boundary Dataset* (SBD) [53].



(a) Basic Network (b) Deeply Supervised Network (DSN) (c) CASENet

FIGURE 3.4 – Trois architectures CNN permettant de décrire la distinction entre une architecture ResNet classique (a), son approche profondément supervisé (b) et l'architecture de CASENet [125] (c) (Illustration extraite de [125]).

Plusieurs méthodes basées sur de l'apprentissage profond ont depuis été proposées pour répondre à la tâche de la détection de contours sémantiques. La méthode de référence est encore aujourd'hui CASENet [125]. Cette méthode reprend la forme du réseau proposé par HED [121]. L'architecture de CASENet reprend la structure de ResNet [56] et ajoute une forme de supervision profonde, *deep supervision* ou *deeply supervised* en anglais. Cet ajout, proposé par Lee et al. [66], propose de superviser plusieurs couches du réseau à la fois en plus de la prédiction finale. Dans CASENet, l'extraction de caractéristiques des couches 1, 2, 3 sont concaténées avec les résultats de la classification de la dernière couche du réseau pour mettre en place cette supervision profonde (voir la figure 3.4). Pour entraîner le réseau, Yu et al. [125] utilisent une entropie croisée binaire pondérée, appelée fonction de coût multi-label et définie par :

$$\begin{aligned}\mathcal{L}(\mathbf{W}) &= \sum_k \mathcal{L}_k(\mathbf{W}) \\ &= \sum_k \sum_{\mathbf{p}} \{-\beta \hat{\mathbf{Y}}_k(\mathbf{p}) \log \mathbf{Y}_k(\mathbf{p}|\mathbf{I};\mathbf{W}) \qquad (3.3) \\ &\quad - (1-\beta)(1-\hat{\mathbf{Y}}_k(\mathbf{p})) \log(1-\mathbf{Y}_k(\mathbf{p}|\mathbf{I};\mathbf{W}))\}, \qquad (3.4)\end{aligned}$$

avec β le poids défini par le pourcentage de pixels non-contours dans l'image, $\mathbf{I}$ l'image d'entrée, $\hat{\mathbf{Y}}_k$ et $\mathbf{Y}_k$ les cartes de probabilités de contours et les cartes de contours pour la classe k.

Les méthodes concurrentes ont proposé d'autres formes de supervision profondes [73], en particulier en évaluant la capacité du réseau à prédire à la fois des contours agnostiques et les contours sémantiques.

Ces avancées dans la détection de contours ont été intégrées pour d'autre tâches comme la fermeture des contours [59] pour obtenir des régions fermées, la prédiction des contours sur des images de haute résolution [130], ou directement la segmentation sémantique [127, 111, 19] ou segmentation d'instances [60].

3.2.2.5 Jeux de données

Les jeux de données utilisés pour la détection de contours sont :

— **Berkley Segmentation Dataset** [1] (BDS500) [8] Ce jeu contient 500 images naturelles annotées manuellement par en moyenne 5 personnes différentes. C'est une extension du jeu de données BDS300 [84]. La consigne donnée aux annotateurs était intentionnellement vague avec la seule contrainte que chaque région isolée devait avoir une importance égale aux autres et de distinguer entre 2 à 20 régions. L'idée était de vérifier si la perception de la segmentation d'une scène était consistante d'un annotateur à l'autre. Ce jeu de données est utilisé en particulier dans [104, 14, 123, 61, 121, 30, 97].

— **Semantic Boundary Dataset** [2](SBD) [53] Ce jeu contient les annotations de contours pour 11355 images extraites du jeu de données PASCAL VOC 2011 [34] pour 20 catégories d'objets de la vie courante (avion, vélo, oiseau, personne, voiture, etc.). Ce jeu de données est utilisé en particulier dans les travaux de [53, 4, 126, 125]

— **NYU-Depth v2** [3] (NYUD) [105] Ce jeu de données est composée de séquences vidéos de scènes intérieures capturées par une caméra Kinect. Ce jeu contient 1449 images contenant à la fois les données brutes (images RVB et cartes de profondeur) ainsi que les segmentations des instances des objets présents. Ce jeu de données est utilisé en particulier dans [104, 121, 30, 97]

— **Cityscapes** [4] [27] Ce jeu contient des segmentations de scènes de ville en Allemagne extraites d'une caméra embarquée. Ce jeu contient 5000 images finement annotées et 20 000 annotées plus grossièrement. Chaque pixel de l'image annotée finement est associée à une des 30 classes décrivant l'environnement urbain (sol, humain, véhicules, ...). Ce jeu de données est utilisé en particulier dans [4, 126, 125].

— **Barcelona Images for Perceptual Edges Dataset** [5] (BIPED) [97] Ce jeu contient 250 images d'extérieur en 720p annotées pour la détection d'arêtes.

— **PASCAL-Context Dataset** [6] [89] Ce jeu de données a été obtenue à partir des images du jeu PASCAL VOC 2010, qui ont été annotées de manière exhaustive avec un nombre de classes non définies à priori. Les annotateurs avaient carte blanche pour définir les classes pour annoter l'ensemble de l'image. Après regroupement des redondances, 540 classes différentes ont été utilisées pour annoter 10 103 images pour l'entraînement et la validation, et 9637 images de tests. Ce jeu de données est utilisé en particulier dans [97, 61].

1. BDS500 : www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/resources.html
2. SBD : http://home.bharathh.info/pubs/codes/SBD/download.html
3. NYUDv2 : https://cs.nyu.edu/~silberman/datasets/nyu_depth_v2
4. Cityscapes : https://www.cityscapes-dataset.com/
5. BIPED : https://xavysp.github.io/MBIPED/
6. PASCAL-Context : https://cs.stanford.edu/~roozbeh/pascal-context/

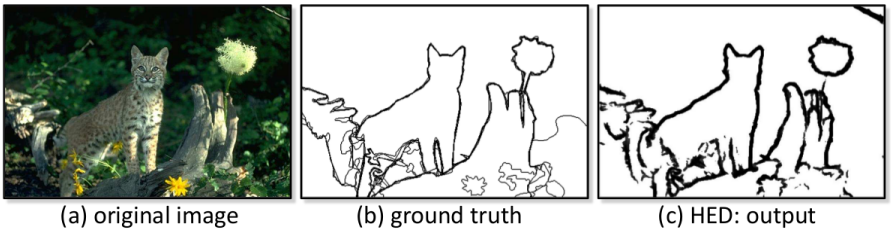


FIGURE 3.5 – Illustration de l'épaisseur des contours à la sortie d'un réseau convolutif (ici HED [121], illustration extraite de l'article original). On constate que les contours prédits sont d'une épaisseur bien plus grande par rapport à l'annotation de la vérité terrain.

3.2.2.6 Limites

Les méthodes de détection de contours avec CNNs souffrent d'un problème récurrent : les contours obtenus sont particulièrement épais. Plusieurs travaux plus récents se sont donc concentrés sur l'amincissement des contours obtenus [4, 126, 30, 97]. Certains sont partis du constat que les annotations du jeu d'entraînement utilisées n'étaient pas suffisamment précises. En effet, la plupart des outils d'annotation proposent d'annoter une image en plaçant manuellement les sommets d'une ligne brisée ou d'un polygone. Cette approche rend l'annotation plus pratique mais fait nécessairement apparaître un décalage de quelques pixels entre l'annotation et la position réelle du contour. Ce décalage répété sur l'ensemble du jeu de données rend la détection plus ambiguë (voir figure 3.6). Durant l'entraînement, il devient effectivement difficile pour le réseau d'interpréter la position exacte des contours, ce qui se traduit par une prédiction plus épaisse. Plusieurs méthodes ont ainsi proposées de raffiner les annotations sur les arêtes détectées au cours de l'entraînement [4, 126]. Il faut également préciser que dans le cadre d'images médicales, la frontière entre l'organe et d'autres structures internes est parfois floue. Les conditions de prise de vue, d'éclairage peuvent également rendre cette frontière encore plus confuse qu'elle ne l'est déjà.

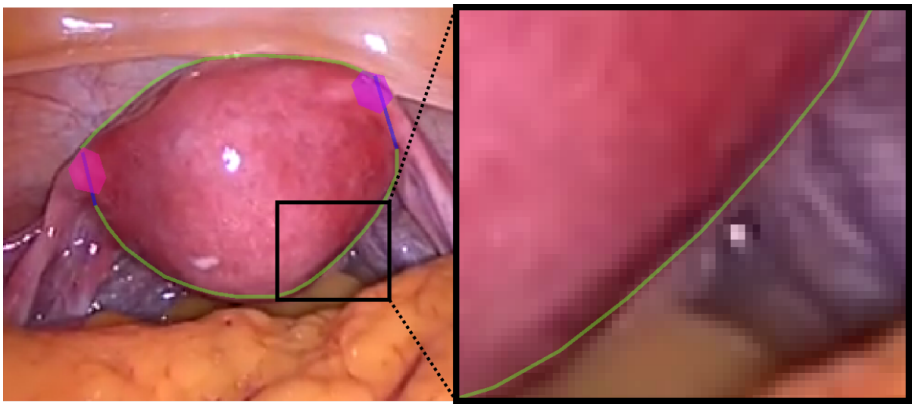


FIGURE 3.6 – Exemple d'une image extraite de notre jeu de données. Sur la zone zoomée (droite), on peut constater un léger décalage entre l'annotation (vert) et le contour réel.

3.2.3 Applications médicales

3.2.3.1 Motivations

Nos travaux sur la détection de contours occultants de l'utérus sont motivés par la nécessité d'annoter les contours occultants de l'utérus lors de l'utilisation du logiciel de réalité augmentée dédiée à la myomectomie.

L'algorithme de recalage utilisé par le logiciel a déjà connu de nombreuses améliorations [24, 25, 23] et plus récemment [26]. La chaîne de traitement globale est décrite dans le chapitre 2.4. Les principaux changements se sont dans un premier temps concentrés sur l'amélioration de la robustesse et l'utilisabilité du logiciel. Une des limites pour l'acceptation et l'utilisabilité de cet outil est de nécessiter un certain nombre d'étapes préliminaires bien souvent manuelles pour obtenir le suivi de l'organe. Parmi ces étapes, l'annotation des contours occultants de l'utérus est particulièrement critique pour le bon déroulement de l'étape du recalage. La précision requise demande une certaine expertise même pour un chirurgien en gynécologie et un certain temps pour réaliser l'annotation. Un très grand nombre d'images annotées permet de bien contraindre la solution du recalage, à savoir selon un grand nombre de points de vue. En pratique, le nombre d'images annotées est compris entre 15 et 20 de sorte que le temps d'annotation soit acceptable. Les prochaines étapes d'évolution sont donc de chercher à automatiser un certain nombre de tâches comme l'annotation de ces contours occultants [39], et la segmentation du modèle 3D peropératoire à partir de l'image IRM [26].

L'application de notre tâche de détection de contours occultants aux images cœlioscopiques de l'utérus est une tâche très spécifique. Il existe encore aujourd'hui peu de solutions fonctionnelles dans le domaine médical. Ceci peut en partie être justifié par la difficulté à disposer de jeux de données d'images sur le sujet désiré. La collecte d'images et leur annotation pour l'établissement d'un jeu de données est particulièrement problématique dans ce domaine car les données de santé sont, d'une part, soumises au secret médical et, d'autre part, souvent difficiles à annoter par des utilisateurs non-experts.

3.2.3.2 Méthodes existantes

Créer son propre jeu de données est bien souvent un prérequis à l'utilisation de techniques mettant en jeu de l'apprentissage profond. Tandis que beaucoup de travaux de recherches dans le domaine médical se concentrent sur la constitution de jeux de données pour la segmentation automatique d'images 3D telles que des scanners TDM ou IRM, encore peu se consacrent à la segmentation automatique d'images cœlioscopiques. Il existe quelques jeux de données annotées sur des images 2D destinées à la détection d'actions chirurgicales [67], détection de phases [113, 108], et de structures anatomiques [67]. Ces jeux de données sont spécifiques à un type d'opération, en l'occurrence pour les cholécystectomies [113, 67, 108] et la résection de fibromes [112, 67].

Une approche similaire avec des motivations communes a été proposée en parallèle de nos travaux pour la détection des contours occultants du rein [55]. Ces travaux ont entraîné un réseau inspiré du réseau U-Net [101] sur un jeu de données dédié. Leur jeu de données [7] est constitué de 15 séquences de 100 images stéréo ob-

7. https://endovissub2017-kidneyboundarydetection.grand-challenge.org/

tenues à partir d'un robot Da Vinci pour des expériences de néphrectomies sur des porcs. Hattab et al. [55] ont également proposé plusieurs scores d'évaluation de leur détecteur de contours.

3.3 Jeu de données pour la cœlioscopie de l'utérus

Nous proposons le premier jeu de données sur des images cœlioscopiques d'utérus avec des annotations de contours pour un ensemble de 3818 images. Plus de détails sur la construction du jeu de données sont fournis en annexe A.

3.3.1 Images

Les images sont extraites de 79 vidéos d'opérations de l'utérus, 29 obtenues via une étude validée par le CPP et 50 depuis des vidéos extraites de Youtube. Ces vidéos présentent un large panel d'opérations dont des hystérectomies, résections d'endométrioses, de nodules et de kystes. Pour chaque vidéo, nous avons extrait de multiples images de manière à s'assurer que notre jeu de données contient les deux sources de variabilités requises. La première est la variabilité intra-patiente qui est due au changement de point de vue, la déformation de l'utérus et les changements de couleurs au cours de l'opération. La seconde est la variabilité inter-patiente qui peut être due aux différentes formes et/ou apparence de l'utérus, ainsi que les changements spécifiques au type d'opération et de maladie rencontrée. Nous nous sommes également assuré d'inclure certaines situations typiques d'une opération réelle telles que l'occultation de la scène par des instruments de chirurgie et des images floues. Finalement, nous avons préféré une sélection d'images semi-automatique où les images redondantes sont filtrées manuellement plutôt qu'une approche complètement automatique qui extrait des images à une fréquence donnée. En effet, l'ajout d'images très similaires n'apporte pas d'intérêt pour l'entraînement d'un réseau de neurones et peut même au contraire créer des biais d'apprentissage sur les situations surreprésentées. D'un point de vue pratique, cela limite le nombre total d'annotations, qui sont réalisées manuellement par un expert.

3.3.2 Annotations et classes

Comme illustré sur les figures 3.1 et 3.7, les classes annotées sont les contours occultants, les contours d'occultation et les contours dits de connexion de l'utérus. Les contours de connexion se présentent sur la jonction entre l'utérus et les trompes de Fallope, et avec le col, lorsque l'utérus s'arrête anatomiquement mais il n'y a pas à proprement parler de contour occultant ou d'occultation.

Les contours de connexion ne sont pas utilisés dans notre méthode OC2D mais peuvent être utilisés en combinaison avec les autres classes de contours pour former une région fermée pour une tâche de segmentation (voir l'annexe A). L'annotation a été réalisée par une interne en médecine en utilisant la plateforme d'annotation en ligne Supervisely [3].

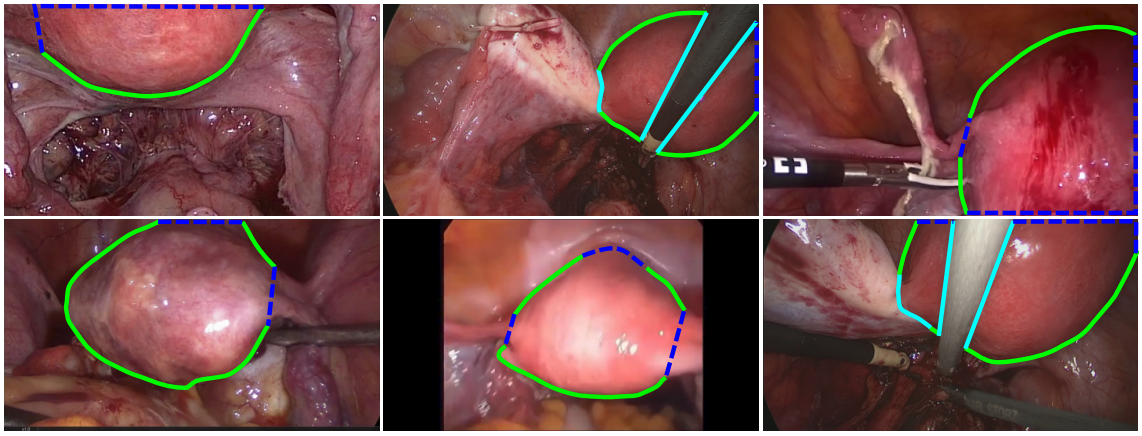


FIGURE 3.7 – Haut : Extraits de notre jeu de données de 3818 images de cœlioscopie annotées. L'utérus est le principal organe d'intérêt. Les contours occultants sont en vert, les contours d'occultation en cyan et les contours de connexion en bleu.

3.4 Score d'évaluation pour la détection de contours

3.4.1 Motivation

Comme détaillé dans l'état de l'art, il nous a paru intéressant de proposer un score qui vise à respecter les critères de Canny dans leur ensemble. En plus des critères de Canny, nous avons défini 2 nouveaux critères :

— **C4** le score doit être invariant à la résolution de l'image
— **C5** le score doit être invariant à la quantité de vrais contours.

Le but de ces nouveaux critères est de garantir que le score est invariant à la déformation de l'objet considéré, aux paramètres intrinsèques de la caméra et à la quantité d'occultation. Ce sont des conditions particulièrement importantes pour notre application car on souhaite que les contours occultants contraignent de manière équivalente le recalage. Ces critères sont comparables entre deux jeux de données qui ne partagent pas les mêmes dimensions d'images ou la même prévalence des contours.

3.4.2 Définition d'un nouveau score d'évaluation des contours

Soit $\mathcal{I}$ l'ensemble des coordonnées des pixels de l'image, $C \subset \mathcal{I}$ les vrais-contours et $R \subset \mathcal{I}$ les réponses proposés par un détecteur de contours. Nous utilisons une distance de tolérance $d_{\max}$ telle que un *contour manqué* est défini comme un vrai-contour sans réponse localisée à une distance inférieure à $d_{\max}$; et une *réponse parasite/fallacieuse* comme une réponse localisée à une distance plus importante que $d_{\max}$ du plus proche pixel vrai-contour. En pratique, la valeur de $d_{\max}$ est choisie pour valoir 2% de la diagonale de l'image [30]. Un *contour manqué* et une *réponse parasite* sont respectivement notés FN et FP par la suite. Les réponses situés dans la zone de tolérance $\mathcal{T} = \{r \in \mathcal{I} \mid d(r, C) < d_{\max}\}$ sont notées VP (Vrai-Positif) et les réponses en dehors de $\mathcal{T}$ sont notées FP. Ces définitions ainsi que les différents termes du score S sont illustrés sur la figure 3.8.

Le score proposé $S(R, C)$ combine l'utilisation de $d_{\max}$ pour les FP et les FN, et la distance entre les vrais-contours et les réponses. Il se compose de 3 termes sous

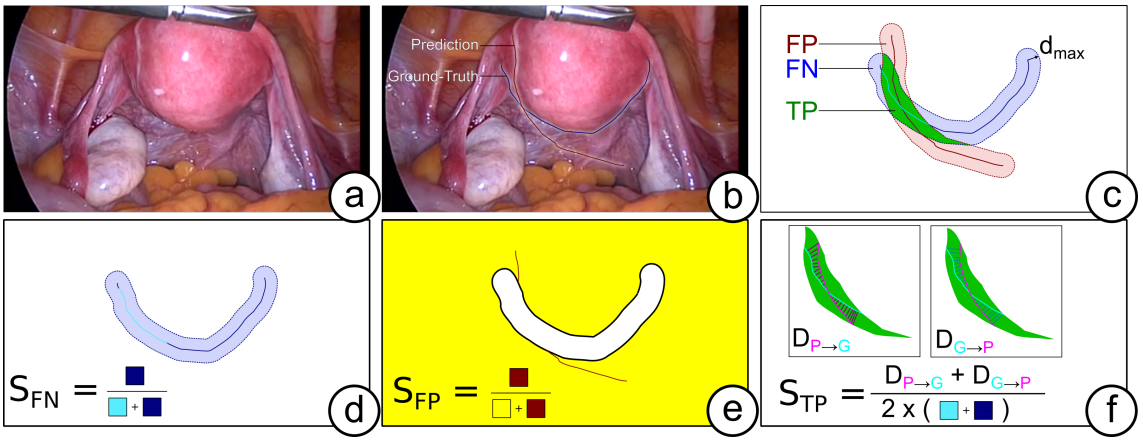


FIGURE 3.8 – Illustration de la formulation du score de contour S. Un détecteur de contour fictif propose une prédiction (b) à partir de l'image d'entrée (a). Les différentes zones définies par la distance seuil $d_{\max}$ sont illustrées en (c). La formule permettant de calculer l'erreur dans chaque zone est illustrée séparément dans (d,e,f).

la forme suivante :

$$S_{VP} = \frac{1}{2}\left(\frac{1}{|C|}\sum_{r\in R\cap\mathcal{T}} d(r, C\backslash FN) + \frac{1}{|C|}\sum_{c\in C\backslash FN} d(c, R\cap\mathcal{T})\right), \tag{3.5}$$

$$S_{FP} = \frac{d_{\max}}{|I| - 2|C|d_{\max}}|FP| \quad \text{et} \tag{3.6}$$

$$S_{FN} = \frac{d_{\max}}{|C|}|FN|. \tag{3.7}$$

Le score combiné $S(R, C)$ somme les 3 termes et les normalise par $d_{\max}$:

$$\begin{aligned} S(R,C) &= \frac{1}{d_{\max}}\left(S_{VP} + S_{FP} + S_{FN}\right) \\ &= \frac{1}{2|C|d_{\max}}\left(\sum_{r\in R\cap\mathcal{T}} d(r, C\backslash FN) + \sum_{c\in C\backslash FN} d(c, R\cap\mathcal{T})\right) \\ &\quad + \frac{|FP|}{|I| - 2|C|d_{\max}} + \frac{|FN|}{|C|}, \end{aligned} \tag{3.8}$$

où l'on note $d(p, Q)$ la distance euclidienne d'un point p à un ensemble de points Q :

$$d(p, Q) = \min_{q\in Q} \|p - q\|_2, \tag{3.9}$$

3.4.2.1 Conformité avec C1, C2

S_{VP} est une mesure de distance symétrique entre les vrais-contours et les réponses. Cela valide C2, c'est-à-dire des réponses proches des vrais-contours.

S_{FP} et S_{FN} sont respectivement la quantité normalisée de FP et FN, chacun de ces termes pondéré par $d_{\max}$. Chacun de ces termes favorise le critère C1, c'est-à-dire pas de réponse parasite et pas de contour manqué, tout en pénalisant de manière équitable les réponses parasites sans prendre en compte leur distance aux vrais-contours.

3.4.2.2 Conformité avec C3

La difficulté de vérifier le critère C3 vient du terme de distance utilisé dans S_{VP} qui utilise le plus proche contour pour chaque réponse, ce qui associe potentiellement le même vrai-contour pour plusieurs réponses. Ce cas de figure est géré en pénalisant la différence entre le nombre de vrais-contours et le nombre de réponses dans la zone de tolérance, en normalisant par $|C|$, alors que les travaux précédents utilisent $|R \cap \mathcal{T}|$ [33].

3.4.2.3 Conformité avec C4, C5

Un haut taux de FN tend à avoir un impact plus limité qu'un haut taux de FP, ce qui nécessite une pondération appropriée [80]. Nous faisons l'hypothèse que la probabilité d'avoir des réponses parasites est *(i)* uniforme dans la zone de tolérance et *(ii)* similaire à la probabilité d'avoir un vrai-contour manqué. De manière à pénaliser équitablement FP et FN à l'intérieur et à l'extérieur de la zone de tolérance, notre pondération est de normaliser S_{FP} et S_{FN} par leur région spatiale, en particulier $||\mathcal{I}| - 2|C|d_{\max}|$ pixels, considérés comme une bonne approximation du nombre de pixels en dehors de la région de tolérance, pour S_{FP}, et $|C|$ pixels pour S_{FN}.

En résumé, tous les trois termes sont normalisés en fonction du nombre de vrais-contours $|C|$ et le deuxième terme intègre aussi la résolution de l'image ce qui permet de satisfaire les critères C4 et C5.

3.4.3 Évaluation du score avec l'application de perturbations

Pour évaluer comment le score proposé réagit aux différents types d'erreurs de prédiction, nous avons simulé sept types de perturbations, nommées P1-P7, pour altérer artificiellement les vrais-contours et les évaluer en tant que réponses obtenues par un détecteur de contours. Certaines de ces perturbations ont été empruntées des travaux de Lopez-Molina et al. [75]. Les types de perturbations visent à mesurer la conformité du score proposé vis-à-vis des critères C1-C5. Ces perturbations sont progressivement appliquées pour passer de la réponse idéale, qui correspond aux vrais-contours, à une réponse très perturbée.

- **P1 : Ajout de FP, 1 (C1 et C3)** : à chaque niveau de perturbation, on ajoute de manière aléatoire des FP dans la réponse. Cette perturbation permet de tester l'impact de réponses parasites (C1) quand les réponses ajoutées aléatoirement sont en dehors de la zone de tolérance, et des réponses multiples (C3), quand les réponses sont dans la zone de tolérance.
- **P2 : Ajout de FP, 2 (C3)** : On augmente le nombre de FP en appliquant l'opération morphologique de dilatation des vrais-contours. Cette perturbation a pour but de rendre les contours prédits de plus en plus épais. Cette perturbation teste l'impact de réponses multiples autour des vrais-contours

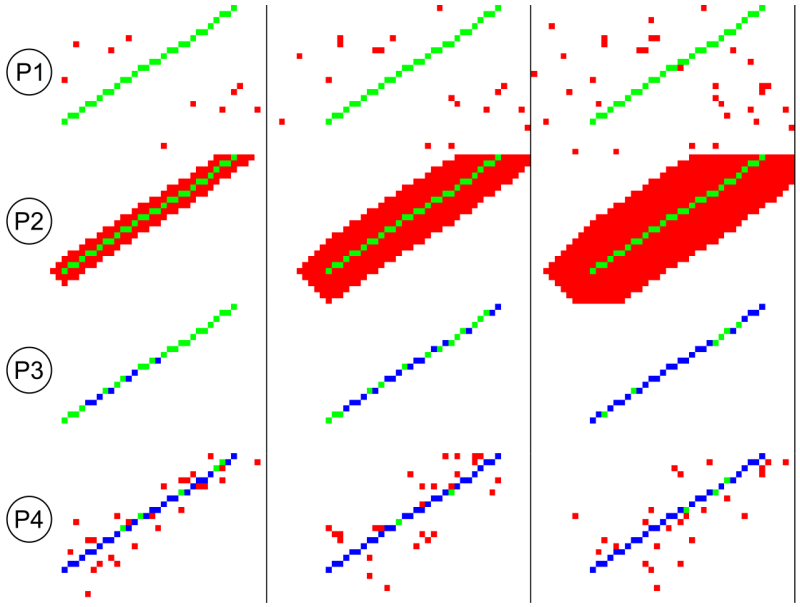


FIGURE 3.9 – Illustration de plusieurs étapes de l'application des perturbations P1-4. Chaque colonne représente un stade d'application de la perturbation : 25% (gauche), 50% (centre), et 75% (droite). L'image est un zoom sur une portion du contour pour une meilleure visibilité. Les couleurs traduisent la classification des pixels : vert pour les VP, rouge pour les FP, bleu pour les FN et blanc pour les VN.

(C3). Si on dilate les contours suffisamment pour sortir de la zone de tolérance, on teste également le critère (C1) dans une moindre mesure.

— **P3 : Ajout de FN (C1)** : Des pixels des vrais-contours sont progressivement retirés de la prédiction, ce qui augmente le nombre de vrais-contours non prédits. Cette perturbation poussée à l'extrême retire tous les vrais-contours de la réponse et propose donc une réponse vide. La valeur du score dans ce cas précis permet de vérifier si l'équilibre choisi pour évaluer de manière équitable les FP et les FN est satisfaisant. Si la valeur de ce score est trop différente de la valeur du score atteint lorsqu'on ajoute une énorme quantité de FP, avec la perturbation P1 par exemple, alors on sait que notre score a tendance à sous-estimer ou surestimer les FP par rapport aux FN ou vice-versa.
— **P4 : localisation (C2)** : Les positions des vrais-contours, utilisés pour réponse, sont aléatoirement et indépendamment perturbées avec une amplitude de plus en plus grande. Cette perturbation vise à simuler la prédiction de réponses proches des vrais-contours mais pas parfaitement alignées. Cette perturbation ne maintient pas la connectivité du contour.
— **P5 : sous-échantillonnage (C4)** : l'image est progressivement sous-échantillonnée en maintenant un taux de FN constant. Pour cette perturbation, l'étape 0 ne propose pas les contours-vrais comme réponse. On utilise comme réponse initiale une portion des vrais contours avec un rappel de 50%. Des FP sont ajoutés sous forme de bruit sur l'ensemble de l'image, à une probabilité de 1% pour chaque pixel. Cette perturbation vise à simuler des changements de paramètres intrinsèques de la caméra pour capturer la scène. La taille de l'image est de plus en plus petite.
— **P6 : Changement d'échelle des contours (C5)** : L'échelle des vrais-

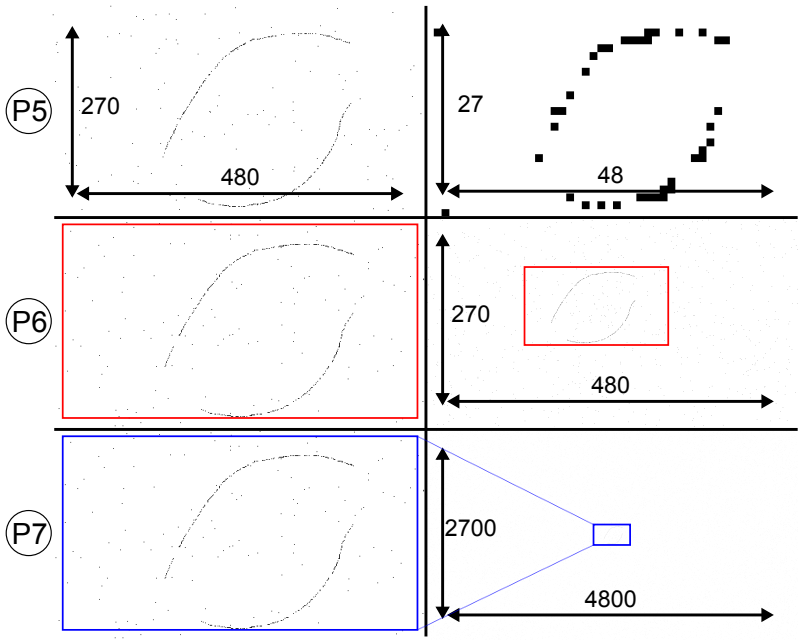


FIGURE 3.10 – Illustration des perturbations P5-7. La première colonne représente l'image originale qui est commune aux trois types de perturbations. La colonne de droite montre la perturbation correspondante au stade maximal de notre étude. Les dimensions des images sont exprimées en pixels.

contours et de la réponse sont progressivement réduits en maintenant un rappel constant. Tout comme P5, on utilise comme réponse initiale une réponse avec un nombre de FN et FP non nul. Leur rapport est maintenu au cours de la perturbation. La taille de l'image est constante mais le nombre de pixels contours diminue. Cette perturbation vise à simuler un changement d'échelle de l'objet suivi dans l'image, lorsqu'un objet suivi s'éloigne de la caméra par exemple.

— **P7 : Changement d'échelle de l'image (C5)** : La taille de l'image est augmentée progressivement en gardant des contours identiques dans l'image. Tout comme P5 et P6, on utilise comme réponse initiale une réponse avec un nombre de FN et FP non nul. Leur rapport est maintenu au cours de la perturbation. Cette expérience est complémentaire de P6. La taille de l'image augmente (multipliée par 10 entre l'état initial et la dernière étape), le nombre de pixels contours est constant. Cette perturbation permet de simuler l'effet du rognage d'une image pour se concentrer sur une zone d'intérêt.

Un score qui respecte les critères C1-C5, doit augmenter au cours des perturbations P1-P4 et rester le plus constant possible pour les perturbations P5-P7. Les perturbations sont illustrées sur les figures 3.9 et 3.10.

En plus du score proposé S, nous avons évalué plusieurs scores utilisés dans la littérature qui sont SD_1 [53], RDE_1 [53] et $1 - MF$ [33, 85]. Les formules de SD_1 et RDE_1 sont données par :

$$SD_1(R,C) = \frac{1}{R \cup C}\left(\sum_{r\in R} d(r,C) + \sum_{c\in C} d(c,R)\right) \tag{3.10}$$

$$RDE_1(R,C) = \frac{1}{|R|}\sum_{r\in R} d(r,C) + \frac{1}{|C|}\sum_{c\in C} d(c,R), \tag{3.11}$$

$$\tag{3.12}$$

Les résultats de l'évaluation sont présentés sur la figure 3.11. Pour rappel, seul le score $1-MF$ est borné, dans l'intervalle $[0,1]$ alors que les trois autres scores ont une borne théorique qui dépend de la taille de l'image, qui contraint la distance maximale observable. Le score S possède une borne théorique qui ne dépend pas vraiment des dimensions de l'image grâce à la distance tronquée. Cette valeur maximale dépend du nombre de vrais-contours. En pratique, le score S explose surtout lorsque la réponse est très épaisse autour des vrais-contours.

- **P1 et P2** : On constate que tous les scores réagissent comme voulu aux perturbations P1 et P2. On peut vérifier que les métriques basées sur les distances (SD_1, RDE_1) sont très sensibles au FP via P1, plus que via P2. Au contraire, on constate que le score S tend à limiter fortement l'impact des FP loin des contours-vrais grâce à l'impact de $d_{\max}$. L'impact des contours épais est extrême. Les deux types de perturbations sur les FP ont un impact quasi-identique pour le score $1-MF$.
- **P3** : Les métriques basées sur les distances et S sont assez peu sensibles au FN. Ils ont un impact conséquent une fois que le nombre de VP diminue vraiment (au-delà de 75% de la perturbation). Le score $1-MF$ évalue graduellement et reste dans les mêmes ordre de grandeur que pour les FP.
- **P4** : On constate une augmentation sur les métriques RDE_1, SD_1 et S. L'impact du bruit de localisation est négligeable sur $1-MF$. C'est la variation de valeur la plus basse pour ce score. On peut ainsi dire que $1-MF$ ne permet pas de distinguer des erreurs sur le bruit de localisation. L'impact de P4 sur SD_1 et RDE_1 est très similaire à celui de P2. En effet, la distance des pixels déplacés par P4 est comparable à la distance des contours épais généré par P2. En revanche, la comparaison entre P2 et P4 est très différente pour S. Pour ces deux perturbations, la majorité des pixels pénalisés sont dans la zone $d_{\max}$ mais dans P4, le nombre de pixels n'augmentent pas. Le score S reste donc dans des valeurs bien plus basses par rapport à celles obtenues avec P2.
- **P5** : Le sous-échantillonnage de l'image diminue le nombre de pixels de l'image. Les distances étant évaluées en pixel, on constate que cela fait diminuer artificiellement les mesures SD_1 et RDE_1. Ce rapprochement des pixels est également observé sur $1-MF$ dans une moindre mesure. L'impact sur le score S est très limité et S tend même à augmenter de manière très faible pour les étapes où la perturbation est la plus forte (de l'ordre de 10^{-3}). Cela peut s'expliquer par la valeur de $d_{\max}$ qui est choisie en fonction de la diagonale de l'image.
- **P6-P7** : Que ce soit en réduisant l'échelle des contours dans l'image ou en augmentant la taille de l'image, on tend à augmenter la distance entre des fausses réponses et les vrais-contours ce qui fait augmenter les scores SD_1

et RDE_1. Le score $1 - MF$ augmente également surtout lorsqu'on augmente la taille de l'image. Le score S est très peu impacté par ces modifications structurelles de l'image. Pour rappel, la situation de départ est obtenue en ajoutant des FP générés sous forme de bruit sur l'image et 50% des vrais-contours sont retirés. On peut constater que le score S pénalise assez peu les FN. Lorsque $d_{\max}$ augmente (avec l'augmentation de la taille de l'image), il y a de plus en plus de chance d'intégrer des FP dans la zone du score S_{VP}. Lorsqu'un FP entre dans cette zone, il fait augmenter brusquement le score S. La normalisation de S par $d_{\max}$ le fait descendre pour un nombre de réponses constant dans la zone $d_{\max}$.

En résumé, les métriques basées sur les distances sont facilement perturbées par des fausses réponses loin des vrais-contours. $1 - MF$ est un très bon candidat pour les perturbations P1-P4 mais n'est pas constant pour P5-P7. Le score proposé S donne de très bons résultats sur les différentes perturbations. On note toutefois un manque de sensibilité vis-à-vis des FN.

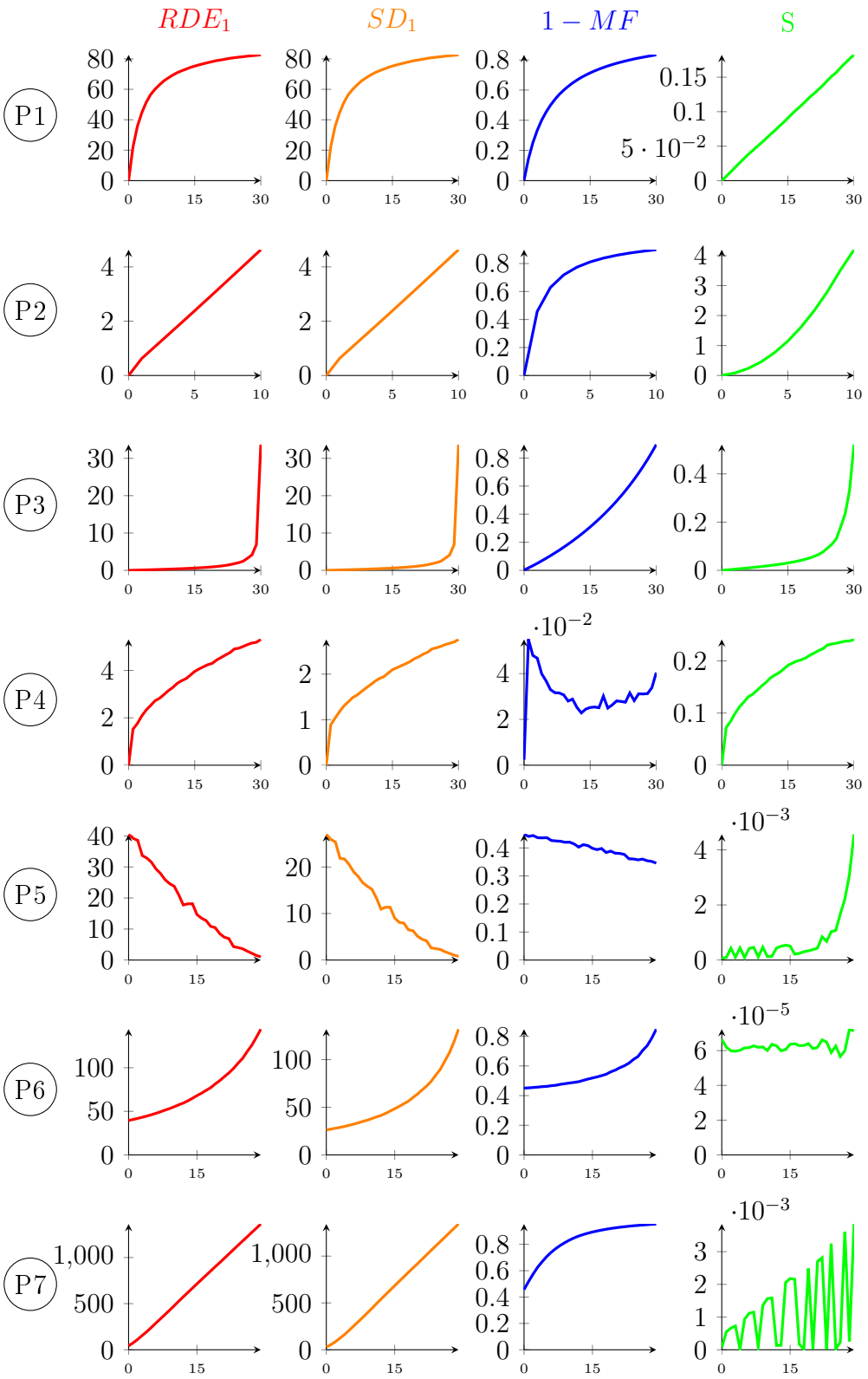


FIGURE 3.11 – Comparaison des métriques (colonnes) en fonction des différentes perturbations (lignes), moyennées sur 10 contours initiaux différents.

3.5 Fonction de pénalité pour amincir la prédiction des contours

3.5.1 Motivation

Notre but est de proposer une méthode pour résoudre le problème de OC2D de bout en bout. Nous avons pris soin de valider les critères C1-C5, en particulier C3, c'est-à-dire de retourner une seule réponse par pixel vrai-contour. C'est probablement le critère le plus complexe puisque l'épaisseur des réponses est une des limites principales des méthodes courantes de détection de contours basées sur des CNNs. Le problème est aussi un classique dans la détection d'arêtes à partir du gradient d'une image. Dans cet exemple, ces détecteurs se déclenchent dès que la magnitude du gradient est plus grande que le seuil de détection. Un seuil bas tend à sur-détecter et ne respecte donc pas le critère C3, tandis qu'un seuil trop haut tend à augmenter le nombre de FN, mettant en défaut le critère C1. Trouver un seuil pour respecter les critères C1 et C3 ensemble n'est souvent pas possible. Le populaire détecteur de Canny [18] résout cette impasse en utilisant un seuil bas et en appliquant des opérations morphologiques pour amincir les contours.

Le détecteur proposé s'inspire du détecteur de Canny mais utilise un CNN et une méthode pour entraîner ce réseau de bout-en-bout. L'idée principale est de définir des pénalités à intégrer à la fonction de coût. Nous avons utilisé un réseau d'architecture U-Net car il présente de bonnes performances en segmentation sémantique avec une quantité limitée de données d'entraînement.

Nous produisons trois cartes de probabilités $\mathcal{P} = \{p_{\text{oc}}, p_{\text{ob}}, p_{\text{bg}}\} \in [0,1]^3$ respectivement pour les contours occultants, les contours d'occultation, et les pixels non-contours. Nous utilisons une fonction *Softmax* pour garantir $p_{\text{oc}} + p_{\text{ob}} + p_{\text{bg}} = 1$.

Nous proposons un entraînement en 3 étapes, qui intègre progressivement deux nouvelles pénalités dans la fonction de coût : la Pénalité de Binarisation (BiP) et la Pénalité d'Amincissement (*Thinning Penalty*, TiP).

3.5.2 Définition des pénalités

3.5.2.1 Première étape d'entraînement : apprentissage de la tâche initiale

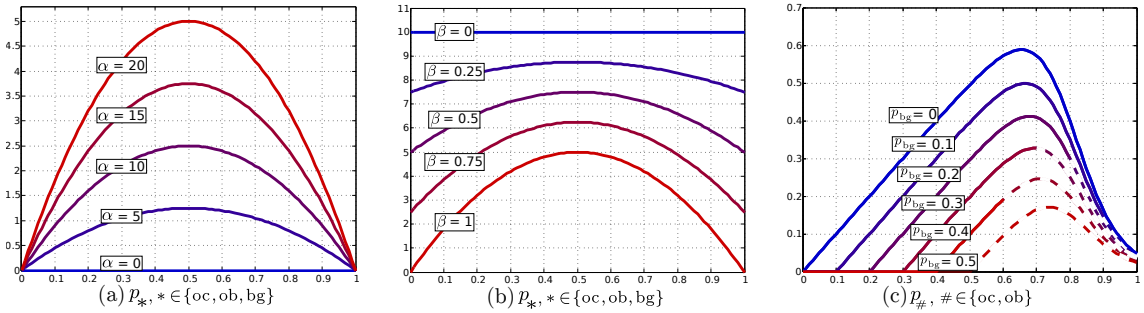
La première étape d'entraînement consiste à spécialiser le modèle à la tâche d'OC2D en utilisant l'entropie croisée de manière classique :

$$\mathcal{L}_1(\mathcal{P}, \mathcal{Y}) = \sum_{*\in\{\text{oc,ob,bg}\}} \mu_* \mathcal{L}_{\text{CE}}(p_*, y_*), \tag{3.13}$$

avec $*$ qui parcourt les 3 classes, $\mathcal{Y} = \{y_{\text{oc}}, y_{\text{ob}}, y_{\text{bg}}\} \in \{0;1\}^3$ sont les vrais labels avec $y_{\text{oc}} + y_{\text{ob}} + y_{\text{bg}} = 1$, $\mu_{\text{oc}} = 1$, $\mu_{\text{ob}} = 1,5$ et $\mu_{\text{bg}} = 0,01$ sont les poids prédéfinis, et $\mathcal{L}_{\text{CE}}$ la fonction de l'entropie croisée. L'entraînement est arrêté lorsque la prédiction du modèle ne s'améliore plus.

3.5.2.2 Deuxième étape d'entraînement : binarisation

La deuxième étape d'entraînement affine les résultats obtenus par le modèle de la première étape en binarisant la sortie, comme la binarisation d'une image.



(a) $p_*, * \in \{\text{oc}, \text{ob}, \text{bg}\}$ (b) $p_*, * \in \{\text{oc}, \text{ob}, \text{bg}\}$ (c) $p_\#, \# \in \{\text{oc}, \text{ob}\}$

FIGURE 3.12 – (a) Pénalité de binarisation avec la stratégie d'amplitude, BiPα, avec $\alpha \in [0; 20]$ et $\beta = 1$. (b) Pénalité de binarisation avec la stratégie de fréquence, BiPβ, avec $\alpha = 20$ et $\beta \in [0; 1]$. (c)Pénalité d'amincissement, TiP, pour $p_{\text{bg}} \in [0; 0,5]$. Les parties hachurées ne sont pas réalisables, avec $p_{\text{bg}} + p_\# > 1$.

Cette étape combine la fonction d'entropie croisée avec une nouvelle pénalité de binarisation (BiP) B conçue pour favoriser des sorties binaires, c'est-à-dire dont les valeurs sont proches de 0 ou 1 :

$$\mathcal{L}_2(\mathcal{P}, \mathcal{Y}) = \mathcal{L}_1(\mathcal{P}, \mathcal{Y}) + \sum_{*\in\{\text{oc,ob,bg}\}} \alpha\big((1-\beta)K + \beta \text{B}(p_*)\big). \tag{3.14}$$

B est combiné avec une constante K, et utilise deux hyper-paramètres α et β qui sont modifiés au cours de l'entraînement. Nous proposons $\text{B}(x) = x(1-x)$ comme la plus simple forme de BiP. Nous proposons deux stratégies qui ont pour but d'augmenter progressivement l'effet de BiP, illustré sur la figure 3.12. La *stratégie d'amplitude*, noté BiPα, où α augmente progressivement de 0 à 20 tandis que β est fixé à 1. La *stratégie de fréquence*, noté BiPβ, où β augmente progressivement de 0 à 1 tandis que α est fixé à 20. Les valeurs de α et β sont augmentées de 0,05 après chaque époque. L'entraînement est arrêté lorsque la prédiction du modèle ne s'améliore plus, et on conserve le meilleur modèle pour les valeurs $\alpha_{\text{opt}}, \beta_{\text{opt}}$. Les sorties du modèle sont toujours comprises dans l'intervalle $[0; 1]$ mais sont très proches de $\{0; 1\}$.

3.5.2.3 Troisième étape d'entraînement : amincissement

La troisième étape d'entraînement adapte le modèle pour favoriser les réponses fines, tout comme une opération d'amincissement avec des opérations morphologiques. Cette étape combine l'entropie croisée avec une nouvelle pénalité d'amincissement, nommée *Thinning Penalty* en anglais (TiP). Cette fonction pénalise les pixels dont la probabilité d'être un contour occultant est plus grande que celle de ne pas en être une tout en étant proche. Sa formule est donnée par :

$$\mathcal{L}_3(\mathcal{P}, \mathcal{Y}) = \mathcal{L}_1(\mathcal{P}, \mathcal{Y}) + \sum_{\#\in\{\text{oc,ob}\}} \gamma \max(0, p_\# - p_{\text{bg}}) \sigma(\theta(\lambda - p_\#)), \tag{3.15}$$

où # décrit les classes de contour. Dans le terme Tip, présenté sur la figure 3.12, γ est un hyperparamètre qui varie dans l'intervalle $[0, 40]$, augmentant d'un pas de 0,05 par époque. Le premier facteur pénalise les pixels pour lesquels $p_\# > p_{\text{bg}}$ linéairement en fonction de l'écart. Le second facteur pénalise les pixels pour lesquels $p_\# < \lambda$, avec $\lambda \in [0, 1]$ un seuil constant que l'on fixe à 0,8. Une valeur proche de 1 signifie que seuls les pixels proches des vrais contours doivent être détectés. Il utilise la

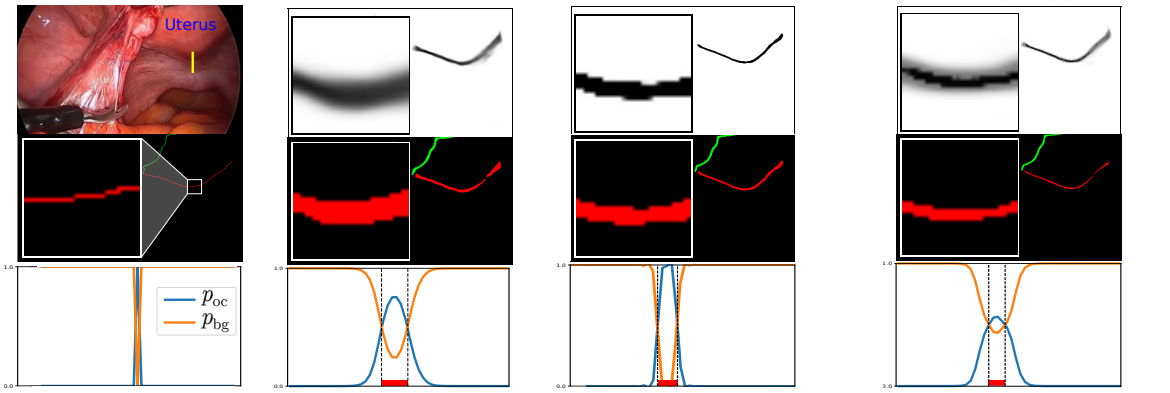


FIGURE 3.13 – (colonne 1, haut) Image d'entrée utilisée pour la tâche d'OC2D sur l'utérus, avec une coupe perpendiculaire à un contour occultant en jaune. (colonne 1, milieu) Vrais contours occultants en rouge et contours d'occultation en vert. Le carré représente un zoom sur la zone d'intérêt. (colonnes 2-4, haut) Carte de probabilité de la sortie du réseau p_{oc} après chaque étape d'entraînement EC est Entropie Croisée, BiP est notre pénalité de binarisation et TiP celle d'amincissement. (colonnes 2-4, milieu) Résultats de la prédiction OC2D. (toutes les colonnes, bas) Probabilités p_{oc} et p_{bg} obtenues le long de la coupe sélectionnée. Les lignes hachurées verticales représentent la transition entre la prédiction du contour et la prédiction du fond et vice-versa.

fonction sigmoïde σ et une valeur de pente $\theta = 15$. L'entraînement est stoppé quand le modèle cesse de s'améliorer, on garde le meilleur modèle pour l'hyper-paramètre γ_{opt}. Les sorties obtenues avec cette dernière étape présentent des contours bien plus fins.

3.5.3 Entraînement et évaluation

3.5.3.1 Tâche prétexte

Il est courant de recourir à un réseau pré-entraîné sur une tâche similaire pour initialiser les poids du réseau utilisé. Dans notre situation, nous avons préféré utiliser un réseau que nous avons pré-entraîné avec une tâche prétexte. Une tâche prétexte est une tâche pour laquelle il est possible d'automatiser l'annotation, qui sert d'intermédiaire à la tâche cible. Le choix de cette tâche a deux limites. Il faut que cette tâche permette au réseau de se familiariser avec le type de données utilisées. Deuxièmement, cette tâche ne doit pas nécessiter d'annotation coûteuse.

Pour pré-entraîner le réseau, nous avons décidé d'entraîner le réseau U-Net à reproduire les contours prédits par un détecteur de Canny avec un paramétrage adapté à notre type de données. Ce paramétrage vise notamment à s'assurer que les contours que l'on veut extraire dans la tâche finale (contours occultants et occultés de l'utérus) sont bien inclus dans les contours extraits via l'approche classique de Canny. Cette approche a également l'avantage de proposer une sur-détection des contours. On sait que la répartition des classes est très déséquilibrée en faveur de la classe de non-contours. Cela implique que le réseau va avoir tendance à prédire la classe sur-représentée. Avoir un état initial qui va à l'encontre de ce déséquilibre va nous permettre de favoriser la détection des contours.

3.5.3.2 Vue d'ensemble de l'évaluation

Nous avons évalué notre méthode OC2D proposée pour chacune de ses étapes contre l'approche de référence et les travaux existants. On désigne la première étape d'entraînement, un réseau U-Net entraîné avec une entropie croisée, l'approche de référence. L'entraînement présenté dans §3.5.2 est CE-BiPx-TiP, avec $x \in \{\alpha, \beta\}$. On utilise le symbole '-' pour séparer chaque étape d'entraînement. De manière à comprendre le rôle de chaque fonction de coût des étapes d'entraînements, nous avons distinguer 4 scenarii dont les noms décrivent explicitement l'approche : CE-BiPx, CE-BiPα-BiP+TiP et CE-TiP, où '+' traduit l'agrégation de termes de fonctions de coût. Nous comparons les résultats avec CASENet [125].

3.5.3.3 Implémentation

Nous utilisons l'implémentation d'U-Net et CASENet avec Pytorch de [101] et [4] respectivement. Nous avons affiné (*fine-tune* en anglais) le réseau CASENet avec notre jeu de données à partir du modèle pré-entraîné sur Semantic Boundary Dataset (SBD) [53]. Nous avons utilisé une descente de gradient stochastique (SGD) et multiplié le taux d'apprentissage initial par 0,1 toutes les 10 époques.

3.5.3.4 Résultats

Plusieurs résultats quantitatifs et qualitatifs de la méthode OC2D proposée appliquée sur des exemples particulièrement difficiles sont présentés sur la figure 3.15. Ils montrent en particulier la robustesse de la méthode OC2D proposée avec des occultations importantes de l'utérus, la présence de sang et de fumée.

La figure 3.13 illustre visuellement le résultat des différentes étapes de l'entraînement. Sur cet exemple, on peut constater l'effet de binarisation de l'étape 2 et l'amincissement de l'étape 3.

Les performances quantitatives pendant l'entraînement sont présentées dans la figure 3.14, en utilisant le score proposé S, et une décomposition de ses 3 termes S_{TP}, S_{FP} et S_{FN}. A part pour CASENet qui montre de très mauvais résultats, nous observons que l'approche de référence EC présente les moins bons résultats. L'étape de binarisation améliore les résultats en comparaison à l'approche avec EC, de manière similaire que ce soit avec l'approche CE-BiPα ou CE-BiPβ. L'entraînement complet CE-BiPα-TiP avec les deux termes de pénalités obtient les meilleurs résultats, améliorant particulièrement les termes sur les vrais-positifs et sur les faux-positifs. Par contre, on constate que cette approche dégrade un peu le terme sur les faux-négatifs. On peut faire l'hypothèse que l'amincissement des contours tend à provoquer une sous-détection des contours. CE-TiP, qui passe sur l'étape de binarisation, montre de moins bons résultats. Curieusement, CE-BiPα-BiP+TiP, qui inclut les deux pénalités dans la dernière étape d'entraînement, obtient des résultats très proches de la méthode CE-BiPα-TiP. Les scores sur les FN s'améliorent mais ceux sur FP et TP sont dégradés.

Les mauvais résultats de CASENet peuvent s'expliquer par le nombre limité d'images utilisées pour l'entraînement en comparaison du nombre d'éléments à entraîner. Il est à noter que nous n'avons pas donné la même priorité à faire fonctionner cette méthode pour nous concentrer sur nos méthodes originales.

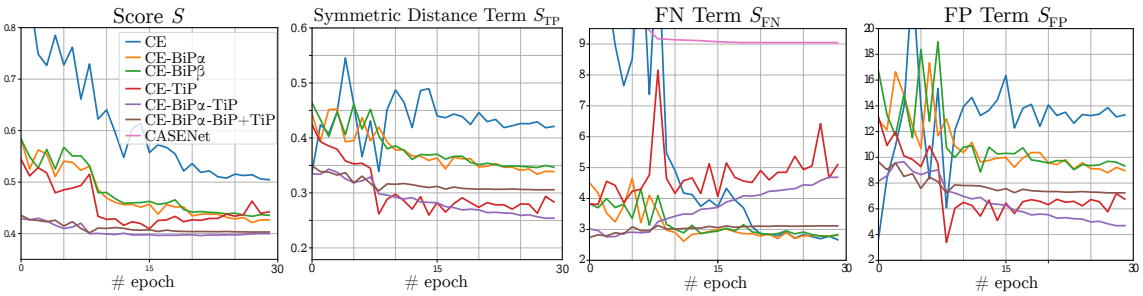


FIGURE 3.14 – Performances quantitatives évaluées pendant l'entraînement de OC2D. De gauche à droite, le score global $S = \frac{1}{d_{\max}}(S_{TP} + S_{FP} + S_{FN})$ et ses trois termes S_{TP}, S_{FP} et S_{FN}. Les résultats pour la méthode CASENet sont en dehors du graphique pour les scores S, S_{TP} et S_{FP}.

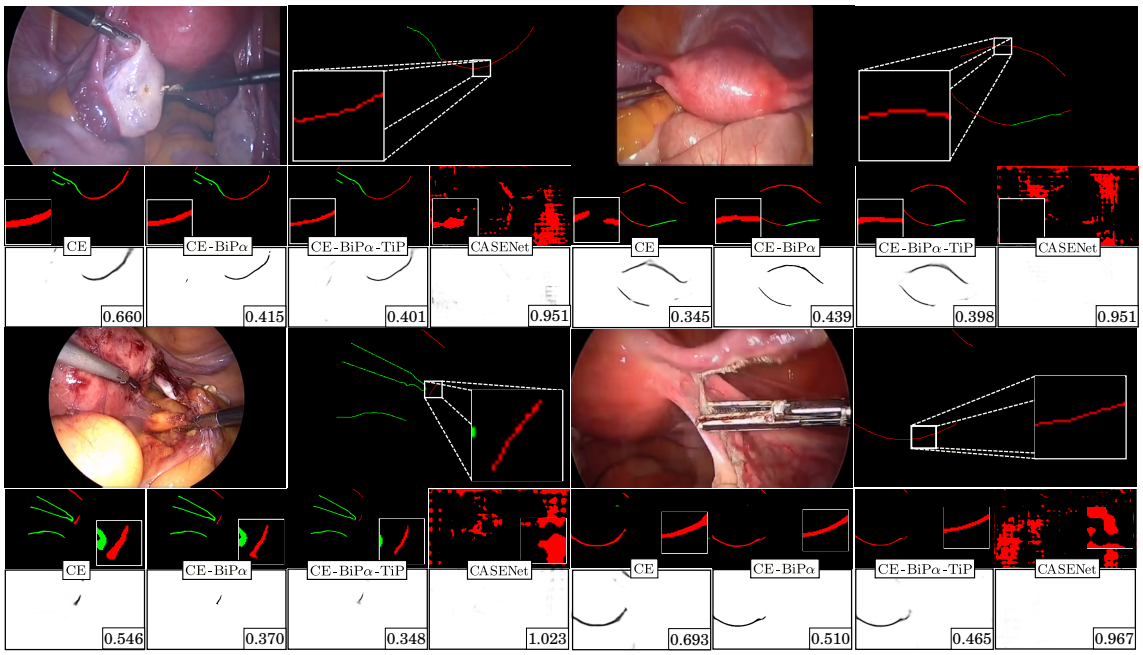


FIGURE 3.15 – Les réponses de notre méthode OC2D pour 4 exemples. Pour chaque exemple, la première ligne représente l'image cœlioscopique en entrée et la vérité terrain annoté manuellement. Les contours occultants sont marqués en rouge et les contours d'occultation en vert. La deuxième ligne présente la réponse de 4 détecteurs. La dernière ligne décrit les probabilités en sorties du réseau pour la classe "contour occultant" avec le score S correspondant.

3.6 Étude de cas : application de la détection automatique de contours occultants pour le logiciel de réalité augmentée Uteraug

3.6.1 Méthodologie

Nous disposons de 10 jeux de données complets d'opérations de cœlioscopie pour permettre de reproduire l'utilisation du logiciel dans les conditions d'une opération réelle. Ces jeux de données contiennent en particulier les images peropératoires qui doivent être annotées, en moyenne 18 par opération, et une vidéo qui servira de support à la réalité augmentée.

Les méthodes d'annotation dites automatiques sont les méthodes OC2D développées dans le chapitre 3, c'est-à-dire la méthode de référence CE, la méthode CE-BiPα-TiP et CASENet.

L'annotation manuelle a été réalisée par 5 chirurgiens, composés de 3 internes et 2 seniors. Toutes les personnes impliquées étaient familières avec l'utilisation du logiciel et ont eu l'occasion de tester le système d'annotation tactile avant la phase de test. Pour reproduire des conditions d'annotations proches de celles de la salle d'opération, on a demandé aux annotateurs leur propre compromis entre annoter les contours le plus rapidement possible et le plus précisément possible. Les annotateurs étaient conscients d'être chronométrés.

Pour chaque méthode, les contours générés sont utilisés pour contraindre l'étape de recalage qui permet de déformer le modèle préopératoire pour correspondre avec l'état de l'organe dans la salle d'opération. Le logiciel [23, 26] réalise l'augmentation de la vidéo en utilisant le modèle recalé.

Nous évaluons la précision en évaluant l'erreur de reprojection du modèle 3D virtuel sur 10 images extraites de la séquence vidéo. L'erreur de reprojection est définie par la distance moyenne entre les contours occultants du modèle 3D et leur annotation réalisée avec précision. Les 10 images ont été sélectionnées de sorte à proposer des points de vues différents et pour lequel au moins 10% du modèle 3D virtuel est reprojeté dans l'image.

Cette expérience a été réalisée pour les 10 opérations, avec les annotations de 5 chirurgiens et 3 méthodes OC2D, ce qui donnent un total de 80 cas.

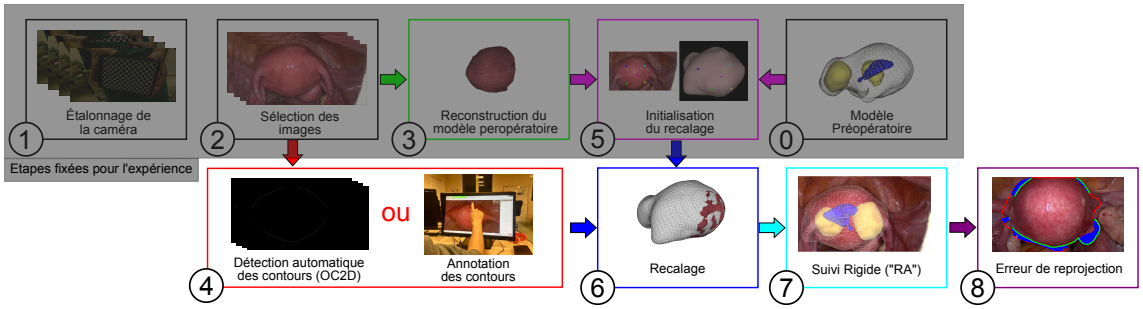


FIGURE 3.16 – Explication schématique de l'utilisation du logiciel Uteraug pour évaluer l'impact des contours. La partie grisée du schéma est constante pour toutes les méthodes. Les contours sont générés via annotation manuelle ou via les méthodes OC2D (4). Ces contours modifient le recalage (6). La silhouette du modèle virtuel reprojeté est enregistrée au cours de la cœlioscopie augmentée (7). (8) On mesure la distance (bleu) entre les contours occultants (vert) et la silhouette (rouge).

3.6.2 Résultats

Les résultats sont présentés dans le tableau 3.1. On observe que EC et EC-BiPα-TiP donnent une erreur de recalage quasiment identique à celle obtenue grâce à l'annotation manuelle. Ces méthodes OC2D permettent d'obtenir ces résultats en réduisant le temps d'annotation de 3 minutes et 53 secondes, ce qui représente une réduction de 97,4%. L'exécution des méthodes automatiques n'a pas été optimisée. Les temps affichés dans le tableau 3.1 correspondent à l'ensemble des étapes qui permettent de réaliser cette annotation automatique. Ce temps inclut donc le chargement des images, l'inférence des images dans le réseau de neurones, le redimensionnement des contours et leur conversion dans le format utilisé par le logiciel. Le temps de la détection des contours seule représente 25% du temps affiché ce qui représente environ 86 ms par image.

Malgré les résultats complètement aberrants de CASENet (voir 3), l'erreur moyenne observée n'augmente que de 14 pixel par rapport aux méthode EC et EC-BiPα-Tip, ce qui n'est pas si conséquent par rapport à ce qui était attendu. Une hypothèse est que le M-estimateur utilisé pour prendre en compte les contours occultants dans le recalage permet de ne pas considérer ces résultats aberrants. Il faut également noter que le recalage est contraint par un terme qui minimise l'écart entre les surfaces des modèles préopératoire et peropératoire, et par un terme de régularisation. Ces termes permettent de limiter une déformation trop importante incitée par des contours très lointains.

On peut également constater que l'erreur est grande, quelle que soit la méthode considérée. 60 px représente environ 2,7 % de la diagonale des images utilisées (1080p). Cette erreur est perceptible par l'utilisateur qui peut alors voir une différence entre la position réelle et la position virtuelle de l'organe suivi.

Cas	Manuel	EC-BiPα-TiP	EC	CASENet	Temps OC2D	Temps Manuel
1	**34,97**	42,13	42,89	71,85	7,7"	4'56"
2	56,41	53,93	**53,14**	60,58	7,8"	5'12"
3	**93,40**	94,83	95,92	127,44	6,7"	4'39"
4	42,13	**40,84**	43,42	50,46	7,8"	5'37"
5	85,13	88,10	**80,33**	93,31	8,5"	4'34"
6	90,47	**90,07**	91,37	100,30	5,1"	3'37"
7	96,34	90,62	92,17	**84,72**	6,6"	3'24"
8	**46,76**	48,56	49,03	54,58	5,5"	3'28"
9	**32,27**	33,92	33,47	49,30	6,1"	3'30"
10	39,58	41,43	**38,39**	67,00	2,2"	1'28"
Moyenne	**61,75**	62,44	62,01	75,95	6,4"	4'02"

TABLE 3.1 – Étude de cas pour 10 cœlioscopies. Les résultats manuels sont moyennés pour les 5 chirurgiens. L'erreur de reprojection est exprimée en pixel (on veut une erreur faible). Le temps est exprimé en minutes (') et secondes ("). Le temps OC2D est évalué pour la méthode EC-BiPα-TiP, mais les autres méthodes présentes des valeurs similaires. Les cas contiennent en moyenne 18 images à annoter.

3.7 Conclusion

Nous avons proposé un nouveau score pour comparer deux courbes échantillonnées sur des images. Ce score permet de respecter les différents critères de Canny ce qui demande de trouver un équilibrage pour pénaliser équitablement les fausses réponses, les vrais-contours manqués et les prédictions épaisses. Notre score tend à pénaliser fortement les prédictions épaisses et assez faiblement les fausses réponses. Ce sont des contraintes adaptées à notre usage puisque les solutions qui utilisent des réseaux convolutifs ont tendance à produire des réponses épaisses mais avec un nombre de fausses réponses limitées.

Nous avons utilisé ce score pour évaluer une méthode de détection de contours occultants. Cette méthode basée sur l'utilisation d'un encodeur-décodeur propose plusieurs étapes d'entraînement pour permettre d'affiner les contours prédits sans utilisation d'opérations morphologiques.

Ces méthodes ont été couplées avec l'utilisation du logiciel Uteraug. Une étude de cas a été proposée pour montrer l'intérêt de notre méthode en termes de précision et de temps gagné. Cette étude montre la faisabilité de l'utilisation des méthodes OC2D pour remplacer l'annotation automatique sans perte de précision. Pour envisager une intégration complète dans le logiciel Uteraug et pour une utilisation réelle, une première étape serait de proposer les contours annotés automatiquement à un opérateur humain qui pourrait alors les valider et les modifier si besoin. Cette étude est un premier pas pour la validation de l'utilisation de l'annotation automatique de contours occultants de l'utérus.

Lors d'autres tests des approches $BiP{-}TiP$, notamment sur l'expérience de l'annexe A qui teste l'impact du nombre d'images de notre jeu de données, nous avons pu constater que ces approches permettent des améliorations lorsque la méthode avec l'Entropie Croisée prédit des contours épais. Lorsque celle-ci tend à sous-détecter les contours, la binarisation de l'approche BiP et surtout l'amincissement de la méthode TiP dégrade les résultats. Il est également juste de préciser que les prédictions obtenues en appliquant une étape d'amincissement avec des opérations morphologiques, comme le squelette morphologique, donnent encore de meilleurs résultats avec le

score S. Cela signifie que la potentielle amélioration de l'alignement de nos entraînements ne compense pas l'épaisseur des contours dans le score S. Cela peut venir en partie du fait que ce score pénalise fortement les prédictions épaisses. Néanmoins, cela signifie que notre approche ne parvient pas à atteindre l'épaisseur minimale pour les contours prédits. Il pourrait être intéressant de combiner notre approche avec les approches proposées par des travaux comme STEAL [4] ou SEAL [126]. Ces méthodes proposent de réaligner les annotations pendant l'entraînement afin de limiter la confusion apportées par les annotations qui utilisent des polygones. Ces travaux proposent également l'ajout de fonctions de coût dédiées pour vérifier que l'angle des contours soit cohérent et qu'un contour ne provoque qu'une seule prédiction.

Chapitre 4

Détection d'incision et mise à jour topologique du modèle 3D peropératoire

Ce chapitre est basé sur notre contribution [38], publiée pour le congrès international MICCAI 2021. Un certain nombre d'extensions ont été proposées par rapport au contenu de ce premier article et la rédaction d'un article de journal est en cours en conséquence.

4.1 Introduction

La RA est obtenue en superposant les structures anatomiques internes, extraites depuis une IRM ou une TDM, au flux vidéo cœlioscopique. La RA repose donc sur la capacité de calculer les transformations géométriques entre le modèle 3D préopératoire et les images du flux vidéo cœlioscopique. Il s'agit du problème de recalage, qui représente un enjeu technique majeur à cause des déformations de l'organe. Les méthodes existantes de recalage en temps réel fonctionnent en deux étapes [26, 55, 51]. Premièrement, elles déforment le modèle 3D préopératoire pour correspondre aux déformations de l'organe. Cette étape utilise un modèle 3D reconstruit à partir de méthodes comme *Structure-from-Motion* (SfM), SLAM ou d'une vision stéréo. Le modèle 3D préopératoire déformé forme alors le modèle 3D peropératoire. Deuxièmement, ces méthodes calculent le suivi de l'organe en utilisant le modèle 3D peropératoire. Cette dernière étape fonctionne correctement tant que l'organe est manipulé sans changer sa forme. Ainsi, une fois que le/la chirurgien(ne) commence l'incision de l'organe, la forme de l'organe est fortement affectée, ce qui met en échec les méthodes existantes de recalage. En conséquence, la RA n'est actuellement disponible que pendant les premières phases de la chirurgie et s'arrête une fois que l'incision de l'organe débute. Notre objectif principal est de développer une méthode qui permet de répondre à cette limite majeure.

Nous avons proposé un processus complet de recalage qui réalise le suivi de l'organe et met à jour le modèle 3D peropératoire en fonction des transformations observées sur l'organe pendant la chirurgie. Concrètement, cela signifie que nous mettons à jour le modèle 3D peropératoire. Sans cette mise à jour, la topologie du modèle 3D peropératoire, qui est un modèle 3D virtuel, ne correspond plus à la topologie de l'organe observé lorsque le/la chirurgien(ne) incise l'organe. L'adaptation

gère les changements de la forme du modèle 3D dus à la déformation de l'organe et les changements de la topologie du modèle 3D dus à l'incision de l'organe. Une approche similaire à la nôtre est celle de Paulus et al. [94], qui adapte la topologie du modèle 3D peropératoire en utilisant un critère géométrique. L'idée clé de Paulus et al. [94] est de calculer le recalage et de détecter les incisions seulement à partir de ce résultat. Concrètement, l'allongement excessif d'une arête du maillage 3D déclenche la suppression de cette arête. C'est une idée intéressante mais qui fonctionne mal en pratique car elle nécessite que le recalage soit très précis : le moindre bruit dans le recalage, même temporaire et léger, peut créer l'apparition d'une incision fallacieuse, sans possibilité de pouvoir la rétablir plus tard. Puisque le recalage déformable est un problème difficile, et puisque la topologie du modèle 3D peropératoire est par définition fausse avant le recalage, on ne peut attendre du recalage d'être toujours très précis.

Notre processus présente une nouvelle idée clé pour mettre à jour le modèle 3D peropératoire : la détection d'incision sur l'image. En pratique, les berges visibles de l'incision sont détectées sur l'image courante extraite du flux vidéo cœlioscopique. Il est important de noter que cette détection est indépendante du recalage. Contrairement au critère géométrique de Paulus et al. [94] qui dépend du recalage, notre détection d'incision sur l'image peut être exploitée pour renforcer le calcul du recalage. Nous utilisons un réseau convolutif (CNN) pour détecter les berges visibles de l'incision dans l'image ; leurs positions sont ensuite transférées vers le modèle 3D peropératoire, et nous mettons à jour la topologie du modèle 3D avant de calculer le recalage non-rigide.

Transférer les berges de l'incision vers le modèle 3D peropératoire est une étape clé, qui est obtenue grâce à une fonction de déformation de l'image qui exploite les berges de l'incision explicitement. Intuitivement, on veut que la fonction de transformation ferme l'incision de l'image courante, où l'incision a été détectée, vers les images de références, où l'organe est encore intact. On rappelle que les images de référence ont servi à la reconstruction du modèle 3D peropératoire par SfM et qu'il existe donc une correspondance dense entre elles et avec le modèle SfM. Les incisions ont des similarités avec les auto-occultations, qui ont été étudiées avec attention [44, 96]. Pour les deux cas, la transformation doit adopter un comportement spécifique sur une zone, que l'on appelle zone de singularité, définie dans l'image source, pour que cette zone soit réduite à une simple courbe dans l'image cible. Pour gérer les auto-occultations, Gay-Bellile et al. [44] ont proposé l'ajout d'un terme d'amincissement dans la fonction de coût, nommé *shrinker* ("to shrink" en anglais signifie rétrécir, réduire). Nous le démontrerons plus tard, mais les incisions et les auto-occultations restent, malgré leurs similitudes, fondamentalement différentes. En quelques mots, la zone de singularité est une donnée pour estimer la transformation qui ferme l'incision tandis que pour les auto-occultations, il est nécessaire d'estimer à la fois la transformation et la zone de singularité en même temps.

Nous avons évalué notre processus en 4 parties. Premièrement, nous avons évalué notre détecteur d'incision sur des images cliniques annotées manuellement en utilisant des métriques de détection de bords. Deuxièmement, nous avons évalué l'étape de transfert d'incision. Troisièmement, nous avons évalué les bénéfices de mettre à jour la topologie du modèle 3D peropératoire dans une expérience utilisant 5 reins de porc ex-vivo. Finalement, nous avons testé notre processus sur des vidéos de chirurgie de cœlioscopie de manière qualitative.

4.2 État de l'art et contributions

4.2.1 Détection d'incision basée sur l'image

La détection d'incision basée sur l'image n'a pas été adressée spécifiquement dans la littérature scientifique, mais de nombreuses tâches de détection et de segmentation ont été explorées [70]. En chirurgie assistée par robot, *Robot-Assisted Surgery* (RAS), la détection de coupure automatique a été récemment explorée pour contrôler l'incision réalisée par un robot en utilisant des indices visuels [50]. En cœlioscopie, la détection et la segmentation d'outils [42, 54], et la détection de contours spécifiques à certains organes [55, 39] sont des sujets de recherche actifs. Ces travaux entraînent un réseau de type encodeur-décodeur dérivé de l'architecture U-Net [101] avec un jeu de données dédié. Nous avons ainsi essayé d'entraîner un réseau U-Net pour résoudre la détection d'incision, ce qui n'a pas encore été tenté dans des travaux précédents.

4.2.2 Recalage d'organes déformables

Dans la littérature, il y a deux approches majeures pour résoudre le recalage initial entre le modèle 3D préopératoire et les images cœlioscopiques 2D : le recalage rigide et déformable.

Le recalage rigide n'est pas adapté à la cœlioscopie puisque les organes se déforment, en particulier à cause de l'insufflation de CO2. Ces méthodes alignent manuellement le modèle 3D [99] ou appliquent des méthodes semi-automatiques pour aligner des repères anatomiques marqués manuellement [90, 81, 35]. Les méthodes de recalage déformable automatiques sont donc les seules options viables. Ces méthodes se décomposent en deux catégories. La première catégorie implique des techniques de simulation pour modéliser les forces appliquées à l'organe. La seconde catégorie résout un problème de minimisation d'une énergie composée d'un terme d'attache aux données et d'un terme de régularisation. Le terme d'attache aux données évalue à quel point le modèle 3D recalé correspond aux images cœlioscopiques. Il est généralement basé sur des repères anatomiques ou les contours de l'organe. Le terme de régularisation contrôle la déformation avec comme *a priori* la régularité de la déformation [6, 26], ou des contraintes biomécaniques [5, 52]. Ces méthodes nécessitent un paramétrage spécifique à l'organe concerné, en particulier pour contrôler l'influence du terme de régularisation. Une troisième catégorie de méthodes pourrait apparaître dans le futur, en suivant les évolutions de l'exploitation de méthodes d'apprentissage profond [115, 17, 95]. Les méthodes courantes de cette catégorie ne sont malgré tout pas encore applicables à la cœlioscopie monoculaire, puisqu'elles nécessitent le déplacement 3D de la partie visible de l'organe [95, 17] et semble ne fonctionner que pour des données synthétiques ou semi-synthétiques.

Il est important de noter que la grande majorité des méthodes existantes suppose que la topologie du modèle 3D est fixe et sont donc amenés à échouer très probablement si cette topologie ne représente pas l'état actuel de l'organe considéré. A notre connaissance, seule la méthode proposée par [94] met à jour la topologie du modèle 3D pendant le recalage.

Paulus et al. [94] utilise une méthode par éléments finis, *Finite Element Method* (FEM) en anglais, pour simuler l'énergie de déformation du modèle 3D et

contrôle visuellement cette déformation grâce à des points-clés SURF. Une métrique est proposée pour évaluer l'évolution de la distance entre les points déformés et leurs correspondances. Si une paire de points est déplacée de manière significative, un point d'incision est ajouté. Les expériences décrites dans [94] présentent une incision profonde et très longue avec une déformation limitée mais avec un mouvement quasiment rigide des parties de l'organe. La méthode nécessite que le recalage soit résolu parfaitement depuis les points SURF, ce qui est peu probable en pratique.

4.2.3 Modèle de déformation de l'image

Recaler deux images optiques est une tâche importante pour développer des méthodes telles que la reconstruction 3D d'objets déformables [92]. Cela nécessite de trouver les transformations géométriques et photométriques qui permettent de rendre les images aussi similaires que possibles. La transformation géométrique, appelée *warp* en anglais, modifie la position des pixels tandis que la transformation photométrique modifie la valeur des pixels. Les modèles de transformation géométrique populaires sont les fonctions à base radiale, *Radial Basis Functions* (RBF) en anglais, comme la Thin-Plate Spline (TPS)

Les paramètres de la transformation géométrique sont calculés en minimisant une fonction de coût, composée le plus souvent d'un terme d'attache aux données et d'un terme de régularité. Il y a deux types de terme d'attache aux données : ceux basés sur les pixels et ceux basés sur des points-clés. Les termes basés sur les pixels mesurent l'écart photométrique alors que ceux basés sur les points-clés mesurent la distance entre des points correspondants. Dans les images cœlioscopiques, la couleur n'est pas aussi discriminante pour obtenir une solution précise de recalage. Les solutions basées sur les points-clés sont ainsi préférées [82, 13]. Le terme de lissage, appelé *smoother* en anglais, contrôle la régularité du champ de déplacement à partir des dérivées secondes de la transformation.

Les occultations représentent une des difficultés majeures du recalage d'image. On peut les distinguer en occultations externes et auto-occultations. Les occultations externes sont causées par un objet qui cache la surface d'intérêt. Les auto-occultations apparaissent quand une partie de la surface d'intérêt est courbée de sorte à cacher une partie d'elle-même. Les occultations externes sont gérées efficacement grâce à la combinaison d'un terme d'attache aux données robuste et d'un terme de lissage. Une difficulté spécifique aux auto-occultations est qu'il faut à la fois détecter la zone auto-occultée et estimer la transformation. Dans la zone occultée, la transformation géométrique est contraint uniquement par le terme de lissage. Dans les zones auto-occultées, le terme de lissage peut ne pas être suffisant puisque l'optimisation nécessite de courber la transformation d'une manière extrême. Une solution pour ce problème est d'utiliser un terme d'amincissement, en plus de l'attache aux données et du terme de lissage. Le terme d'amincissement garantit un comportement de la déformation spécifique dans la zone auto-occultée. Dans [44], deux types de termes d'amincissement ont été proposés pour gérer les auto-occultations. Le but de ces termes d'amincissement est d'une part de pénaliser les variations de signe des dérivées de la transformation et d'autre part, de forcer la transformation à s'effondrer le long de la direction des auto-occultations. La première contrainte va empêcher la transformation de plier la surface. Dans cette configuration, les zones auto-occultées correspondent aux zones où la norme de la dérivée directionnelle est faible dans une

Paulus et al. [94]	Gay-Bellile et al. [44]	Pizarro et al. [96]	Méthode proposée
incision	auto-occultation	auto-occultation	incision
◦ transformation (3D) ◦ détection de la singularité [images source et cible]	◦ transformation (2D) et détection de la singularité [images source et cible]	◦ transformation (2D) ◦ détection de la singularité [images source et cible] ◦ transformation (2D) avec contrainte	◦ détection de la singularité [image cible] ◦ transformation (2D) avec contrainte ◦ détection de la singularité [image source]

TABLE 4.1 – Description schématique des différentes approches présentées. Chaque élément de la liste indique une étape de la méthode décrite. Dans chaque colonne, on précise quel type de singularité est traitée : auto-occultation ou incision. Le terme "transformation" est utilisé dans un sens large pour désigner l'estimation des paramètres d'un modèle 2D ou 3D. La détection de la singularité peut être dans l'image source, cible ou les deux.

direction. Ce critère permet de détecter les zones d'auto-occultations sur lesquelles on peut appliquer le terme d'amincissement.

Pizarro et al. [96] ont proposé de renforcer le poids du terme de lissage dans les zones auto-occultées. Les zones auto-occultées sont détectées en repérant les zones de la transformation où le déterminant de la matrice jacobienne est négatif. Dans ces zones, cela signifie que le système de coordonnées est retourné. Comme énoncé plus tôt, les zones auto-occultées ne sont contraintes que par le terme de lissage. Ils ont décidé de renforcer le poids du terme de lissage dans ces zones pour forcer une transformation affine. L'avantage de cette approche est qu'elle n'ajoute pas de terme non-linéaire comme le terme d'amincissement et permet ainsi une résolution convexe.

Pour ces deux méthodes [44, 96], l'aire de la zone à auto-occulter est grande par rapport à la surface source. Elle dépasse les 50% dans une expérience de [44]. Les incisions observées dans notre contexte représente une portion bien plus faible de la surface visible de l'organe, environ 20% sur la figure 4.6. Pour ces deux méthodes, la transformation estimée sur l'image précédente est utilisée pour initialiser l'estimation sur l'image suivante.

4.2.4 Simulation d'incision

Couper un modèle 3D peut être obtenu de plusieurs manières [119], qui ont été soigneusement étudiées en géométrie algorithmique, *Computational Geometry* en anglais. Par exemple, des simulations pour mettre à jour la topologie de maillages ont été proposées pour gérer les coupures et les déchirures de tissus mous (*soft-tissues* en anglais) [15, 88, 28]. En général, un maillage volumétrique est utilisé pour représenter l'objet simulé avec une méthode par éléments finis pour la discrétisation numérique des équations. L'objet est représenté par une multitude de tétraèdres, ou hexaèdres. Certaines méthodes proposent même des méthodes sans maillage en utilisant des particules dispersées arbitrairement dans le domaine du problème. Ce dernier type de méthodes est particulièrement pertinent lorsque le modèle 3D est déformé de manière extrême et subit un nombre répété de coupures, ce qui peut générer des maillages mal définis avec les premières méthodes. Wu et al. [119] présentent une liste exhaustive

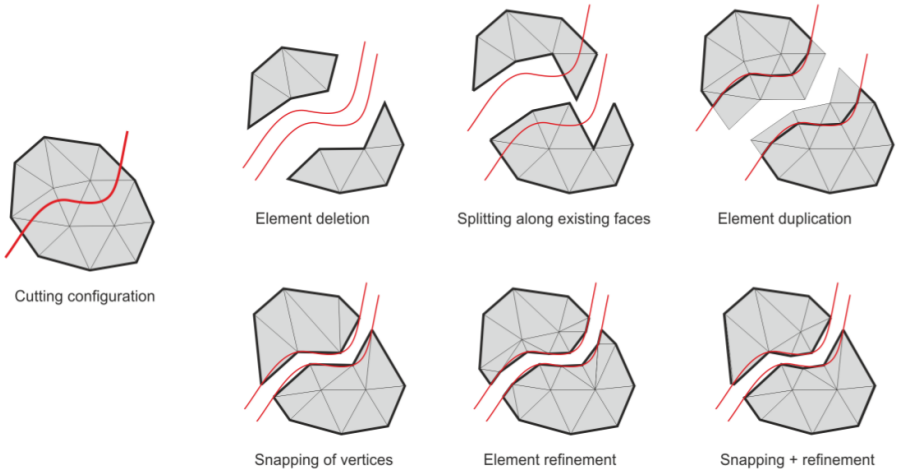


FIGURE 4.1 – Illustration (originale extraite de [119]) pour représenter les différentes techniques pour inciser un tétraèdre. Pour simplifier les illustrations, l'incision est réalisée sur un maillage plan composé de triangles. Toujours pour plus de visibilité, les deux morceaux de l'objet après incision sont écartés pour voir la topologie associée.

des manières d'incorporer des coupures dans un maillage tétraédrique. Les méthodes de coupure sont illustrées sur la figure 4.1. Nous avons décidé d'utiliser l'approche du raffinement des éléments, *element refinement* en anglais. Chaque tétraèdre traversé par la coupure est coupé et raffiné pour être décomposé en de nouveaux tétraèdres.

Tant que l'objet considéré n'est pas coupé avec répétition ou fortement déformé, avec un déchirement important par exemple, les méthodes décrites devraient fonctionner de manière adéquate.

4.2.5 Contributions

Nous avons proposé un processus pour la mise à jour topologique du modèle 3D peropératoire, qui permet au recalage et ainsi à la RA de supporter l'incision de l'organe. Notre processus exploite la détection de l'incision sur l'image et se décompose en 4 étapes :

- La détection de l'incision basée image, obtenue en entraînant un U-Net sur un nouveau jeu de données dédié à la détection d'incision. Nous proposons un jeu de données, constitué d'images cœlioscopiques, extraites de myomectomies, où les berges de l'incision sont annotées.
- Le transfert d'incision de l'image vers le modèle 3D, où l'incision détectée est transférée vers le modèle 3D peropératoire avant l'étape de recalage. Nous avons proposé de nouveaux termes pour la fermeture de l'incision.
- La mise à jour topologique du modèle 3D peropératoire depuis l'incision transférée.
- Le recalage non rigide entre le modèle 3D peropératoire mis à jour et l'image courante, inspiré de la méthode utilisée dans [23]. Nous avons proposé l'ajout d'un terme pour contraindre l'alignement de l'incision du modèle 3D peropératoire sur l'incision détectée dans l'image courante.

4.3 Cadre proposé

Notre processus est basé sur la méthode de recalage de l'utérus proposée dans [26]. Nous considérons que le modèle 3D peropératoire a été reconstruit avec succès en utilisant SfM (pour rappel, voir la section 2.4 dans le chapitre 2). Nous supposons qu'un jeu de correspondances $C_p = (p_1 \cdots p_N)$ et $C_q = (q_1 \cdots q_N)$ a été établi entre l'image courante $\mathcal{I}$ et une image-clé $\mathcal{T}$ utilisée dans le modèle SfM. Un jeu de correspondances peut être obtenu de manière automatique grâce à la détection et mise en correspondance de points-clés comme SIFT [76] ou SURF [11]. Dans les cas réels, et particulièrement avec des images cœlioscopiques, un nombre significatif de ces correspondances obtenues automatiquement peuvent être des fausses correspondances, appelés *mismatches* en anglais. Dans ce chapitre, nous ne nous intéresserons pas à la question de la mise en correspondance. Les correspondances de points utilisées ont été établies manuellement.

4.3.1 Vue d'ensemble

Mettre à jour le modèle 3D en fonction du flux vidéo cœlioscopique suppose de détecter l'incision à la fois temporellement et spatialement. Pour simplifier le problème, nous faisons les hypothèses suivantes : *(i)* Une unique incision est observée pendant la séquence vidéo. *(ii)* Il existe au moins une image dans laquelle l'incision est visible entièrement, ce qui est réaliste puisque le/la chirurgien(ne) vérifie l'état de l'incision au moins une fois son geste terminé. Bien sûr, la longueur de l'incision ne peut qu'augmenter avec le temps. Ces hypothèses nous permettent de détecter l'incision uniquement dans l'image courante et mettre à jour le modèle 3D peropératoire lorsque la longueur de l'incision détectée a grandi. Une incision plus courte sera détectée à chaque fois qu'elle sera occultée par du sang ou les outils chirurgicaux, ce qui est très probable.

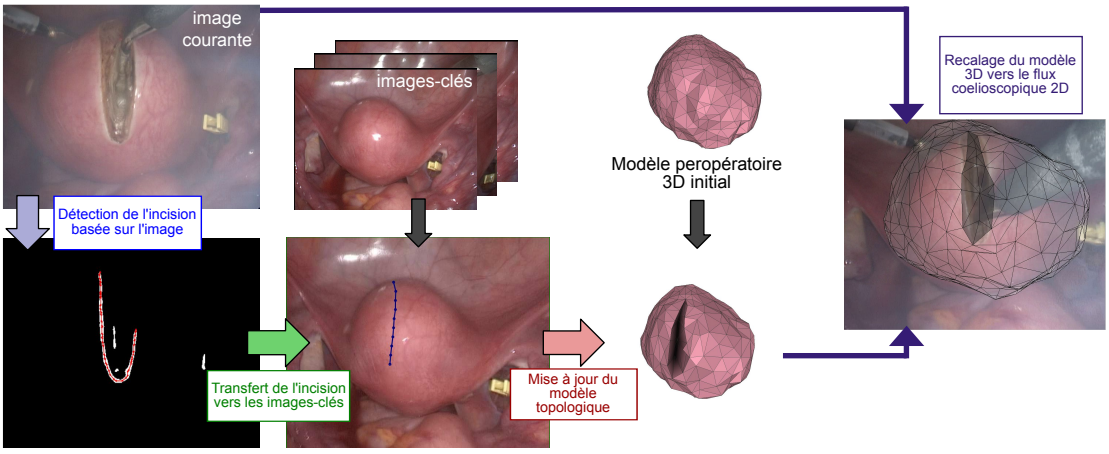


FIGURE 4.2 – Description du processus complet. (colonne 1) Détection de l'incision basée sur l'image. (colonne 2) Transfert de l'incision vers les images-clés. (colonne 3) Mise à jour topologique du modèle 3D. (colonne 4) Modèle 3D recalé reprojeté sur l'image courante.

L'ensemble du processus peut être résumé en 4 étapes. (1) Nous détectons la position de l'incision dans l'image courante. (2) La position de l'incision détectée est

transférée indépendamment vers plusieurs images-clés $\mathcal{T}_i$, où l'organe est toujours intact. Ces images-clés ont été utilisées dans la reconstruction SfM précédemment. Le modèle SfM fournit alors la transformation pour fusionner toutes les incisions transférées sur une image-clé de référence. (3) L'incision transférée est alors rétroprojetée sur le modèle 3D peropératoire grâce au modèle SfM et le modèle 3D est incisé virtuellement en adéquation avec l'incision détectée. (4) Le recalage est estimé entre le modèle 3D mis à jour et l'image courante. Dans ce processus, la sous-détection de l'incision est bien gérée et ainsi préférée à une sur-détection de l'incision. Le processus complet est illustré sur la figure 4.2.

4.3.2 Détection de l'incision basée sur l'image

Nous avons collecté et annoté un total de 181 images de 10 vidéos d'opérations de myomectomies. Toutes les participantes à cet jeu de données ont donné leur consentement éclairé en suivant l'approbation du CPP n 2018-A03130-55. Nous avons entraîné un réseau U-Net [101] pour prédire les berges visibles de l'incision. Cela donne une répartition très déséquilibrée des classes puisque les pixels annotés comme incision ne représentent pas plus de 0,05% des pixels de l'ensemble du jeu de données. Nous avons fait le choix d'annoter uniquement les berges. Un autre parti pris aurait été d'annoter la zone de l'incision directement. C'est une autre possibilité qui pourra être comparée par la suite.

Ce jeu de données contient des méthodes d'incision variées qui utilisent différents types d'outils. Le premier type d'outil sont les ciseaux standards. Les berges de l'incision faite par cet outil sont difficiles à percevoir si l'incision ne saigne pas. Le second type d'outil sont les bistouris électriques. Ces outils incisent grâce à une impulsion électrique ce qui crée une brûlure. Ces brûlures rendent les berges plus faciles à percevoir mais l'incision peut alors provoquer de la fumée.

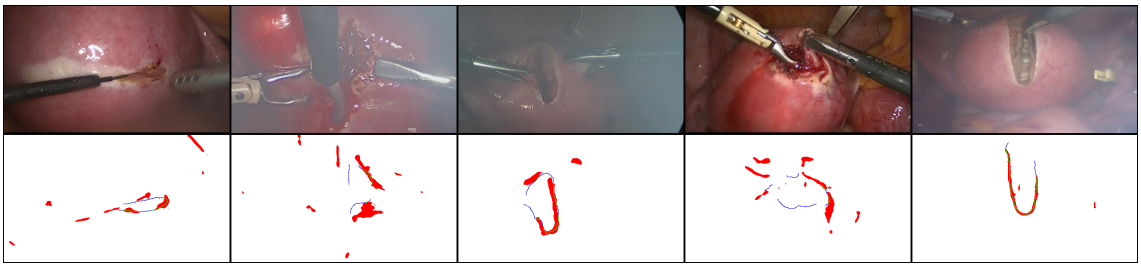


FIGURE 4.3 – Détection d'incision basée sur l'image sur des images de test. La vérité terrain est en bleu, la détection en rouge et leur superposition en vert.

Notre réseau produit une sortie $P(x, y)$ représentant la probabilité d'un pixel d'appartenir à la berge de l'incision aux coordonnées (x, y). L'incision détectée est obtenue en sélectionnant la composante connectée la plus grande, puis en l'affinant pour finalement la convertir en une ligne brisée. Ce post-traitement enlève énormément de pixels faux positifs, ce qui est bénéfique à la précision de la méthode.

4.3.3 Estimation de la transformation géométrique

4.3.3.1 Modèle de déformation de l'image

Notre but est d'estimer une transformation qui transfère l'incision détectée de l'image courante $\mathcal{I}$ vers une image-clé $\mathcal{T}$, sur laquelle l'organe est intact.

Soit $\mathbf{q}_i \in \mathbb{R}^2$ le vecteur de coordonnées d'un point dans l'image $\mathcal{I}$. La transformation $\mathcal{W} : \mathbb{R}^2 \times \mathbb{R}^{l\times 2} \mapsto \mathbb{R}^2$ déforme les points 2D de l'image courante $\mathcal{I}$ vers l'image-clé $\mathcal{T}$ et dépend d'un jeu de l points de contrôle $c_1, \ldots, c_l$ empilés dans la matrice de paramètres $\mathsf{L} \in \mathbb{R}^{l\times 2}$. La fonction de transformation paramétrique générale [9] est donnée par :

$$\mathcal{W}(\mathbf{q}_i, \mathsf{L}) = \mathsf{L}^\top \nu(\mathbf{q}_i), \tag{4.1}$$

avec $\nu : \mathbb{R}^2 \to \mathbb{R}^l$ une fonction non-linéaire qui multipliée avec L donne les valeurs de la transformation $\mathcal{W}$. La valeur de L est choisie de sorte à minimiser la fonction de coût :

$$\varepsilon(\mathsf{L}) = \varepsilon_d(\mathsf{L}) + \lambda_s \varepsilon_s(\mathsf{L}) + \lambda_f \varepsilon_f(\mathsf{L}), \tag{4.2}$$

composée d'un terme d'attache aux données ε_d qui est la distance moyenne entre les points $C_\mathbf{q}$ après la transformation et leur points correspondants $C_\mathbf{p}$, un terme de lissage ε_s qui contrôle la régularité du champ de mouvement, et un terme d'amincissement ε_f qui force la déformation à fermer la zone de l'incision. λ_s et λ_f représentent les poids relatifs au terme de lissage et au terme d'amincissement respectivement.

Les deux premiers termes, décrit dans [96], sont définis par :

$$\varepsilon_d(\mathsf{L}) = \frac{1}{n} \sum_{i=1}^{n} \|\mathcal{W}(\mathbf{q}_i, \mathsf{L}) - \mathbf{p}_i\|^2 \tag{4.3}$$

$$\varepsilon_s(\mathsf{L}) = \frac{1}{m_1} \|\mathsf{ZL}\|_\mathcal{F}^2, \tag{4.4}$$

où n est le nombre de correspondances et Z correspond aux dérivées secondes de la transformation empilées évaluées en m_1 points.

4.3.3.2 Comportement théorique du terme d'amincissement

Nos premières tentatives pour modéliser la fermeture sont basées sur les modèles d'auto-occultation de la littérature. Dans le cas des auto-occultations, les méthodes [96, 44] empêchent la transformation de se plier, en renforçant le terme de lissage [96] et en pénalisant les changements de signes des dérivées directionnelles de la transformation [44]. Des schémas théoriques de l'impact de ces méthodes sont illustrés dans le cas 1D sur la figure 4.4. En pratique, il est très probable que les environs de l'incision soient altérés visuellement, ce qui rend la mise en correspondance difficile, voire impossible. Nous avons également schématisé l'impact de ce manque de contraintes de données autour de l'incision. Sans autre contrainte, le terme de terme de lissage risque de favoriser une courbure faible et ainsi d'"ouvrir" l'incision. Le terme d'amincissement qui pénalise les changements de signe de la dérivée n'empêche aucunement cette situation. Un modèle de transformation conçu pour fermer l'incision doit, en plus, garantir que tous les points de la zone d'incision soient transférés sur la ligne d'incision. La transformation doit ainsi s'effondrer, c'est-à-dire que ses dérivées premières doivent s'annuler, dans la direction orthogonale à la direction de l'incision. La définition du terme d'amincissement est donc bien adaptée à

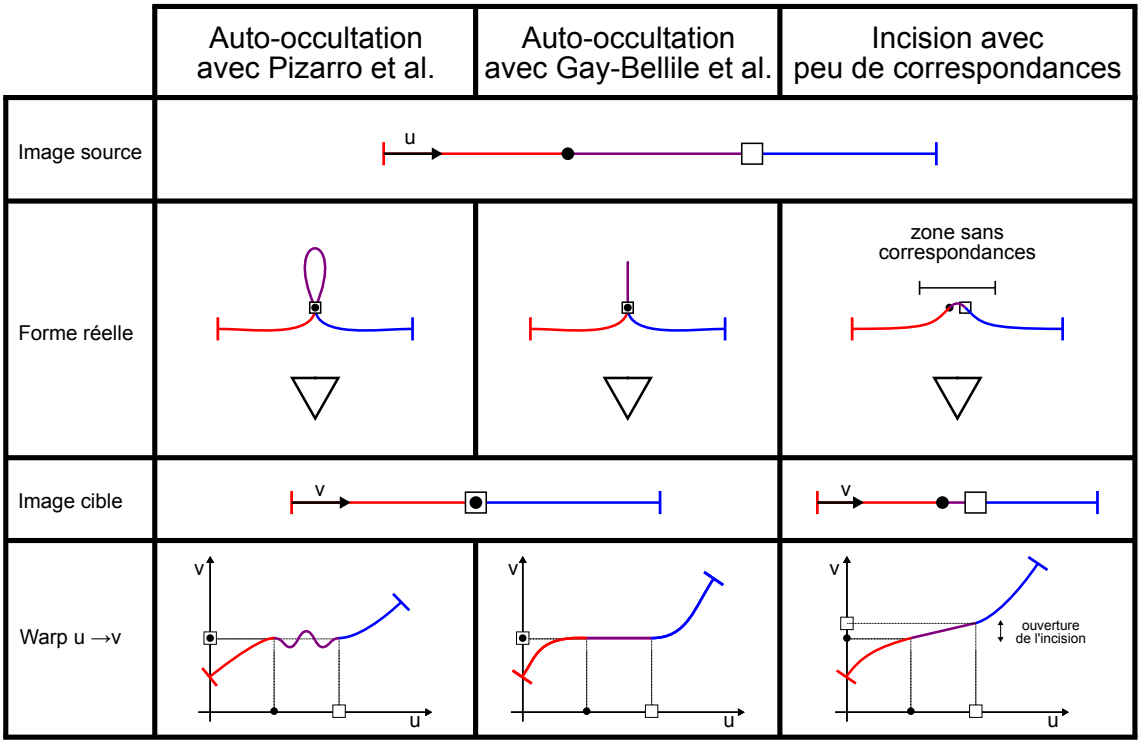


FIGURE 4.4 – Illustration théorique des auto-occultations et de l'incision dans le cas 1D. Dans l'image source, la portion violette représente la zone de singularité qui doit être fermée dans l'image cible. Avec l'approche de Pizarro et al. [96], le terme de lissage est renforcé sur la portion violette pour que la configuration la plus lisse soit favorisée pour la zone auto-occultée. Avec l'approche de Gay-Bellile et al. [44], le terme d'amincissement empêche la dérivée de la transformation de changer de signe. Cela implique que la transformation $u \rightarrow v$ soit constant dans la portion violette. Ces situations sont réalistes si le terme d'attache aux données donne des contraintes sur l'ensemble des portions rouge et bleue (cette hypothèse utilisée pour représenter les deux premières situations théoriques). Dans le cadre des incisions, il est peu probable que des correspondances soient trouvées autour de l'incision. Dans cette portion sans correspondance, le terme de lissage va minimiser la courbure et donc ouvrir l'incision. Les solutions apportées par les méthodes de Pizarro et al. [96] et Gay-Bellile et al. [44] ne sont pas suffisantes pour fermer l'incision dans la 3^{e} situation.

notre situation. En revanche, l'implémentation du terme d'amincissement utilisé par Gay-Bellile et al. [44] pour estimer la transformation n'est pas suffisante.

L'implémentation d'un terme d'amincissement qui impacte directement la norme des dérivées premières dans une direction va provoquer une singularité de premier ordre dans la zone d'incision. Dans cette zone, on force les dérivées à s'annuler dans une direction. En dehors de cette zone, les dérivées ne sont pas nulles. La transition entre les bords de l'incision et la zone de l'incision n'est donc pas lisse et va rentrer en conflit avec le terme de lissage qui s'applique partout de manière équivalente. Il pourrait être intéressant de désactiver le terme de lissage autour de la zone d'incision pour faciliter la fermeture de l'incision avec le terme d'amincissement.

Dans le schéma 1D, on ne perçoit pas le fait que la direction orthogonale à la fermeture de l'incision ne doit pas être impactée. Dans le cas contraire, on viendrait

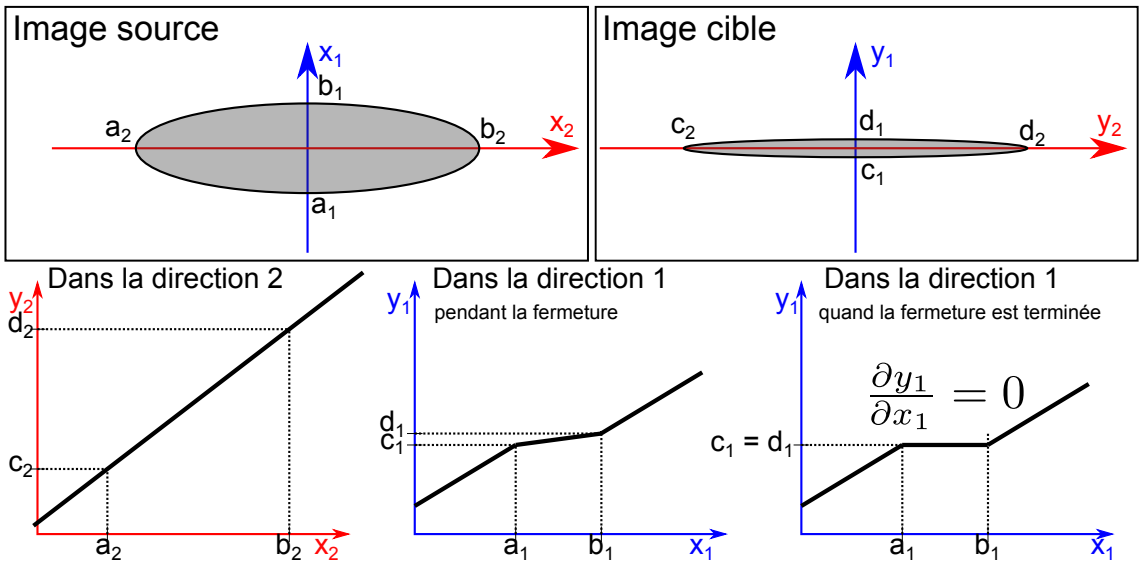


FIGURE 4.5 – Illustration du comportement théorique que l'on veut imposer à la transformation pour transférer l'incision. Dans l'espace d'entrée (haut à gauche), l'incision est schématisée par une ellipse qui croisent les points a_1, a_2, b_1, b_2. Dans l'espace de sortie, la zone de l'incision est représentée par une ellipse qui passent par les points c_1, c_2, d_1, d_2. On veut que les points d_1 et c_1 coïncident alors que c_2 et d_2 doivent être laissés tels quels. Dans la direction 2, le comportement de la transformation est uniquement impacté par la transformation de la surface entre les deux images (changement d'échelle, rotation, translation, et potentiellement des déformations non affines). La transformation de l'intervalle décrit par (x_2, y_2) dans $[a_2, b_2] \times [c_2, d_2]$ est cohérente avec la transformation du reste de l'image (pas de rupture de pente). Dans la direction 1, l'évolution de la transformation s'annule pour fermer l'espace entre c_1 et d_1. Quand la fermeture est obtenue, on a $c_1 = d_1$ (en bas à droite). En pratique dans le cas 2D, l'association (c_1, d_1) est inconnue. Pour un point du contour de l'incision, on sait juste qu'il existe un autre point qui partage la même image, sauf pour les extrémités de l'incision.

déformer la longueur de l'incision ce qui peut avoir un impact sur la mise à jour du modèle 3D virtuel. Cette contrainte est illustrée dans la figure 4.5.

4.3.3.3 Définition des termes d'amincissement candidats

Concrètement, nous avons comparé quatre termes différents. Les deux premiers sont empruntés des travaux de [44] où ils étaient utilisés pour gérer les auto-occultations. Nous avons proposé deux nouveaux termes d'amincissement.

Le premier terme utilisé par Gay-Bellile et al. [44] consiste à pénaliser directement les changements de signe des dérivées directionnelles dans toutes les directions. Ce terme est construit sur la fonction γ dont la formule est donnée par $\gamma(x) = x^2$ si $x < 0$ et 0 sinon. Cette fonction permet de pénaliser le produit des dérivées gauche et droite uniquement lorsqu'elles sont de signes opposés. Le terme d'amincissement est formulé ainsi :

$$\varepsilon_{f_1}(\mathsf{L}) = \frac{1}{m_2} \sum_{\mathbf{a}\in\mathcal{R}} \sum_{\mathbf{d}\in\mathcal{F}} \sum_{c\in\{x,y\}} \gamma(\mathbf{E}_l^c(\mathbf{d},\mathbf{a},\mathsf{L})\mathbf{E}_r^c(\mathbf{d},\mathbf{a},\mathsf{L})), \tag{4.5}$$

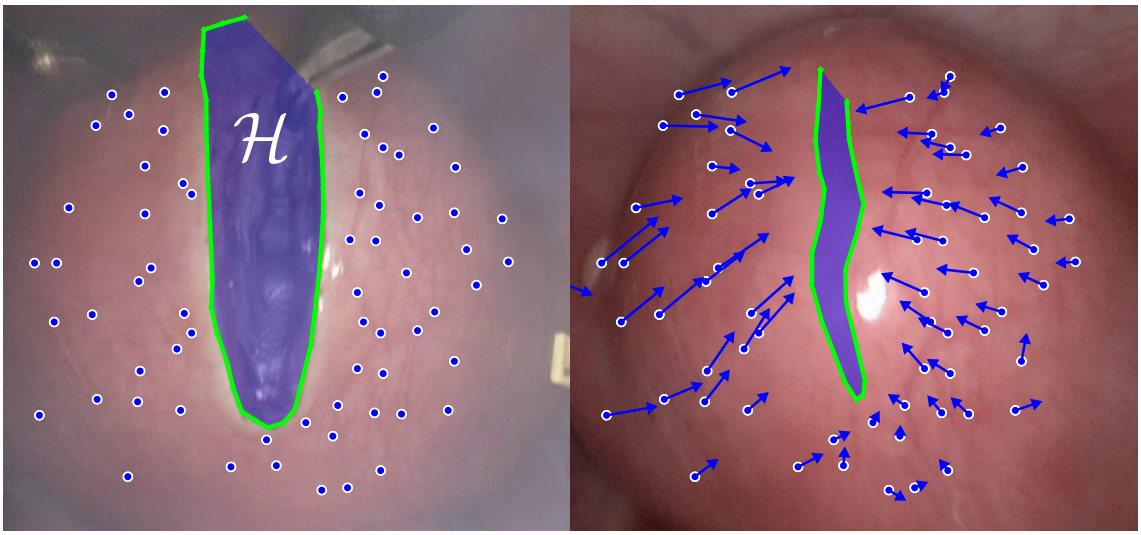


FIGURE 4.6 – Exemple pour illustrer le déplacement des pixels entre $\mathcal{I}$ (utérus incisé) et $\mathcal{T}$ (utérus intact). Gauche : Les points bleus représentent les points-clés utilisés dans le terme d'attache aux données de l'estimation de la transformation. Ces points ont été manuellement sélectionnés et certains extraits automatiquement en utilisant SIFT. La zone bleue représente la zone d'incision, notée $\mathcal{H}$. Droite : Déplacement de chaque point-clé (flèches bleues) de l'image $\mathcal{I}$ vers $\mathcal{T}$. Les bords de l'incision transférée (ligne brisée verte) sont affichés sur l'image $\mathcal{T}$. On constate que l'estimation de la transformation n'est pas parvenue à fermer complètement l'incision avec le terme d'amincissement ε_{f1}.

avec m_2 le nombre de points sur lesquels le terme d'amincissement est évalué, $\mathcal{F}$ un sous-ensemble fini du cercle unité $\mathbb{S}^1$, $\mathbf{E}_l$ et $\mathbf{E}_r$ les approximations en différence finie des dérivées directionnelles gauche et droite respectivement de $\mathcal{W}$. $\mathbf{E}_l^c(\mathbf{d},\mathbf{a},\mathsf{L}) = \frac{\mathcal{W}(\mathbf{a},\mathsf{L})-\mathcal{W}(\mathbf{a}-\epsilon\mathbf{d},\mathsf{L})}{\epsilon}$, avec ϵ petit.

Le second terme d'amincissement [44] est utilisé pour détecter explicitement la zone d'auto-occultation. Grâce à l'utilisation du terme d'amincissement ε_{f1}, la zone occultée correspond alors aux points pour lesquels il existe une direction dans laquelle la norme de la dérivée directionnelle est faible. Pour chaque point testé $\mathbf{a}$, ils calculent ainsi :

$$\sigma_0 = \min_{\mathbf{d}\in\mathbb{S}^1} \left\| \frac{\partial_{\mathbf{d}}\mathcal{W}(\mathbf{a},\mathsf{L})}{\partial \mathbf{a}} \right\|^2, \tag{4.6}$$

avec $\mathbb{S}^1$ le cercle unité, et $\frac{\partial_{\mathbf{d}}\cdot}{\partial \mathbf{a}}$ la dérivée directionnelle selon la direction d, évaluée en $\mathbf{a}$. Ce problème de minimisation a une solution convexe. On note J la matrice Jacobienne de $\mathcal{W}$ évaluée en $\mathbf{a}$:

$$\sigma_0 = \min_{\mathbf{d}\in\mathbb{S}^1} \mathbf{d}^\top \mathsf{J}^\top \mathsf{J}\mathbf{d}, \tag{4.7}$$

σ_0 correspond à la plus petite valeur propre de la matrice carrée symétrique $\mathsf{J}^\top\mathsf{J}$. Finalement, le terme d'amincissement correspondant ε_{f_2} est défini par :

$$\varepsilon_{f_2}(\mathsf{L}) = \frac{1}{m_2} \sum_{\mathbf{a}\in\mathcal{H}} \|\sigma_0(\mathbf{a},\mathsf{L})\|^2. \tag{4.8}$$

En plus de ces deux premiers termes, nous proposons deux termes d'amincissement. Ces termes pénalisent directement la norme de la dérivée directionnelle dans

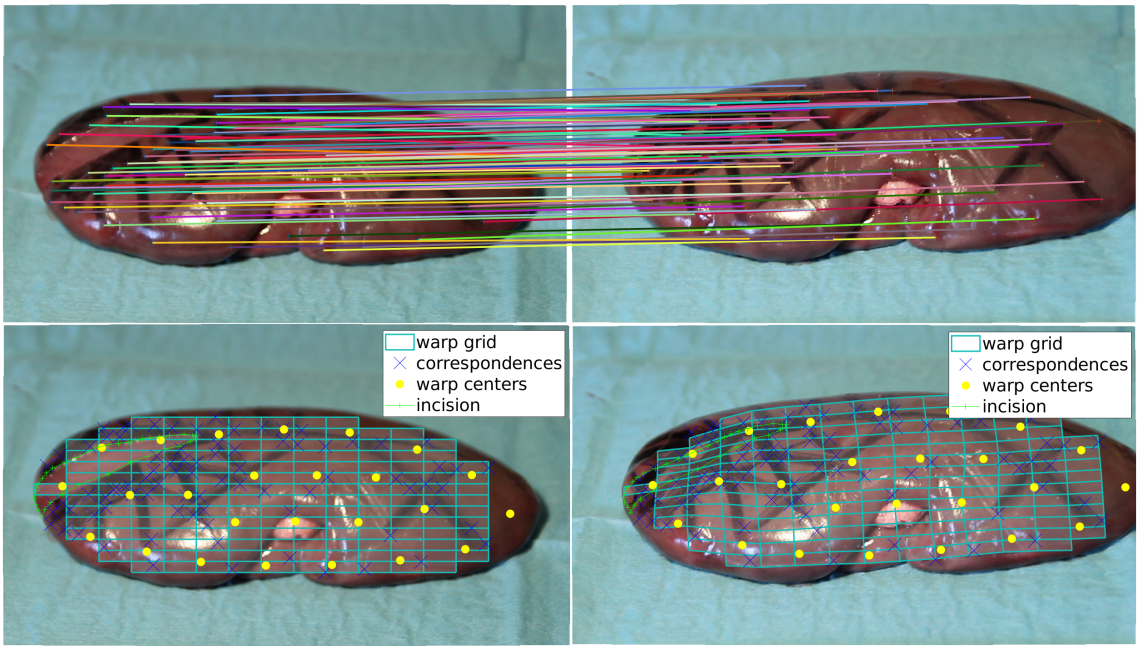


FIGURE 4.7 – Ligne du haut : Exemple de la mise en correspondance entre l'image $\mathcal{I}$ (gauche) et l'image $\mathcal{T}$ (droite) pour K5 utilisé dans nos expériences ex-vivos. Ligne du bas : une grille sur l'image $\mathcal{I}$, affichée en bleu transparent, est transférée sur l'image $\mathcal{T}$ grâce à l'application de la transformation estimée.

une direction $\mathbf{d}$. La direction $\mathbf{d}$ doit correspondre au côté court de l'incision. Nous proposons ainsi deux manières de déterminer cette direction. La première sélectionne la direction $\mathbf{d}_3$ qui minimise la norme de la dérivée directionnelle pour chaque point $\mathbf{a}$ dans la zone d'incision. La seconde manière consiste à prendre la direction $\mathbf{d}_4$ qui minimise les dérivées directionnelles sur l'ensemble des points de la zone d'incision :

$$\mathbf{d}_3(\mathbf{a}, \mathsf{L}) = \underset{\mathbf{d}\in\mathcal{F}}{\operatorname{argmin}}(\|\mathbf{E}_l(\mathbf{d}, \mathbf{a}, \mathsf{L})\|^2 + \|\mathbf{E}_r(\mathbf{d}, \mathbf{a}, \mathsf{L})\|^2), \tag{4.9}$$

$$\mathbf{d}_4(\mathsf{L}) = \underset{\mathbf{a}\in\mathcal{H}}{\operatorname{mode}}(\mathbf{d}_3(\mathbf{a}, \mathsf{L})). \tag{4.10}$$

Une fois les directions établies, on construit le terme d'amincissement en combinant les dérivées directionnelles pour ces directions. Les termes d'amincissement ε_{f_3} et ε_{f_4} sont finalement donnés par :

$$\varepsilon_{f_3}(\mathsf{L}) = \frac{1}{m_2}\sum_{\mathbf{a}\in\mathcal{H}} \|\mathbf{E}_l(\mathbf{d}_3(\mathbf{a}, \mathsf{L}), \mathbf{a}, \mathsf{L}) \cdot \mathbf{E}_r(\mathbf{d}_3(\mathbf{a}, \mathsf{L}), \mathbf{a}, \mathsf{L})\|^2 \tag{4.11}$$

$$\varepsilon_{f_4}(\mathsf{L}) = \frac{1}{m_2}\sum_{\mathbf{a}\in\mathcal{H}} \|\mathbf{E}_l(\mathbf{d}_4(\mathsf{L}), \mathbf{a}, \mathsf{L}) \cdot \mathbf{E}_r(\mathbf{d}_4(\mathsf{L}), \mathbf{a}, \mathsf{L})\|^2. \tag{4.12}$$

4.3.4 Transfert d'incision de l'image vers le modèle 3D

4.3.4.1 Initialisation

Le minimum de la fonction de coût globale (4.2) privée du terme d'amincissement a une solution convexe [9] donnée par :

$$\mathsf{L} = (\mathsf{A}^\top\mathsf{A} + n\mu^2\mathsf{Z}^\top\mathsf{Z})^{-1}\mathsf{A}^\top\Phi = \mathsf{T}\Phi, \tag{4.13}$$

avec $\mathsf{A}^\top = (\nu(\mathbf{q}_1) \cdots \nu(\mathbf{q}_n))$ et $\Phi^\top = (\mathbf{p}_1 \cdots \mathbf{p}_n)$. Avant de minimiser ε, nous utilisons la solution proposée par Pizarro et al. [96] pour retirer les outliers. Ils proposent de retirer ces fausses correspondances du jeu de correspondances initial $\{C_\mathbf{q}, C_\mathbf{p}\}$ en supposant que la surface considérée reste lisse localement. Pour chacun de ces points candidats, ils vérifient ainsi si la transformation du voisinage de ce candidat permet d'expliquer la transformation du point candidat.

4.3.4.2 Fermeture de l'incision

Les berges de l'incision transférées depuis $\mathcal{I}$ vers $\mathcal{T}$, où l'organe est intact, doivent se superposer pour former une courbe (voir la ligne brisée verte sur la figure 4.6) qui correspond à la zone où la topologie de l'organe doit être mise à jour. La transformation estimée doit fermer la zone de l'incision, notée $\mathcal{H}$, décrite par la zone comprise entre les berges de l'incision. Nous utilisons Levenberg-Marquardt pour minimiser la fonction de coût globale définit dans l'équation (4.2). Les paramètres optimisés de la transformation L_f qui minimisent l'équation (4.2) sont appliqués sur les points qui décrivent l'incision détectée $\mathbf{Q}_c$. Les points transférés de l'incision sont donnés par $\mathbf{Q}_f = \mathcal{W}(\mathbf{Q}_c, \mathsf{L}_f)$.

4.3.4.3 Consensus des image-clés

La méthode de transfert de l'image vers une image-clé peut être estimée pour chaque image-clé $\mathcal{T}_i$, à partir du moment où suffisamment de correspondances sont établies entre les deux images. SfM fournit la pose relative de chaque image-clé et la structure dense 3D de l'organe, ce qui nous permet de transférer n'importe quel point de l'organe entre les image-clés. Pour chaque transformation estimée $\mathcal{W}_i$ du jeu de correspondances $\{C_\mathbf{q}, C_\mathbf{p}\}_i$ entre $\mathcal{I}$ et $\mathcal{T}_i$, nous pouvons ainsi transférer les berges de l'incision transformées dans une image-clé de référence $\mathcal{T}_r$. L'image-clé de référence est sélectionnée comme étant l'image-clé qui partage le plus de correspondances avec l'image courante. On note $\mathbf{Q}_i = \mathcal{W}_i(\mathbf{Q}_c, \mathsf{L}_f)$ la liste des points de l'incision transformée vers l'image $\mathcal{T}_i$. La liste des point de l'incision transformée vers l'image $\mathcal{T}_i$ puis transférée vers l'image de référence $\mathcal{T}_r$ est donnée par :

$$\mathbf{Q}_i^r = \mathcal{P}(\mathcal{C}_i, \mathcal{C}_r, \mathbf{Q_i}), \tag{4.14}$$

avec $\mathcal{P}$ la fonction de transfert de l'image associée à la caméra $\mathcal{C}_i$ vers la caméra de référence $\mathcal{C}_r$.

Toutes les incisions transférées doivent s'aligner dans l'image de référence. Nous pouvons obtenir de manière robuste l'incision transférée en estimant la médiane des incisions transférées. Une illustration du résultat est présentée dans la figure 4.8. Le j^e point de l'incision transférée avec consensus $\mathbf{Q}_{l,j}$ est obtenu en estimant la médiane des points à la j^e position des k transferts d'incision, donné par la formule :

$$\mathbf{Q}_{l,j} = \underset{\mathbf{a}\in\mathbb{R}^2}{\operatorname{argmin}} \sum_{i\in\{1,\cdots,k\}} \|\mathbf{Q}_{i,j}^r - \mathbf{a}\|, \tag{4.15}$$

$\mathbf{Q}_{i,j}^r$ est le point positionné à la j^e position dans la liste des points $\mathbf{Q}_i^r$.

L'incision transférée finale $\mathbf{Q}_f$ est obtenue en appliquant un amincissement de l'incision transférée $\mathbf{Q}_l$, grâce à des opérations morphologiques.

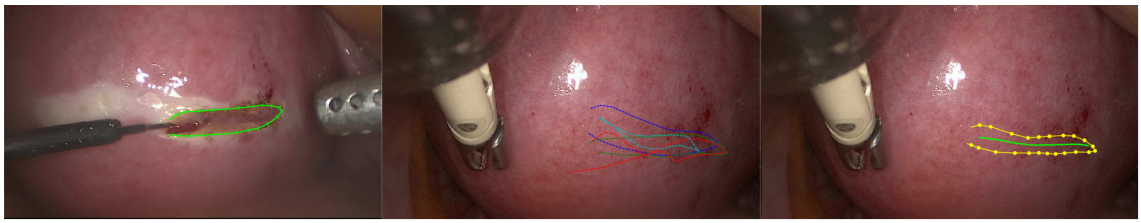


FIGURE 4.8 – Transfert d'incision vers l'image-clé de référence. Gauche : Image courante avec les bords de l'incision détectée. Centre : Les différents transferts d'incision sont reprojetés dans l'image de référence (chaque couleur représente un transfert différent). Droite : Médiane des incisions transférées $\mathbf{Q}_l$ (jaune) et incision transférée après amincissement $\mathbf{Q}_f$ (vert).

4.3.5 Mise à jour topologique du modèle 3D

L'implémentation de notre solution est simple et fiable. A chaque fois que la longueur de l'incision transférée finale $\mathbf{Q}_f$ augmente, nous mettons à jour le modèle 3D. Pour cela, nous créons un modèle 3D qui représente l'incision sur le modèle 3D intact. Au lancement du processus, nous avons défini une longueur initiale non nulle $d_0 > 0$ pour éviter des mises à jour suite à des réponses fallacieuses du détecteur d'incision.

Une fois l'incision rétro-projetée sur le modèle 3D, nous créons un modèle 3D qui représente le passage de l'outil sur le modèle 3D intact. Le fond du modèle 3D de l'incision est obtenue en translatant l'incision d'une profondeur prédéfinie et dans la direction de la normale de la surface. La profondeur peut être définie en utilisant le modèle 3D préopératoire, par exemple en estimant la distance entre la tumeur et la surface de l'organe. La direction de l'incision peut également être modifiée pour correspondre à la direction de l'outil. La seule contrainte est que ces paramètres sont prédéfinis dans la version actuelle.

On définit la largeur suffisamment petite pour limiter l'extraction du volume mais suffisamment grande pour que l'étape de recalage puisse considérer les deux berges comme distinctes (voir les détails dans la section 4.3.6).

Finalement, le modèle 3D incisé $\mathbf{S}_t$ est obtenu en appliquant une différence booléenne entre le modèle 3D intact et le modèle 3D de l'incision créé. Le modèle 3D obtenu est composé des sommets du maillage du modèle 3D peropératoire intact et des sommets créés pour modéliser l'incision.

Cette méthode est simple et rapide mais peut générer des sommets de manière hétérogène au niveau de l'incision. Nous utilisons Blender [1] pour réaliser cette opération. Il est nécessaire que le modèle 3D peropératoire intact soit fermé et bien défini (*watertight* en anglais).

4.3.6 Recalage entre le modèle 3D et le flux vidéo cœlioscopique

Le recalage est inspiré du recalage initial développé dans [26]. Pour contrôler la déformation du modèle 3D incisé, nous avons créé un modèle 3D en éléments finis (FEM) tétraédriques. Les sommets des tétraèdres sont construits à partir des sommets d'une grille régulière (avec un espacement $\tau = 5$ mm). Nous contrôlons la déformation du modèle 3D en déformant la position des sommets de la grille de

sommets utilisée. Nous utilisons la fonction de transformation $f(\mathbf{p}, \mathbf{x}) : \Omega \rightarrow \mathbb{R}^3$ qui transforme un point 3D $\mathbf{p}$ depuis le domaine du modèle 3D Ω vers les coordonnées de l'image cœlioscopique courante, où $\mathbf{x}$ représente les paramètres de déformation de la grille 3D. Nous utilisons l'implémentation de l'énergie de déformation de Saint Venant-Kirchoff [26] pour estimer l'énergie interne du modèle FEM. Comme le modèle FEM contient le modèle 3D incisé, l'énergie interne nécessaire pour déformer le modèle FEM peut être virtuellement plus grande que celle nécessaire pour déformer le modèle 3D incisé. Pour limiter cette énergie supplémentaire qui rendrait le modèle plus raide que nécessaire, nous faisons une distinction entre les tétraèdres dont un des sommets est situé à l'extérieur du modèle 3D incisé. Pour ces tétraèdres extérieurs, nous limitons le module d'Young E_y et le coefficient de Poisson ν_P, par rapport aux valeurs utilisées dans les tétraèdres internes, puisque ces tétraèdres contrôlent partiellement la déformation de vide.

En fonction de l'espacement τ utilisé pour créer la grille 3D, le maillage tétraédrique FEM pourrait en réalité fermer l'incision en créant un lien direct entre les deux bords de l'incision. Pour s'assurer qu'un tétraèdre ne puisse pas couvrir les deux côtés de l'incision, nous choisissons une valeur d'espacement τ plus petite que la largeur de l'incision. Cela garantit uniquement que les bords de l'incision ne soient pas liés par un seul tétraèdre sur le haut de l'incision. Il est très probable qu'en se rapprochant du fond de l'incision sur le modèle 3D mis à jour, un seul tétraèdre lie les deux bords de l'incision.

Nous formulons le recalage sous la forme d'un problème de minimisation d'énergie, avec l'énergie venant d'un terme d'*a priori* et de termes d'attache aux données. Le terme d'a priori encode l'énergie interne du modèle 3D, qui est utilisée pour régulariser le problème. Nous utilisons deux termes d'attache aux données différents. Le premier évalue l'erreur moyenne de projection des correspondances du modèle 3D vers les correspondances $\mathbf{q}$ de l'image courante $\mathcal{I}$. Le second terme évalue la distance entre la reprojection de l'incision du modèle 3D avec l'incision détectée $\mathbf{Q}_c$. La fonction de coût d'énergie globale $E \in \mathbb{R}^+$ est donnée par :

$$E(\mathbf{x}) = E_c(\mathbf{x}, \mathbf{p}, \mathbf{q}) + \lambda_a E_a(\mathbf{x}, \mathbf{Q}_c) + \lambda_i E_i(\mathbf{x}), \tag{4.16}$$

avec λ_a, λ_i les poids relatifs pour contrôler la déformation non-rigide.

Nous optimisons E avec une méthode d'optimisation itérative non-linéaire Gauss-Newton en utilisant une stratégie *raide-vers-flexible* (*stiff-to-flexible* en anglais) proposée par [23]. Cette stratégie améliore la convergence en commençant l'optimisation avec un modèle plus raide, matérialisé par une valeur de λ_i plus élevée. A chaque nouvelle convergence de E, on réduit le poids λ_i de l'énergie interne pour réduire progressivement la raideur du modèle, pour finalement atteindre la valeur λ_i désirée.

4.4 Résultats expérimentaux

Acquérir une vérité terrain précise et valide est un enjeu majeur pour la RA chirurgicale. La détection d'incision basée sur l'image peut être évaluée efficacement sur un jeu de données de test d'images cliniques annotées. Le reste du processus est évalué sur des organes ex-vivos, pour lesquels la détection de l'incision est contrôlée. L'ensemble du processus est testé sur des vidéos enregistrées d'opérations réelles pour être évalué qualitativement.

4.4.1 Détection de l'incision sur l'image

Pour l'évaluation, nous avons appliqué une méthode de validation croisée 10-fold en utilisant les jeux d'images de 10 opérations différentes. Cela signifie que nous réalisons 10 permutations pour lesquelles un dossier d'images est considéré comme le jeu de tests, les 9 dossiers restants sont utilisés pour entraîner notre réseau U-Net. Les résultats après concaténation sur les images tests de toutes les permutations donnent 0,049, 0,356 et 0,084 pour la précision, le rappel et le score f1 respectivement.

On remarque que la qualité des prédictions diffère grandement entre les opérations (voir la figure 4.3). Les meilleurs résultats ont été obtenus en entraînant avec l'Entropie Croisée Pondérée (ECP), par rapport à la combinaison de ECP avec la fonction de coût de Tversky, souvent utilisée dans les cas de déséquilibre entre les classes. Nous avons également testé l'approche des pénalités décrites dans le chapitre 3 mais celles-ci n'améliorent pas les prédictions obtenues. Ces méthodes étaient conçues pour réduire l'épaisseur des fortes réponses pour la détection de contours occultants. On peut en déduire que la présence de ces faux positifs et les prédictions épaisses dans la détection de contours ne sont pas liées aux mêmes raisons. Une hypothèse plus vraisemblable est que la variabilité pour décrire une incision n'ait pas été couverte par les cas particuliers qui figurent dans notre jeu de données de taille limitée et ainsi notre réseau a encore des difficultés à identifier les incisions dans des situations nouvelles.

Nous avons initialisé notre réseau U-Net avec les poids pré-entraînés pour la détection de contours occultants de l'utérus réalisée dans le chapitre 3. Nous avons utilisé la descente de gradient stochastique pour l'entraînement, en mettant à jour le taux d'apprentissage avec une stratégie en escalier sur 80 époques, le taux étant réduit par 10 toutes les 10 époques. Le taux d'apprentissage initial est défini à 10^{-4} et le modèle ne montre plus d'amélioration significative après 50 époques. Les poids relatifs aux classes pour l'ECP sont réglés à 10 et 0,1 respectivement pour la classe d'incision et la classe de non-incision.

4.4.2 Transfert d'incision de l'image vers le modèle 3D

L'optimisation du transfert d'incision repose sur plusieurs paramètres. Les centres de la transformation sont définis par une grille de $l \times l$. Les centres sont initialisés dans le masque défini par la silhouette de l'organe dans l'image courante $\mathcal{I}$. Nous utilisons l'algorithme de Lloyd [74] pour assurer une distribution homogène des centres dans le masque. Nous avons testé avec les valeurs de $l \in \{3, 5, 7\}$.

Nous avons utilisé une grille de $nC \times nC$ centres pour évaluer les dérivées premières et secondes de la transformation utilisées dans le terme de lissage et le terme d'amincissement. Nous avons testé avec $nC \in \{10, 30, 50\}$. En reprenant les équations des termes d'amincissement présentées dans la section 4.3.3.3, nous avons $m_2 = nC \times nC$.

Le second jeu de paramètres est celui des poids relatifs à chaque terme dans la fonction de coût λ_s, λ_f pour contrôler l'impact du terme de lissage et du terme d'amincissement. Nous avons testé avec $(\lambda_s, \lambda_f) \in \{0, 450, 1350, 4500, 18000, 900000\}^2$.

Les dernières options testées sont les différents termes d'amincissement utilisés dans la fonction de coût décrits dans la section 4.3.3.3.

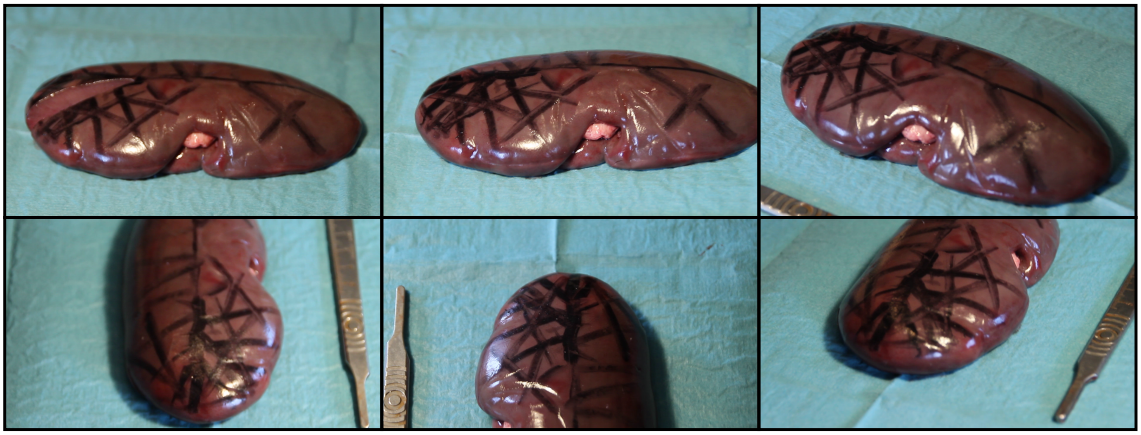


FIGURE 4.9 – Ensemble des images utilisées pour évaluer le transfert d'incision : L'image courante (en haut à gauche) qui présente l'organe incisé et déformé, et 5 vues différentes de l'organe intact utilisées pour reconstruire le modèle SfM. Le modèle 3D a été coloré avec un marqueur noir pour faciliter l'annotation de correspondances entre les deux états de l'organe.

Pour l'évaluation nous proposons de comparer en particulier 2 métriques. La première est le ratio R_D de l'aire d'incision $\mathcal{H}$ décrite par les points de l'incision $\mathbf{Q}_f$, avant l'amincissement morphologique, normalisée par la longueur D de l'incision annotée manuellement dans l'image de référence. Cela revient à représenter l'aire de l'incision comme un rectangle de longueur D et de largeur R_D. Cette métrique évalue à quel point la transformation $\mathcal{W}(\cdot, L_f)$ a fermé les berges de l'incision $\mathbf{Q}_f$ dans l'image de référence $\mathcal{T}$.

La seconde métrique est le score de contour S proposé dans le chapitre 3 pour comparer l'incision annotée sur l'image de référence, noté G avec l'incision transférée et amincie $\mathbf{Q}_f$. Le score de contour S est un score basé sur la distance tronquée pour comparer deux ensembles de pixels. Il utilise une distance seuil d_{max} pour classer les pixels en différentes régions. Dans ce chapitre nous utilisons $d_{max} = 20$px. Cette métrique évalue à quel point les pixels décrits par l'incision transférée sont proches de l'incision annotée qui sert de vérité terrain. Nous distinguons S_r et S_f qui correspondent respectivement à l'incision transférée avant amincissement et celle après amincissement.

Les deux métriques R_D et S_f évaluent des aspects différents du résultat. Le jeu de paramètres qui permet de fermer au mieux l'incision n'est peut-être pas celui qui permet de maximiser la précision de l'incision transférée finale. Nous avons décidé de donner la priorité au critère de la précision de l'incision transférée S_f après amincissement pour sélectionner le jeu de paramètres idéal. L'évaluation est réalisée sur le rein ex-vivo K5 (voir la figure 4.9). Comme la surface externe de l'organe a été marquée, l'annotation des correspondances et de l'incision est bien plus précise que sur les autres exemples pour lesquels nous n'avons pas marqué la surface. Nous avons ainsi marqué 85 correspondances entre une image courante et une image de référence pour cette évaluation.

Pour les sections 4.4.2.1 et 4.4.2.2, nous avons utilisé l'approche simple pour laquelle l'incision transférée est obtenue à partir d'une seule transformation $\mathcal{W}_i$ et donc d'un seul jeu de correspondances $\{C_{\mathbf{q}}, C_{\mathbf{p}}\}$. Pour la section 4.4.2.3, nous avons comparé cette approche simple à l'approche avec consensus.

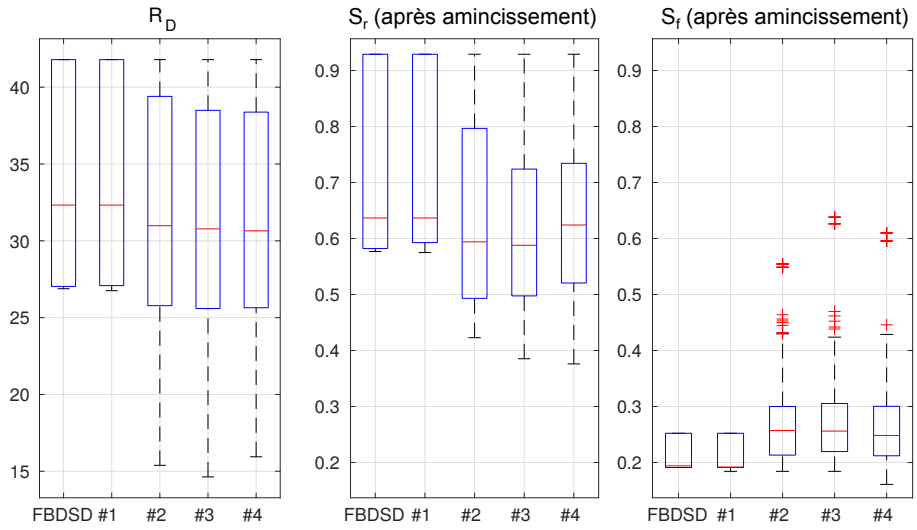


FIGURE 4.10 – Évaluation de l'impact des méthodes de termes d'amincissement sur les différentes métriques. Le diagramme en boîte à moustache décrit l'ensemble des résultats obtenus avec toutes les combinaisons de valeurs de chaque paramètre $[l, nC, \lambda_s, \lambda_f]$. L'axe des abscisses représente les différentes méthodes de terme d'amincissement ainsi que l'initialisation, notée FBDSD [96]. Pour chaque métrique présentée, on préférera l'option qui présente la valeur la plus basse.

4.4.2.1 Évaluation de l'impact du terme d'amincissement

Pour obtenir le meilleur jeu de paramètres, nous avons testé toutes les combinaisons possibles de paramètres décrites dans le paragraphe précédent. Les résultats de ces tests sont présentés sous la forme de diagrammes boîte à moustache où l'évolution de certains paramètres est mise en évidence (voir la figure 4.10). Nous montrons que les méthodes de terme d'amincissement réduisent globalement l'aire de l'incision transférée (R_D) et la précision de l'incision transférée (S_r). Par contre, pour la métrique (S_f), les termes d'amincissement $\varepsilon_{f2}, \varepsilon_{f3}$ et ε_{f4} obtiennent une précision moyenne plus faible que celle obtenue avec ε_{f1} et l'étape d'initialisation FBDSD. Cela signifie que l'amincissement morphologique de l'incision permet d'obtenir une bonne précision même sans terme d'amincissement. Le terme d'amincissement ε_{f4} permet toutefois d'obtenir un S_f plus bas pour au moins une configuration de paramètres.

Ce constat se confirme si on teste des valeurs de λ_f encore plus importantes. On observe alors une augmentation du score S_f alors que R_D diminue. Le jeu de paramètres retenu comme le meilleur compromis pour la précision de l'incision transférée est ainsi $l = 7, nC = 30, \lambda_s = 900000, \lambda_f = 18000$. On retiendra les termes d'amincissement ε_{f1} et ε_{f4} pour la suite.

On peut remarquer sur la figure 4.10 que ε_{f1} ne permet pas de fermer l'incision plus que l'étape d'initialisation. C'est un comportement que nous avions anticipé et décrit dans la section 4.3.3.2. La contrainte sur les variations des dérivées de la transformation n'est pas suffisante pour fermer l'incision efficacement.

4.4.2.2 Robustesse

Dans cette section, nous voulons évaluer les performances du transfert d'incision dans des conditions altérées. Les deux modifications réalistes que nous pouvons altérer sont la proportion de fausses correspondances et le nombre de correspondances utilisées. Dans des cas réels, on parvient à obtenir en moyenne 60 correspondances avec 85% de vraies correspondances grâce à la mise en correspondance de points SIFT.

Pour altérer la qualité des correspondances, nous sélectionnons de manière aléatoire une portion des correspondances et nous leur assignons des coordonnées aléatoires incluses dans la silhouette de l'organe. Nous appelons ces correspondances altérées des fausses correspondances. Pour limiter le nombre de correspondances, nous proposons 2 manières de sous-échantillonner les correspondances. La première consiste à retirer une portion des correspondances. Cette manière nécessite d'être réitéré plusieurs fois pour éviter les cas particuliers. Cette manière est appelée la manière *aléatoire*. La seconde manière réduit le nombre de correspondances de manière uniforme. Nous définissons une grille régulière sur le support des correspondances dans l'image courante. Pour chaque centre de cette grille, la correspondance la plus proche est conservée si elle est suffisamment proche. Celle-ci est appelée la manière *uniforme*.

Pour ces deux types d'altérations, nous avons testé comment réagissait le transfert d'incision lorsqu'on augmente progressivement l'altération : en augmentant la taille de la portion de correspondances altérées ou supprimées, et en réduisant le nombre de centres de la grille régulière pour l'approche uniforme.

Pour l'expérience avec des fausses correspondances, nous combinons le terme d'attache aux données ε_d avec différents M-estimateurs. Le nouveau coût associé au terme d'attache aux données est défini par :

$$\varepsilon_d(L) = \frac{1}{n}\sum_{i=1}^{N} \rho(\|\mathcal{W}(\mathbf{q}_i, L) - \mathbf{p}_i\|), \tag{4.17}$$

avec ρ une fonction M-estimateur. Nous avons comparé l'approche qui utilise le terme d'attache aux données originale (présenté dans l'équation (4.3)), que l'on nomme l2, avec les approches qui utilisent le coût associé à trois M-estimateurs différents : l1-l2, Hubert et Tukey. Nous avons utilisé $c = 30$ pour Tukey, et $k = 1,345$ pour Huber. Les trois fonctions M-estimateurs sont présentées dans la section 2.2.3.1. Pour cette expérience nous avons comparé uniquement le résultat avec les termes d'amincissement ε_1, ε_4 et FBDSD qui montraient les meilleurs résultats sur la première expérience.

Le M-estimateur Tukey semble être une bonne solution quand une grande partie des correspondances sont altérées, comme on peut le constater sur la figure 4.11. Nous constatons que l'erreur de la solution avec la norme l2 augmente rapidement avec l'augmentation du nombre de fausses correspondances. Encore une fois, l'option ε_{f1} montre une meilleure précision qu'avec le terme d'amincissement ε_{f4} et l'initialisation FBDSD.

Lorsqu'on diminue le nombre de correspondances, la transformation tend à fermer la zone de l'incision de plus en plus, mais on constate en parallèle que cela a un impact négatif sur la localisation de l'incision finale transférée (RMSR all, S_r et S_f sur la figure 4.12). L'approche uniforme a moins d'impact sur l'erreur de localisation que l'approche aléatoire. ε_{f1} est globalement moins impactée que les autres

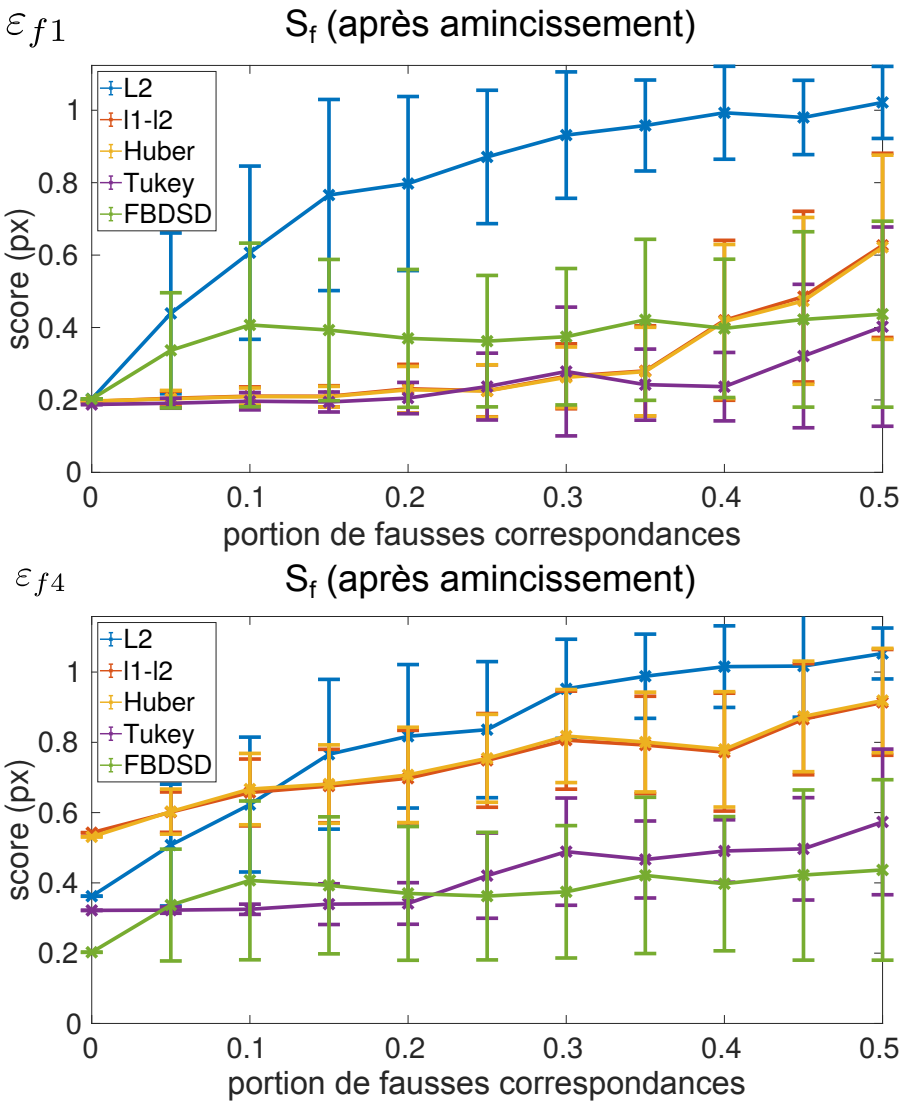


FIGURE 4.11 – Influence de la proportion de fausses correspondances sur le score de contour S_f après amincissement, testé avec le terme d'amincissement ε_{f1} et ε_{f4}. FBDSD représente l'étape d'initialisation. Nous avons généré 50 tirages de fausses correspondances pour chaque portion. Le nombre de correspondances total est de 85.

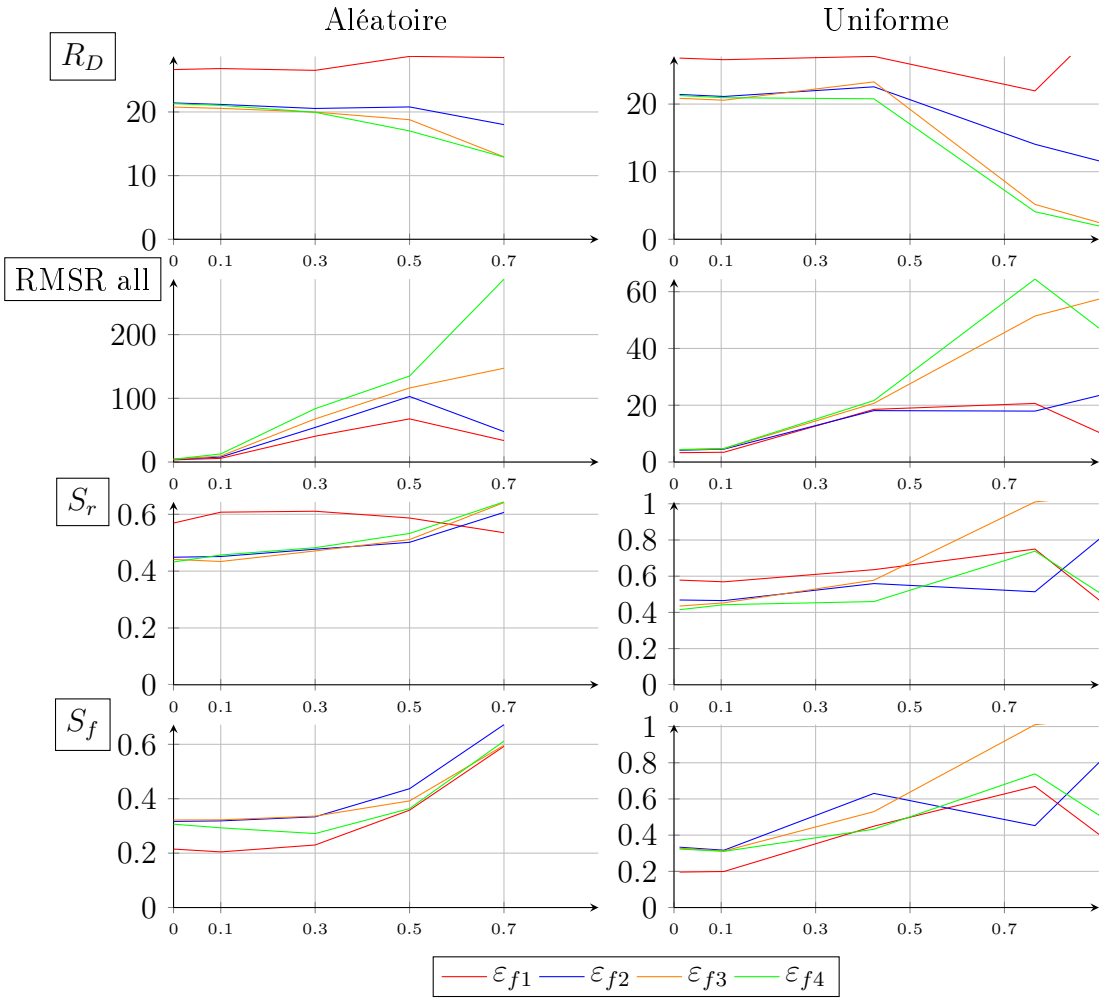


FIGURE 4.12 – Expériences de sous-échantillonnage des correspondances. Chaque colonne décrit une méthode différente. Chaque ligne représente une métrique différente. La métrique "RSMR all" représente la racine de l'erreur moyenne quadratique de reprojection des correspondances évaluées sur toutes les correspondances disponibles (on inclut les correspondances retirées dans l'opération de sous-échantillonnage). R_L et RSMR all sont exprimés en pixels et les scores S_r et S_f sont sans unité (normalisé par $d_{max} = 20$). L'axe des abscisses représente la proportion de correspondances éliminées. Le nombre initial de correspondances est de 85.

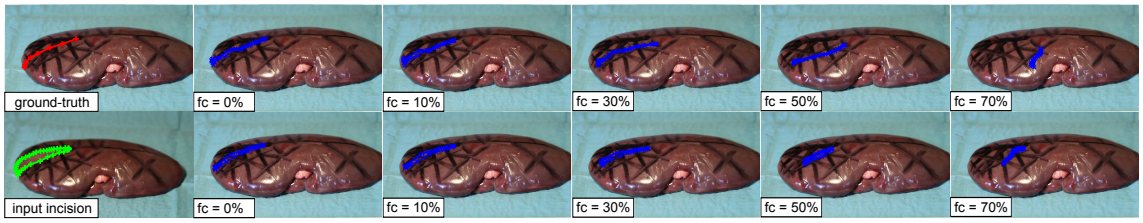


FIGURE 4.13 – Comparaison visuelle entre l'approche simple et l'approche consensus pour le transfert d'incision. (1ère image en haut à gauche) Vue de l'organe intact avec l'annotation de l'incision (rouge) (1ère image en bas à gauche) Image courante avec l'incision détectée (vert). Les images présentent alors l'évolution de l'impact du nombre de fausses correspondances sur l'approche simple (ligne du haut) et l'approche consensus (ligne du bas).

méthodes. Sur le graphique de R_D, on peut remarquer que ε_{f1} ne se comporte pas comme un terme d'amincissement.

Le terme d'attache aux données et le terme d'amincissement devraient tous les deux encourager la transformation à fermer la zone d'incision $\mathcal{H}$. Pourtant, on constate que cette expérience montre que lorsque le terme d'attache aux données est altéré, la transformation tend à fermer la zone $\mathcal{H}$ davantage. Cela peut suggérer que le poids relatif au terme d'amincissement n'est pas assez important. En observant à la fois R_L et S_f, on constate en fait que le terme d'amincissement tend bien à fermer l'incision davantage mais cela se fait au prix d'une perte de précision (S_f augmente). Augmenter le poids relatif du terme d'amincissement de manière disproportionnée réduit la précision du transfert de l'incision.

4.4.2.3 Méthode simple VS consensus

Nous avons également comparé la robustesse entre l'approche qui utilise un seul transfert et celle qui utilise le consensus de plusieurs transferts en capitalisant sur plusieurs images-clés. Pour l'approche avec consensus, nous avons utilisé 5 image-clés (voir la figure 4.9).

Le résultat visuel de cette comparaison est présenté dans la figure 4.13. On constate que l'approche avec consensus tend à raccourcir l'incision mais semble moins sensible à l'augmentation du nombre de fausses correspondances.

4.4.3 Recalage ex-Vivo

4.4.3.1 Description

Nous avons réalisé une étude en réalisant une incision sur 5 reins de porc ex-vivos. Nous avons reconstruit la scène 3D avant et après l'incision en utilisant SfM, ce qui nous a permis d'obtenir l'équivalent du modèle 3D peropératoire intact et la vérité terrain du modèle 3D incisé. Nous avons enregistré la vidéo pendant laquelle le rein est incisé avec un scalpel. Les reins de 1 à 3 et 5 sont incisés profondément, c'est-à-dire de part en part de l'organe et le rein 4 est incisé de manière plus superficielle (entre 6 et 10 mm). Pour le rein 5, nous avons utilisé un marqueur noir pour colorer la surface de l'organe ce qui nous a facilité le travail d'annotation des correspondances et la précision de celles-ci. Les dimensions moyennes des reins sont $140 \times 75 \times 25$ mm.

Exemple	Méthode (A)	GT→A (mm)			A→GT (mm)			Erreur de reprojection (px)	
		moy.	std	max	moy.	std	max	moy.	std
K1	ImageDet+	5,08	2,83	16,00	5,67	3,12	19,13	13,01	9,72
	ImageDet-	4,30	2,27	15,08	4,85	2,95	21,3	**12,80**	9,58
	ImageDet50+	5,41	2,87	16,88	5,60	2,68	10,22	13,86	9,30
	ImageDet50-	**4,05**	2,16	**14,97**	**3,99**	1,89	10,43	13,11	9,89
	Intact	4,18	1,88	18,32	4,46	2,43	**8,43**	12,86	9,5
K2	ImageDet+	2,87	2,08	**12,71**	3,06	2,24	14,43	10,91	6,59
	ImageDet-	**2,84**	2,20	12,87	2,98	2,32	14,41	10,53	6,20
	ImageDet50+	3,19	2,40	14,35	3,12	2,33	14,00	10,52	6,15
	ImageDet50-	2,95	2,28	13,06	**2,91**	2,24	**11,36**	**10,43**	6,20
	Intact	3,4	2,92	18,4	2,98	2,30	11,37	10,44	6,22
K3	ImageDet+	7,04	3,96	19,63	7,85	4,33	15,49	**17,75**	12,45
	ImageDet-	**6,37**	3,60	**18,82**	**7,01**	3,92	15,21	25,08	11,74
	ImageDet50+	6,48	3,77	18,98	7,06	4,05	14,84	20,11	1,73
	ImageDet50-	6,55	3,72	18,85	7,09	3,98	**14,29**	25,27	11,76
	Intact	7,33	4,02	19,1	7,62	4,09	14,48	20,36	13,2
K4	ImageDet+	3,38	1,78	7,22	3,67	2,16	**12,63**	9,45	7,55
	ImageDet-	2,98	1,73	7,06	3,41	2,45	14,44	7,75	6,79
	ImageDet50+	4,28	2,21	8,22	4,45	2,53	14,80	8,78	6,81
	ImageDet50-	2,98	1,75	7,10	**3,30**	2,32	15,19	7,72	6,76
	Intact	**2,91**	1,73	**7,00**	3,32	2,46	15,35	**7,67**	6,77
K5	ImageDet+	4,69	3,32	17,29	5,13	3,90	22,90	11,60	7,20
	ImageDet-	8,72	6,66	34,65	8,43	5,84	25,60	26,49	16,28
	ImageDet50+	**2,32**	1,83	**13,27**	**2,36**	2,11	**14,36**	**10,20**	6,33
	ImageDet50-	9,01	6,14	29,78	8,32	4,94	18,05	23,89	12,49
	Intact	10,34	5,79	24,86	10,38	6,41	31,34	24,21	16,28

TABLE 4.2 – Évaluation de l'erreur de recalage pour les 5 organes ex-vivos. "GT→A" traduit la distance du modèle 3D vérité terrain vers le modèle 3D recalé (l'erreur la plus basse pour chaque cas est affichée en gras).

Nos expériences sont similaires à celles de [94] mais avec moins de déplacement, ce qui est plus réaliste.

Pour le recalage, le poids de l'énergie interne λ_i est fixé à 10^{-3} (10^{-5} pour le rein 3 qui présente une déformation plus importante et qui donnait de mauvais résultats avec $\lambda_i = 10^{-3}$).

Le modèle 3D vérité terrain est obtenu grâce à une reconstruction SfM à partir de plusieurs images capturées après l'incision. Nous avons manuellement rogné la scène reconstruite pour se limiter à la surface externe visible de l'organe. Une limite possible de cette approche est que le modèle SfM ne peut reconstruire que la partie visible de l'incision. Si les berges de l'incision de l'organe se sont partiellement jointes ou lorsqu'elles sont très proches l'une de l'autre, la reconstruction peut anticiper la fin de l'incision. Pour obtenir un modèle 3D vérité terrain plus précis, il aurait fallu utiliser un scanner 3D ce qui n'était pas envisageable pour nous. La reconstruction SfM ne donne que la surface externe de la scène. Le modèle 3D intact $\mathcal{S}_0$ est obtenu en fermant cette surface de manière semi-automatique avec une combinaison de MeshLab [21] et Blender [1].

Nous voulons comparer le recalage qui utilise le modèle 3D initial $\mathcal{S}_0$, que l'on appelle *Intact* avec le recalage qui utilise le modèle 3D mis à jour $\mathbf{S}_t$, appelé *ImageDet*. Nous avons distingué plusieurs variations d'ImageDet. La première option, marquée "+", signifie que le terme d'alignement de l'incision a été utilisé pendant le recalage. On note l'option "-" quand $\lambda_a = 0$. Pour des raisons pratiques, les méthodes "+" ont été initialisées avec leur méthodes "-" respectives. La seconde option est notée ImageDet50% et traduit le fait que seule la moitié de l'incision transférée a été utilisée pour mettre à jour le modèle 3D et pour le recalage. Cette option permet de simuler l'impact d'une sous-détection de l'incision.

4.4.3.2 Détection de l'incision géométrique

Nous avons implémenté [94], que l'on appelle GeoDet. La différence principale avec la méthode originale est que nous utilisons une méthode de recalage différente. Nous n'avons réussi à faire fonctionner cette méthode que sur les deux premiers organes ex-vivos. Dans les cas qui n'ont pas permis de fonctionner, la méthode GeoDet détecte soit beaucoup de faux points de coupures ou ne détecte pas d'incision du tout. Nous avons utilisé les valeurs de paramètres $r_{P_F^0} = 30$ mm, $\tau = 4,5$, $n = 7$ pour GeoDet.

Cette méthode est itérative. Pour chaque image de la vidéo, la méthode GeoDet ne peut détecter qu'un seul point d'incision à la fois pour incrémenter la ligne d'incision. Cette approche a deux inconvénients majeurs. Premièrement, si une large portion d'incision est détectée, le point d'incision est estimé au barycentre des déformations causées par cette large incision, ce qui est en pratique assez imprécis. Deuxièmement, une fois un point d'incision ajouté, il ne peut plus être retiré. L'incision finale estimée peut ainsi contenir des points aberrants. L'apparition de ces points aberrants peut alors perturber davantage l'étape de recalage qui va provoquer l'apparition en chaîne de nouveaux points aberrants.

Dans les deux cas qui ont fonctionné, on constate que l'incision virtuelle est visuellement très loin de ce qui est attendu (voir la figure 4.14).

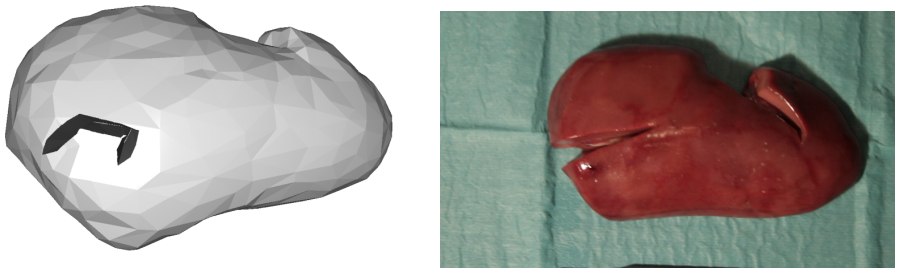


FIGURE 4.14 – (Gauche) Modèle 3D incisé obtenu avec la méthode GeoDet sur l'exemple K1. L'incision est en noir. (Droite) Prise de vue de l'organe correspondant. On constate que l'incision détectée ne correspond pas vraiment à la position réelle.

4.4.3.3 Évaluation

Dans cette étude, nous avons comparé les méthodes Intact, ImageDet+, ImageDet-, ImageDet50+, et ImageDet50-. Dans le tableau 4.2, nous avons comparé d'une part la reprojection moyenne des correspondances entre le modèle 3D recalé et l'image courante. D'autre part, nous avons évalué la distance échantillonnée du modèle 3D recalé vers le modèle 3D vérité terrain et vice-versa.

Un exemple des résultats du recalage pour l'exemple K1 est présenté sur la figure 4.15.

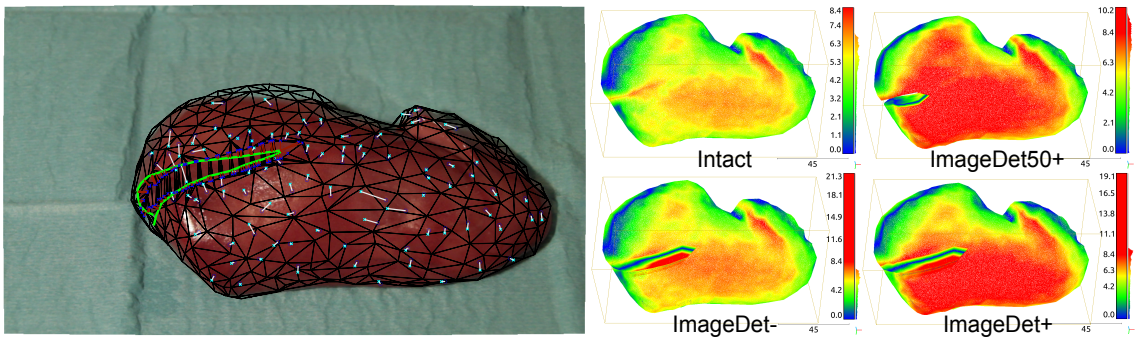


FIGURE 4.15 – Résultats visuels pour le recalage du rein K1. Gauche : Erreur de reprojection entre les correspondances (lignes blanches) pour ImageDet+. L'incision est marquée en vert, les erreurs d'alignement de l'incision sont marquées en rouge. La surface du maillage 3D visible est en noir. Droite : Distance échantillonnée du modèle 3D recalé vers le modèle 3D vérité terrain avec les méthodes Intact, ImageDet50+, ImageDet-, ImageDet+ (le rouge traduit les fortes valeurs et le bleu les faibles valeurs).

Pour les 5 cas ex-vivos, nous avons trouvé que $\lambda_i = 10^{-3}$ était un bon premier candidat et un meilleur résultat a uniquement été trouvé pour K3 avec $\lambda_i = 10^{-5}$.

Pour plusieurs cas (K1,K5), nous constatons que la méthode ImageDet50 donne de meilleurs résultats que la méthode ImageDet qui utilise toute l'incision transférée. Nous pensons que ce constat vient de l'acquisition de la vérité terrain qui tend à anticiper la fin de l'incision. Le modèle 3D vérité terrain présente ainsi une incision plus courte que la réalité.

Nous avons également constaté que l'ajout du terme d'alignement dans le recalage (méthodes notées "+") n'obtient pas nécessairement de meilleurs résultats. Il

est possible que la combinaison des deux termes d'attache aux données ne profite pas au recalage de la manière désirée.

4.4.4 Application du processus complet sur des expériences in-vivo

Nous avons appliqué l'ensemble des étapes décrites sur différents jeux d'images qui ont permis de constituer le jeu de données pour la détection d'incision. Pour ces exemples, des images cœlioscopiques ont été extraites avant et pendant l'incision. Pour chaque cas, nous avons utilisé le modèle entraîné sur les autres exemples pour la détection de l'incision.

Les images cœlioscopiques disponibles avant l'incision décrivent toutes un point de vue très similaire. La reconstruction SfM à partir de ces images donne alors un modèle 3D très plat. Cela est dû au fait que les prises de vue ont une pose relative très proche. Pour ces jeux de données, nous ne disposons pas du modèle 3D préopératoire contenant la forme complète de l'organe et des structures internes de celui-ci. Pour simuler l'étape de recalage initial qui permet d'obtenir un modèle 3D peropératoire fermé (*watertight*) nécessaire au reste du processus, nous avons manuellement déformé et aligné un modèle 3D de sphère sur le modèle 3D reconstruit par SfM.

Le résultat qualitatif du processus complet sur des données patients enregistrées est présenté dans les figures 4.16 et 4.17. Comme le modèle 3D recalé est un modèle 3D factice, celui-ci ne s'aligne pas parfaitement sur l'image courante utilisée. Nous avons donc décidé de ne montrer que la partie incisée du modèle 3D recalé sur la figure 4.17.

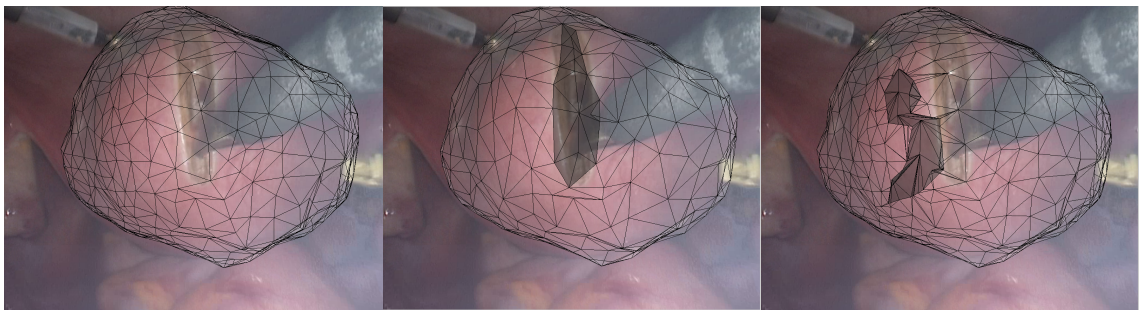


FIGURE 4.16 – Comparaison visuelle des méthodes sur des données patient : de gauche à droite, Intact, ImageDet, et GeoDet [94]. La zone de l'incision sur le maillage est affichée en noir.

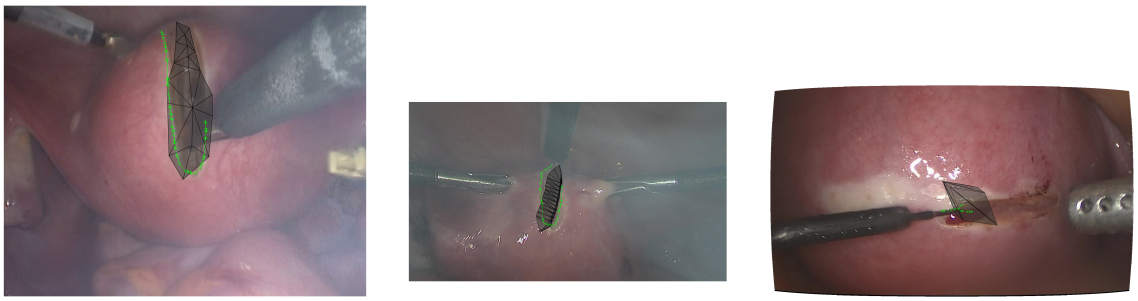


FIGURE 4.17 – Résultats qualitatifs sur les données patients qui présentent la zone de l'incision du maillage (noir) et l'incision détectée (vert).

4.5 Discussion et Conclusion

4.5.1 Détection de l'incision

Notre jeu de données est encore très limité. Nous espérons une amélioration significative des performances en augmentant sa taille. Avec un jeu de données plus conséquent, il pourrait être intéressant de tester une architecture de réseau plus conséquente que celle de U-Net, notamment des architectures qui ont montré des résultats sur une répartition de classes aussi hétérogène comme les réseaux dédiés à la détection de bords comme CASENet [125].

4.5.2 Transfert d'incision

Une des limites de la transformation TPS est qu'elle est globale. Une forte déformation très localisée comme celle due à la fermeture de l'incision peut ainsi impacter une partie beaucoup plus large de la transformation. Pour limiter que la déformation ne s'étende trop largement au reste de la transformation nous avons augmenté le poids du terme de lissage pour lisser la transformation ce qui limite indirectement la déformation impliquée par le terme d'amincissement. Il pourrait être pertinent de tester un type de transformation plus local comme le Bicubic B-spline (BBS).

Nous avons constaté que l'approche avec consensus présente un comportement robuste vis à vis des fausses correspondances (voir la figure 4.13). L'utilisation de cette approche pourrait être bénéfique dans des conditions réelles pour gérer la présence de mauvaises correspondances. Actuellement, dans le processus du logiciel Uteraug présenté dans [26], la détection de points-clés et leur mise en correspondance avec les images-clés disponibles est déjà réalisées en temps réel. Ajouter le calcul du transfert d'incision pourrait alors être facilité.

4.5.3 Recalage

Notre méthode de recalage est particulièrement sensible au poids relatif à l'énergie interne λ_i. Si ce terme est trop grand, le modèle 3D est trop rigide et le recalage ne parvient pas à déformer celui-ci pour aligner les correspondances. Si celui-ci est trop petit, le recalage peut déformer le modèle 3D trop facilement et parvient à obtenir une reprojection parfaite des correspondances en altérant la forme globale du modèle 3D. Cette dernière situation peut ainsi donner un modèle 3D quasiment

aplati dans les cas extrêmes, ce qui n'est pas perceptible sur la reprojection du modèle 3D recalé. Le modèle 3D recalé final peut également s'ajuster très bien une fois projeté sur l'image et pourtant être décalé dans l'espace dans la direction de la caméra. C'est un problème récurrent pour le recalage 3D vers une image 2D.

On observe que la déformation du modèle 3D recalé est souvent plus faible que celle observée en réalité. Cela indiquerait que notre méthode de recalage n'est pas aussi flexible que voulue. Avec une déformation disponible limitée, le recalage ne peut alors satisfaire deux termes d'attache aux données. Le terme d'alignement de l'incision pourrait être particulièrement bénéfique lorsque peu de correspondances ont été trouvées dans des cas réels.

Nous présentons dans la figure 4.18 l'ensemble des distances échantillonnées entre le modèle 3D vérité terrain, noté GT (*groundtruth* en anglais), et le modèle 3D recalé en fonction de la méthode utilisée pour l'exemple K3. Dans cet exemple, le recalage a des difficultés à correspondre avec la forte déformation observée. En particulier, même avec le terme d'alignement de l'incision ("+"), les deux parties de l'organe restent proches l'une de l'autre. L'utilisation du modèle FEM pour contrôler la déformation pourrait rendre le modèle plus rigide qu'en utilisant un modèle tétraédrique directement basé sur le modèle 3D incisé. Cette implémentation a été prévue initialement [26] pour le recalage initial entre le modèle 3D préopératoire et la reconstruction 3D de la surface visible pendant l'opération. Ce recalage nécessite des déformations beaucoup plus lisses distribuées sur l'ensemble du modèle 3D préopératoire. Une autre limite est que les correspondances sont moins faciles à repérer proche de l'incision. Or, c'est dans cette zone que la déformation est la plus importante. Cette dernière hypothèse n'est toutefois pas suffisante sinon l'ajout du terme d'alignement de l'incision aurait montré une amélioration significative.

4.5.4 Processus complet

Nous avons proposé le premier processus qui détecte les incisions depuis le flux cœlioscopique et utilise cette information pour mettre à jour le modèle 3D virtuel pour le recalage.

Le but de ce travail était de présenter un premier processus conceptuel pour gérer la mise à jour topologique du modèle 3D virtuel pendant le guidage de la réalité augmentée. Nous sommes bien conscients que permettre à ce processus de fonctionner en temps réel est un argument clé pour l'acceptabilité de cette méthode. Pour le moment, le recalage peut prendre jusqu'à 15 minutes pour mettre à jour le modèle 3D sur une image avec une grosse déformation (par exemple sur K3). Le reste du processus est quant à lui déjà bien plus rapide et peut s'exécuter en une dizaine de secondes. Ce temps d'exécution pourrait être réduit en implémentant cette solution en C++ plutôt qu'en Matlab.

4.5.5 Perspectives

Nous avions initialement la motivation de combiner l'approche géométrique de Paulus et al. [94] à notre détection d'incision basée sur l'image. En effet, en théorie, ces deux approches sont parfaitement compatibles puisqu'elles sont utilisées à des moments différents du processus : l'approche basée image en amont du recalage et l'approche géométrique une fois le recalage obtenu. L'approche géométrique peut

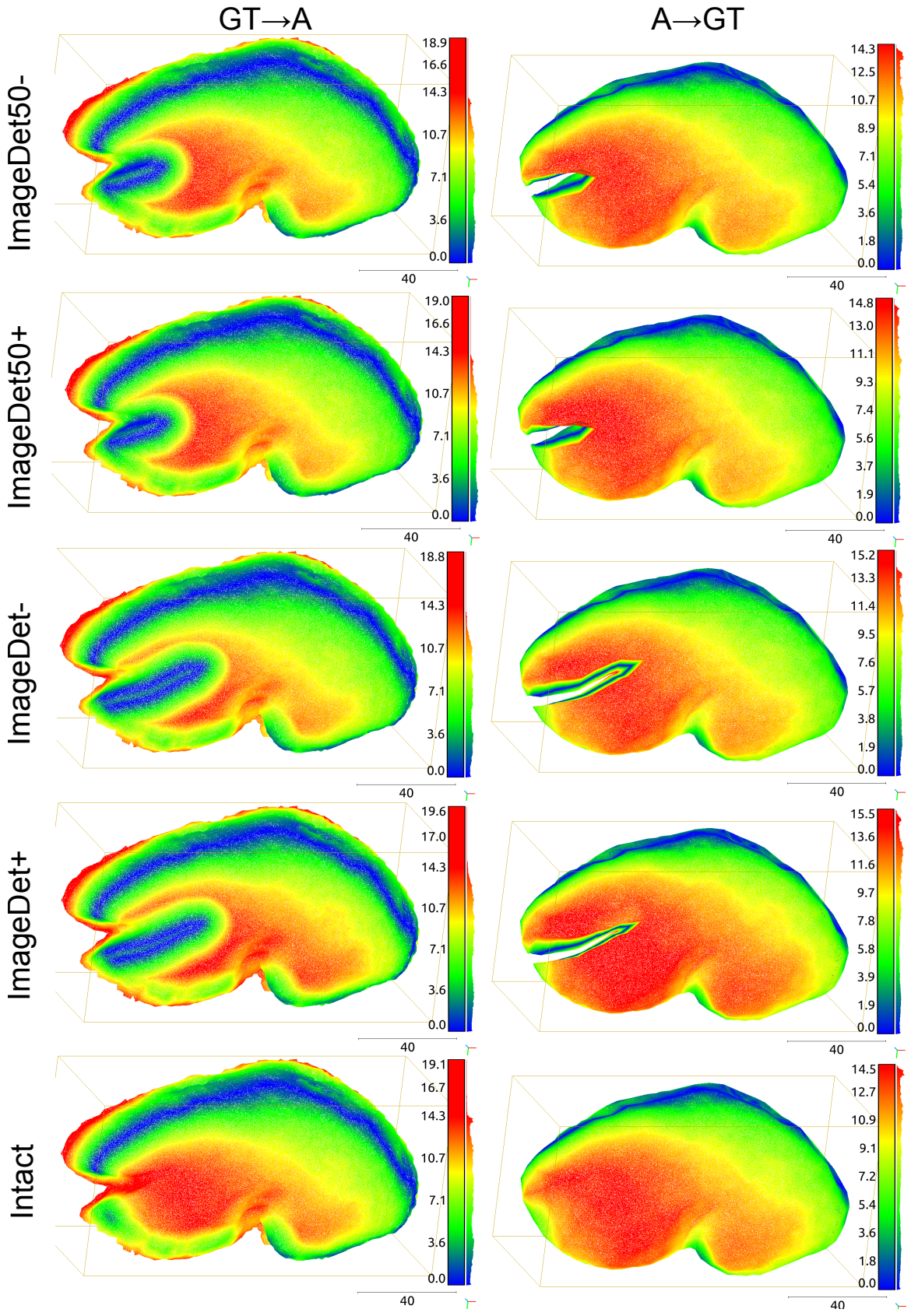


FIGURE 4.18 – Distances 3D échantillonnées pour l'exemple K3. L'échelle de couleur est partagée (le bleu représente les valeurs faibles et le rouge les valeurs fortes).

ainsi repérer une incision qui aurait été manquée en partie par la détection basée sur l'image, à cause de l'occultation d'un outil par exemple. D'autre part, la détection basée image permet de détecter une incision avant que des déformations conséquentes soient observées. Cela limite le scénario où le recalage doit estimer une déformation très importante avec un modèle 3D obsolète, ce qui a de grande chance de ne pas fonctionner correctement. Comme nos résultats avec GeoDet étaient vraiment peu fiables, nous avons laissé cette combinaison pour de futurs travaux.

Chapitre 5

Conclusion

5.1 Travaux réalisés

5.1.1 Détection de contours occultants

L'objectif principal de cette thèse était de proposer des solutions pour résoudre automatiquement des tâches préliminaires au recalage initial pour réduire le temps nécessaire à la mise en place de la RA en salle d'opération. Dans un premier temps, la constitution d'un jeu de données a permis d'explorer l'étude de méthodes d'apprentissage profond pour résoudre la détection automatique des contours occultants et également la segmentation de masque contenant l'utérus. Les méthodes d'apprentissage profond sont relativement adéquates pour ce genre de tâches mais nous avons dû trouver des stratégies pour limiter l'épaisseur des prédictions proposées par ces solutions. Notre solution a permis de remplacer efficacement l'annotation manuelle des contours occultants dans le logiciel Uteraug en obtenant un résultat équivalent en quelques secondes contre plusieurs minutes auparavant.

Nous avons également proposé un score pour évaluer les contours occultants. Ce score est particulièrement adapté à la détection de contours par les méthodes d'apprentissage profond. Ces méthodes ont tendance à prédire des contours épais. Le score proposé est particulièrement sensible à l'épaisseur des contours ce qui n'est pas le cas des autres scores, en particulier pour les mesures d'erreur statistiques.

5.1.2 Mise à jour topologique du modèle virtuel

Une fois ces verrous technologiques levés nous avons décidé de nous intéresser à une autre limite du guidage par réalité augmentée. En effet, les méthodes actuelles pour la réalité augmentée ne permettent pas de maintenir cette assistance pendant l'incision de l'organe ce qui limite son usage. Nous avons ainsi proposé un ensemble d'étapes pour détecter l'incision et mettre à jour le modèle virtuel pour le recalage. Nous avons montré que cette approche donne de meilleurs résultats que l'approche sans mise à jour du modèle 3D. Cette approche fonctionne sur données ex-vivo et sur des données d'opérations réelles enregistrées. Malgré tout, cette approche n'est pas encore applicable dans des conditions réelles, en particulier car elle ne fonctionne pas en temps réel. De plus, il faut compléter le jeu de données qui est encore limité et ne permet pas encore une détection fiable des incisions potentielles. Une fois ce jeu étoffé, il pourrait être pertinent de combiner cette détection à celle d'autres éléments

présents, qui peuvent être corrélés à l'apparition de l'incision comme les outils et le saignement.

5.2 Perspectives

5.2.1 Utilisation du score proposé pour proposer une nouvelle fonction de coût

Le score que nous avons proposé est particulièrement sensible à l'épaisseur des contours prédits. Les stratégies proposées dans cette thèse pour réduire l'épaisseur des contours ne permettent pas encore de s'affranchir des opérations morphologiques réalisées après coup pour obtenir des contours d'une épaisseur d'un pixel. Convertir ce score pour être intégré comme une nouvelle pénalité ou une fonction de coût pourrait améliorer les méthodes de détection de contours par apprentissage profond.

5.2.2 Intégration de nos travaux dans le logiciel Uteraug

Nos travaux ont permis de montrer l'intérêt pratique des solutions proposées dans un cadre théorique. L'intégration dans un logiciel amené à assister le geste chirurgical dans un contexte de routine nécessite un certain nombre d'étapes supplémentaires. En particulier, il est nécessaire de valider notre étude sur un nombre de cas bien plus conséquent et d'étudier les situations qui ne fonctionnent pas comme attendues. Il est également nécessaire de mettre en place une situation pour permettre à l'utilisateur de confirmer l'annotation automatique et potentiellement de la modifier ou l'invalider complètement. Une notion encore plus poussée serait que le système d'annotation soit capable de fournir un niveau de confiance sur ces prédictions. Les approches actuelles nécessitent un réseau parallèle qui évalue les entrées du système pour les comparer aux données utilisées précédemment sur lesquelles le système est censé être fiable, ou indirectement en détectant des conditions difficiles que l'opérateur humain peut identifier : la présence de fumée, des images d'entrées floues, etc.

5.2.3 Amélioration du recalage non rigide

Comme nous l'avions mentionné dans la discussion du chapitre 4, la méthode de recalage non-rigide ne semble pas pouvoir accepter des déformations trop importantes. Nous avions testé d'utiliser la méthode *As-Rigid-As-Possible* [92] pour résoudre le problème du recalage mais cette méthode a donné des résultats peu satisfaisants. Cet effet de rigidité dans les déformations observées par ces méthodes de recalage est potentiellement dû au fait que l'incision n'a pas été exploitée davantage. On pourrait notamment distinguer la déformation due à l'incision de celle du reste de l'organe, qui est plus lisse. Une perspective pour de futurs travaux serait d'utiliser des modèles biomécaniques plus complexes, ou du moins qui autorisent plus de déformations. C'est un problème à long-terme car le but de cette étape est de pouvoir maintenir la réalité augmentée au cours de l'incision qui peut être le théâtre de déformations très importantes.

5.2.4 Implémentation du processus en temps réel

La démonstration du fonctionnement du processus développé dans le chapitre 4 a été réalisée sur des images sauvegardées depuis des opérations. Les efforts pour cette thèse ont été portés sur les concepts clés pour proposer une solution à la mise à jour du modèle 3D peropératoire pendant l'incision. La version actuelle ne permet pas de gérer un flux vidéo en temps réel. La prochaine étape est donc de trouver les moyens de rendre ces concepts réalisables dans les conditions réelles de l'opération. Les travaux présentés n'exploitent par exemple pas du tout la notion de temporalité de la séquence vidéo qui permettraient d'obtenir une meilleur initialisation pour le modèle de déformation ou de transfert de l'incision.

Annexe A

Détermination de la taille d'une base de données pour une tâche donnée

Les premiers résultats de cette section ont été publiés dans [128] et la rédaction d'un article de journal sur le jeu de données complet est en cours.

A.1 Contexte

L'apprentissage profond est une technique d'apprentissage très en vogue en ce moment. La méthode qui a donné et continue de donner de très bons résultats est l'approche supervisée. Pour rappel, cette approche consiste à entraîner un modèle à reproduire les étiquettes associées aux entrées du jeu de données. Pendant la phase d'entraînement, on mesure un taux d'erreur entre la réponse prédite par le modèle et l'étiquette associée à l'entrée utilisée. Cette erreur est utilisée pour modifier les paramètres du modèle de manière à améliorer la prédiction.

Cette approche suppose qu'un jeu de données conséquent et dédié à la tâche visée soit à disposition. Cela signifie qu'en plus de contenir une très grande quantité d'entrées, collectées dans des conditions variées pour limiter les biais possibles, il faut que la vérité terrain associée à chacune de ces entrées soit également disponible. Derrière les résultats souvent impressionnant de l'apprentissage profond se cache des heures et des heures d'annotations manuelles souvent laborieuses et répétitives.

Dans le cadre de jeux de données sur des images naturelles, cette charge de travail manuel est souvent diluée par un nombre conséquent de participants, via des appels à participation ou indirectement en remplissant ces fameux captcha pour vérifier que vous n'êtes pas un robot. Le fait d'avoir plusieurs annotateurs permet également de limiter le potentiel biais lié à la façon d'annoter d'une personne spécifique. Cette option est possible lorsque le savoir collecté est connu par une grande partie de la population sans a priori sur la culture ou le niveau de connaissance sur un sujet particulier. Il est par exemple facile pour n'importe qui de repérer un chien, une voiture, un feu tricolore dans une image donnée. Mais sauriez-vous par contre repérer ou même classifier une tumeur à partir d'une image cœlioscopique ou d'une imagerie 3D ? Il est très probable que non. Il s'agit là de la grande difficulté pour obtenir un jeu de données annotées conséquent sur un domaine très spécifique comme celui de la santé.

Dans le cadre des jeux de données médicales, au-delà des contraintes juridiques (notamment sur le consentement éclairé et l'anonymisation), le nombre de personnes

Nombres d'images par opération	1-10	11-20	21-30	31-40	41-50	51-100	101-694
Nombre d'opérations	16	11	14	7	2	18	5

TABLE A.1 – Tableau de la répartition du nombre d'images par opération.

capables de répondre de manière fiable est très limité. Ces personnes sont d'ailleurs bien souvent très occupées pour ces mêmes raisons. C'est ce qui explique l'engouement pour des méthodes qui permettent de faciliter l'annotation et également pour les méthodes qui utilisent peu de données annotées ou des données faiblement annotées avec des apprentissages dit faiblement supervisé (*weekly supervised learning* en anglais).

Nous avons dit plus tôt que l'efficacité des méthodes d'apprentissage profond reposait sur l'utilisation d'une grande quantité de données annotées. La question est donc quelle quantité de données faut-il collecter pour une tâche donnée ? Il est bien évident que cette quantité dépend de la difficulté de la tâche elle-même et de la précision désirée. A notre échelle, nous avons voulu constater le gain de précision obtenu en augmentant petit à petit la taille d'un jeu de donnée dédiée de l'utérus, pour la segmentation et la détection de contours.

A.2 Description de l'expérience

Nous avons collecté 3815 images cœlioscopiques d'opérations en gynécologie et nous les avons annoté pour deux types de tâches : la segmentation de l'utérus et la détection de contours occultants de l'utérus. Nous voulons évaluer la précision du réseau de neurones pour vérifier si cette quantité d'images annotées est suffisante pour les deux tâches. Pour cela, nous proposons une expérience qui consiste à entraîner un réseau de neurones avec un jeu d'entraînement dont la taille augmente, avec un jeu de test fixe. Nous évaluerons les performances du réseau en fonction de la taille du jeu d'apprentissage.

A.2.1 Jeu de données

Le jeu de données est composé de 3815 images extraites de 79 vidéos d'opérations de l'utérus, 29 obtenues via une étude validée par le CPP et 50 depuis des vidéos extraites de Youtube. La collecte et l'annotation de ces images a été réalisées par Sabrina Madad Zadeh, interne en médecine, au cours de son Master de Recherche courant 2018.

On évalue souvent la qualité d'un jeu de données à la quantité d'images annotées qu'elle contient mais il est capital que ces images représentent des situations différentes de celles déjà présentes dans le jeu de données. L'extraction de nos images a été réalisée manuellement pour collecter uniquement des situations qui paraissent nouvelles aux yeux de l'annotateur. Une option parfois utilisée dans la construction de jeux de données est d'extraire régulièrement, à une fréquence prédéfinie, des images d'une séquence vidéo. Cette approche fournit une quantité extravagante d'images très similaires qui n'apportent quasiment rien à l'entraînement.

Le jeu de données a techniquement été annoté pour les contours de l'utérus qui se distinguent en 3 catégories : les contours occultants de l'organe, les contours occultés de l'organe et les contours dit de fermeture. Cette dernière catégorie permet

de traiter les contours difficiles à traiter, comme les jonctions avec les trompes, pour lesquelles la frontière anatomique est floue. Le but de cette dernière classe de contour est de compléter les deux autres types de contours avec qui elle définit une région fermée pour définir la segmentation de l'utérus.

Ce jeu de données est séparée en 3 : le jeu d'entraînement qui sera utilisé directement par le réseau de neurones pour répondre à la tâche ; le jeu de validation qui est évalué régulièrement pendant l'entraînement, notamment pour vérifier que l'apprentissage fonctionne et se généralise à d'autres images que celles du jeu d'entraînement ; le jeu de test qui contient des images qui n'ont pas été utilisées ou évaluées pendant l'entraînement. Les résultats seront donnés sur ce jeu de test.

Le jeu de données totale est décomposée en 76%/8%/16%, respectivement pour le jeu d'entraînement, de validation et de test. Cette décomposition s'assure que les images qui proviennent d'une même opération ne sont pas séparées sur plusieurs jeux de données.

Nous avons décidé de décomposer le jeu d'entraînement par groupe de 100 images. Cette décomposition suit les dossiers des différentes opérations utilisées. C'est-à-dire que pour compléter le nouveau jeu d'entraînement, on complète avec les images de l'opération courante jusqu'à atteindre la nouvelle taille désirée. Si l'ensemble des images de l'opération courante a été utilisé, on sélectionne l'opération suivante. L'idéal aurait été d'avoir le même nombre d'images par opération, nous avons préféré une découpe par le nombre d'images plutôt que par les différentes opérations.

A.2.2 Métriques

Pour l'évaluation de la segmentation de l'utérus, on utilisera les métriques classiques de classification. La définition de ces métriques est présentée dans la section 3.2.1.2 dans le chapitre 3. Pour rappel, on obtient les catégories vrai-positif (VP), vrai-négatif (VN), faux-positif (FP), faux-négatif (FN) en superposant la prédiction et de la vérité terrain. En agglomérant cette classification pour tous les pixels de l'image, on obtient la quantité de chaque catégorie, notée $|VP|, |VN|, |FP|, |FN|$, pour définir différentes métriques :

$$p = \frac{|VP|}{|FP| + |VP|}; \quad r = \frac{|VP|}{|FN| + |VP|}; \quad IoU = \frac{|VP|}{|FP| + |FN| + |VP|}; \tag{A.1}$$

La précision (p) mesure le rapport de pixel détectés pertinents parmi les pixels détectés, le rappel mesure le rapport de pixels détectés pertinents parmi les pixels à détecter. La métrique IoU (*Intersection-over-Union* en anglais) est très utilisée pour évaluer les résultats en segmentation sémantique. Comme son nom l'indique, elle consiste à évaluer le rapport entre l'intersection de la prédiction et de la vérité terrain et l'union de ces régions. Elle représente un compromis entre la précision et le rappel. Ces scores sont des mesures de similarités. On suppose que ces mesures vont augmenter avec la taille du jeu d'apprentissage.

Pour la détection de contours occultants, nous allons utiliser le score de contour proposé dans la section 3.4 dans le chapitre 3. Ce score utilise une distance seuil d_{max} pour définir la matrice de confusion. Cette approche permet de considérer une prédiction proche de la vérité terrain comme un vrai-positif. Contrairement aux métriques de classification utilisées pour la segmentation, le score est une mesure d'erreur basée sur une mesure de distance entre la prédiction et la vérité terrain. On

suppose ainsi que ce score diminue avec l'augmentation de la taille du jeu d'apprentissage.

A.2.3 Protocole

Nous utilisons le réseau U-Net pour appendre les deux types de tâches. C'est une architecture relativement simple par rapport à d'autres architectures d'apprentissage profond. C'est un type de réseau très utilisé pour des situations avec un jeu de données limité. Nous utilisons malgré tout une augmentation du jeu de données en générant aléatoirement diverses altérations (zoom, rotation, cisaillement, flou gaussien, bruit). On utilise l'entropie croisée pondérée comme fonction de coût. Les poids permettent de rééquilibrer le rapport de fréquence des différentes classes. Dans l'ensemble du jeu de données, il y a une large prédominance des pixels non-contours. La solution pondérée de la fonction de coût permet de rendre les erreurs de classification sur les pixels contours plus importantes que pour les autres pixels. Pour la détection de contours, les poids des classes sont $0,1$, 1 et $1,5$, respectivement pour les classes non-contours, contours occultants, contours d'occultation. Pour la segmentation, les poids sont de $1,0$ pour les pixels de l'utérus et $0,1$ pour les autres pixels.

Pour chaque étape, le réseau est initialisé avec les mêmes poids aléatoires. Le réseau est entraîné sur 50 époques quelque soit la taille du jeu d'entraînement. Les prédictions sont obtenues sur le jeu de test avec le modèle entraîné pour la 50^{e} époque.

Plutôt que de calculer les métriques pour chaque image et d'extraire ensuite une moyenne qui peut altérer l'évaluation en donnant plus d'importance à certains pixels. Nous concaténons l'ensemble des résultats pour calculer les métriques sur l'ensemble du jeu de données.

A.3 Résultats

Pour les deux types de tâches, on constate que plus la taille du jeu de données est grande, meilleurs sont les résultats.

Cependant, on peut constater l'apparition d'un plateau à partir duquel l'évolution des différentes métriques stagne. Pour la détection de contours occultants, on constate qu'au-delà de 1600 images, le score de contour ne diminue que très peu (il vaut 5,35 à 1600 et 5,07 à 2800).

On peut remarquer que, quelle que soit la taille du jeu d'entraînement, le modèle entraîné ne produit pas de faux positifs. Le score de Faux-Négatifs indique que le modèle tend à détecter de plus en plus de contours. Le score des VP, qui traduit à quel point les pixels détectés comme contours sont proches des pixels labellisés comme contours, est le terme qui progresse le moins.

Pour la segmentation de l'utérus, les 3 métriques augmentent rapidement jusqu'à 900 images. Au-delà, on perçoit une hausse plus légère mais relativement régulière pour le rappel et IoU. Dans cette seconde zone, la précision tend à stagner autour de 0,925.

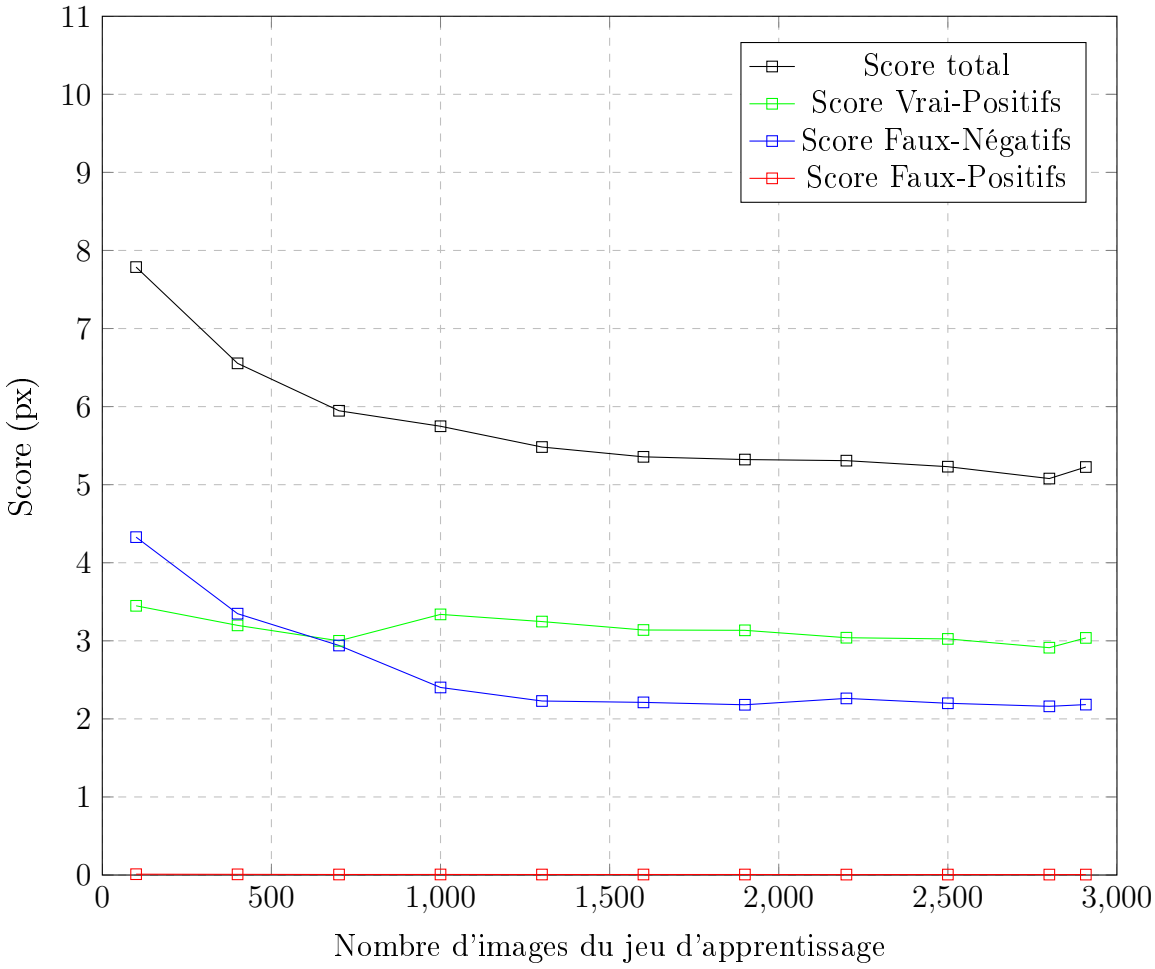


FIGURE A.1 – Évolution de l'impact de la taille du jeu d'apprentissage sur la précision de la tâche de détection des contours occultants de l'utérus (OC2D). Le paramètre d_{max} utilisé pour définir la limite des différentes régions est fixé à 11 px. Les métriques présentées sont des mesures d'erreur. L'idéal vaut 0 si et seulement si la prédiction correspond exactement à la vérité terrain.

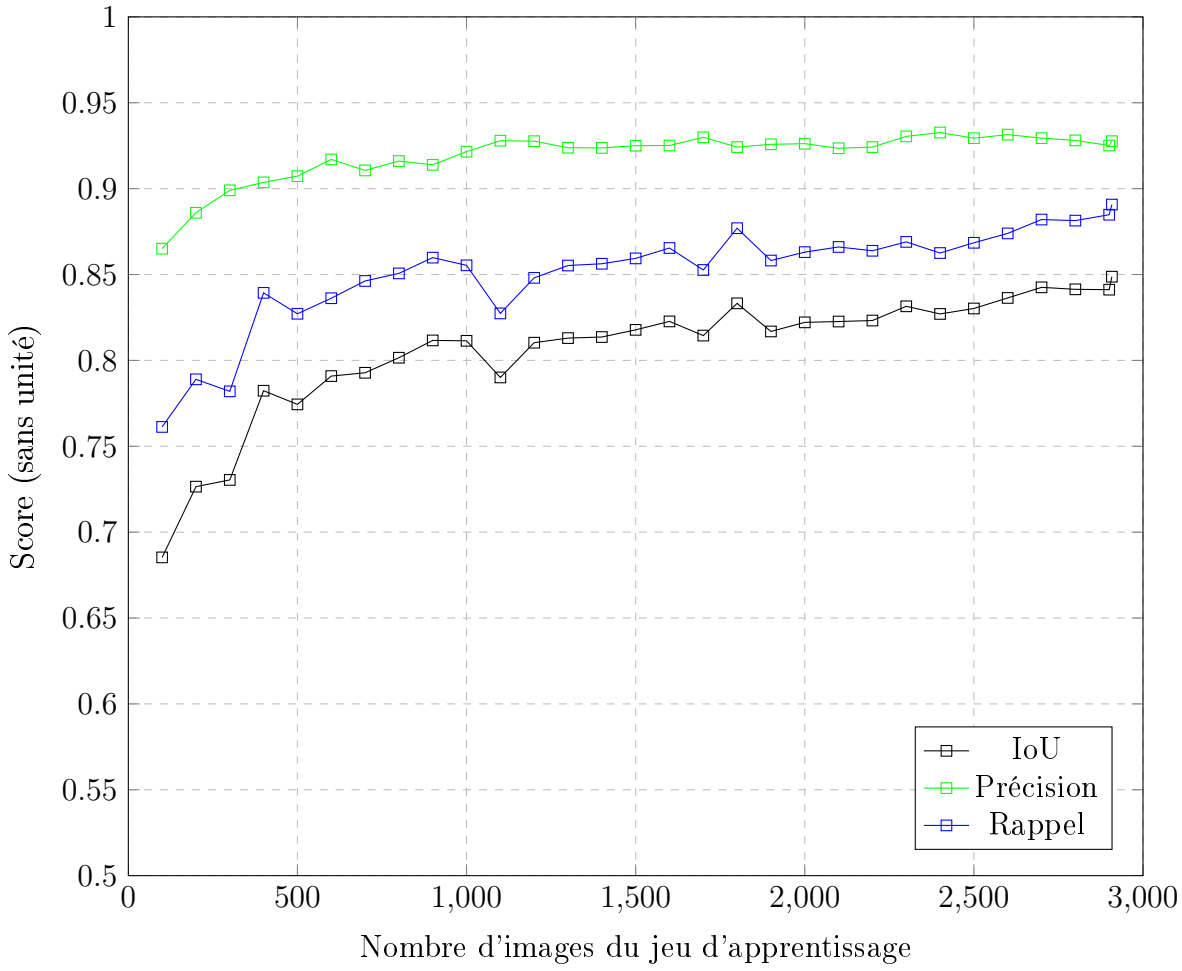
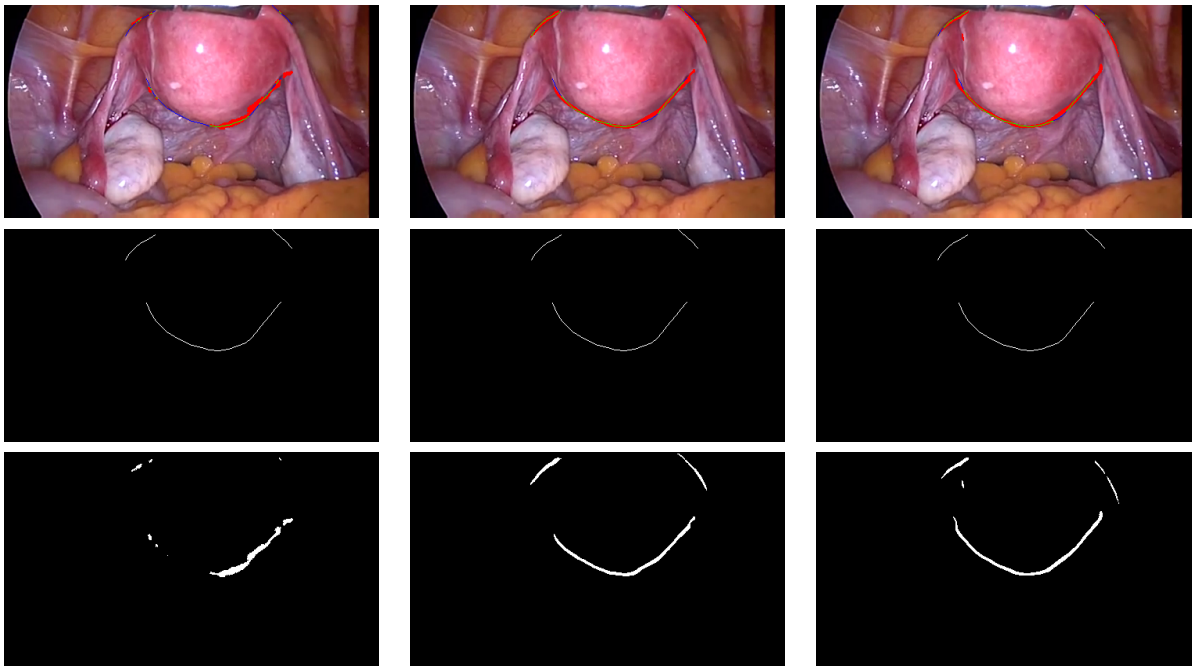
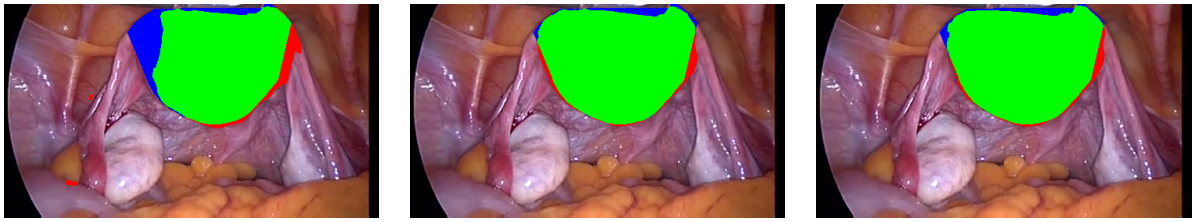


FIGURE A.2 – Évolution de l'impact de la taille du jeu d'apprentissage sur la précision de la tâche de segmentation de l'utérus. Pour les métriques présentées, la valeur idéale est 1.



100 images	1300 images	2900 images
S = 6,56px	S = 4,31 px	S = 4,34 px

FIGURE A.3 – Illustration de l'évolution de la détection de contours occultants en fonction de la taille du jeu d'apprentissage. La première ligne représente l'image d'entrée combinée avec la vérité terrain et la prédiction (TP=vert, FP=rouge, FN=bleu). La deuxième ligne représente la vérité terrain et la troisième ligne la prédiction.



100 images	1300 images	2900 images
IoU = 80,2%	IoU = 92,2%	IoU = 91,8%

FIGURE A.4 – Illustration de l'évolution de la segmentation de l'utérus en fonction de la taille du jeu d'apprentissage. L'image d'entrée est combinée avec la vérité terrain et la prédiction (TP=vert, FP=rouge, FN=bleu).

A.4 Discussion

A.4.1 Limites de l'annotation

Dans plusieurs travaux dédiés à la détection de contours [4, 126], une réflexion est développée sur le potentiel décalage qui existe entre les annotations, réalisées par des annotateurs humains dans un temps souvent limité, et les contours réels. Ce décalage, bien que minime, pénalise malgré tout l'entraînement car il crée une confusion sur ce qui doit être considéré comme contour ou non. Ces travaux proposent alors une manière de réaligner les annotations sur ce que le réseau estime être la position réelle du contour. C'est une hypothèse que nous n'avions pas retenue dans un premier temps. L'annotation du jeu de données dédiée, bien qu'étant une tâche laborieuse, a été menée par une interne en médecine, et vérifiée par la suite. Il semblait donc peu probable que de tels décalages puissent être présents.

Force est de constater que sur certaines images, on peut constater de légers décalages entre l'annotation et la vérité terrain. C'est le cas sur l'exemple présenté sur la figure A.4 pour la segmentation : la fine bande (bleue) qui est considérée comme faux-négatif est en réalité une zone de l'utérus occulté par un outil. Il est possible que ce genre d'erreur soit apparu lors de la fusion des annotations de contours pour obtenir un masque de l'utérus.

Il est également à noter que la plupart des annotations sont réalisées en utilisant des lignes brisées. Il faut également noter que l'annotation des contours a été réalisée grâce à des outils pour marquer des lignes brisées. C'est une modalité pratique pour l'annotation mais qui peut en effet créer des décalages de l'ordre du pixel. Les portions de lignes réalisées de s'alignent ainsi pas parfaitement à la réalité des objets capturés dans l'image. Cette méthode d'annotation crée donc naturellement des décalages dans les annotations.

Il serait intéressant de tester l'utilisation de réalignement des annotations tout en faisant varier la taille du jeu d'apprentissage et la comparer avec notre approche.

A.4.2 Utilisation de modèles pré-entraînés

Nous n'avons pas utilisé de poids pré-entraînés sur une tâche connexe pour initialiser le réseau U-Net. Nous avons fait ce choix pour vérifier la capacité du réseau à apprendre uniquement en fonction des images présentées. Dans ce genre de problématiques, on utilise souvent un réseau pré-entraîné sur des images naturelles pour une tâche similaire ou sur une tâche distincte mais avec des données similaires. Il pourrait être intéressant de vérifier l'impact de l'augmentation du jeu de données d'apprentissage avec un réseau pré-entraîné. On peut faire l'hypothèse que dans cette configuration, les premières étapes avec un jeu de données réduit obtiendraient une meilleur précision par rapport à l'approche sans pré-entraînement. L'impact sur la précision des étapes finales est plus difficile à anticiper. De notre expérience, l'utilisation d'un modèle pré-entraîné sur des données similaires, même pour une tâche différente, améliore de manière radicale la précision finale de l'entraînement.

A.4.3 Tirages aléatoires

Nous avons réalisé un seul tirage pour la construction progressive du jeu d'entraînement. L'ordre dans lequel les données ont été triés pourrait avoir une influence sur

la qualité de l'entraînement. Pour s'assurer que cette influence est limitée, il faudrait refaire la construction avec plusieurs tirages aléatoires.

A.5 Conclusion

Pour cette application, nous avons pu constater qu'un premier ordre de grandeur de nombre d'images pouvaient être trouvé pour permettre de répondre correctement, à défaut de précisément, à la tâche demandée. Ce nombre se trouve autour de 1000 pour la segmentation sémantique de l'utérus et de 1300 pour la détection de contours occultants de l'utérus.

Ces conclusions dépendent évidemment de la tâche, du domaine des images utilisées, et potentiellement de l'architecture du réseau utilisée. Une expérience complémentaire pourrait être de comparer ces observations avec différents types d'architectures.

Bibliographie

[1] Blender v2.81a. URL https://www.blender.org/

[2] Photoscan. URL https://www.agisoft.com

[3] Supervisely. URL https://supervise.ly/

[4] Acuna, D., Kar, A., Fidler, S. : Devil is in the edges : Learning semantic boundaries from noisy annotations. In : CVPR (2019)

[5] Adagolodjo, Y., Trivisonne, R., Haouchine, N., Cotin, S., Courtecuisse, H. : Silhouette-based pose estimation for deformable organs application to surgical augmented reality. In : 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 539–544. IEEE (2017)

[6] Amir-Khalili, A., Nosrati, M.S., Peyrat, J.M., Hamarneh, G., Abugharbieh, R. : Uncertainty-encoded augmented reality for robot-assisted partial nephrectomy : A phantom study. In : Augmented Reality Environments for Medical Imaging and Computer-Assisted Interventions, pp. 182–191. Springer (2013)

[7] Andrade-Loarca, H., Kutyniok, G., Öktem, O. : Shearlets as feature extractor for semantic edge detection : the model-based and data-driven realm. Proceedings of the Royal Society A **476**(2243), 20190841 (2020)

[8] Arbelaez, P., Maire, M., Fowlkes, C., Malik, J. : Contour detection and hierarchical image segmentation. IEEE transactions on pattern analysis and machine intelligence **33**(5), 898–916 (2010)

[9] Bartoli, A. : Maximizing the predictivity of smooth deformable image warps through cross-validation. Journal of Mathematical Imaging and Vision **31**(2-3), 133–145 (2008)

[10] Bartoli, A., Gérard, Y., Chadebecq, F., Collins, T., Pizarro, D. : Shape-from-template. IEEE transactions on pattern analysis and machine intelligence **37**(10), 2099–2118 (2015)

[11] Bay, H., Tuytelaars, T., Van Gool, L. : Surf : Speeded up robust features. In : European conference on computer vision, pp. 404–417. Springer (2006)

[12] Bengio, Y., LeCun, Y., et al. : Scaling learning algorithms towards ai. Large-scale kernel machines **34**(5), 1–41 (2007)

[13] Bernhardt, S., Nicolau, S.A., Soler, L., Doignon, C. : The status of augmented reality in laparoscopic surgery as of 2016. Medical image analysis **37**, 66–90 (2017)

[14] Bertasius, G., Shi, J., Torresani, L. : Deepedge : A multi-scale bifurcated deep network for top-down contour detection. In : Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4380–4389 (2015)

[15] Bielser, D., Glardon, P., Teschner, M., Gross, M. : A state machine for real-time cutting of tetrahedral meshes. In : 11th Pacific Conference onComputer Graphics and Applications, 2003. Proceedings., pp. 377–386. IEEE (2003)

[16] Botchorishvili, R., Velemir, L., Wattiez, A., Tran, X., Bolandard, F., Rabischong, B., Jardon, K., Pouly, J.L., Mage, G., Canis, M. : Coelioscopie et coeliochirurgie : principes généraux et instrumentation. Accès 23 septembre 2021 [en ligne]. URL http://campus.cerimes.fr/chirurgie-generale/enseignement/coelioscopie/site/html/1.html

[17] Brunet, J.N., Mendizabal, A., Petit, A., Golse, N., Vibert, E., Cotin, S. : Physics-based deep neural network for augmented reality during liver surgery. In : International Conference on Medical image computing and computer-assisted intervention, pp. 137–145. Springer (2019)

[18] Canny, J.F. : A computational approach to edge detection. TPAMI **8**(6), 679–698 (1986)

[19] Chen, L.C., Barron, J.T., Papandreou, G., Murphy, K., Yuille, A.L. : Semantic image segmentation with task-specific edge detection using cnns and a discriminatively trained domain transform. In : Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4545–4554 (2016)

[20] Chollet, F. : Xception : Deep learning with depthwise separable convolutions. In : Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1251–1258 (2017)

[21] Cignoni, P., Callieri, M., Corsini, M., Dellepiane, M., Ganovelli, F., Ranzuglia, G. : MeshLab : an Open-Source Mesh Processing Tool. In : V. Scarano, R.D. Chiara, U. Erra (eds.) Eurographics Italian Chapter Conference. The Eurographics Association (2008). DOI 10.2312/LocalChapterEvents/ItalChap/ItalianChapConf2008/129-136

[22] Collins, T., Bartoli, A., Bourdel, N., Canis, M. : Segmenting the uterus in monocular laparoscopic images without manual input. In : International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 181–189. Springer (2015)

[23] Collins, T., Chauvet, P., Debize, C., Pizarro, D., Bartoli, A., Canis, M., Bourdel, N. : A system for augmented reality guided laparoscopic tumour resection with quantitative ex-vivo user evaluation. In : International Workshop on Computer-Assisted and Robotic Endoscopy, pp. 114–126. Springer (2016)

[24] Collins, T., Pizarro, D., Bartoli, A., Canis, M., Bourdel, N. : Real-time wide-baseline registration of the uterus in monocular laparoscopic videos. In : International Workshop on Medical Imaging and Augmented Reality at MICCAI (2013)

[25] Collins, T., Pizarro, D., Bartoli, A., Canis, M., Bourdel, N. : Computer-assisted laparoscopic myomectomy by augmenting the uterus with pre-operative mri data. In : 2014 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), pp. 243–248. IEEE (2014)

[26] Collins, T., Pizarro, D., Gasparini, S., Bourdel, N., Chauvet, P., Canis, M., Calvet, L., Bartoli, A. : Augmented reality guided laparoscopic surgery of the uterus. IEEE Transactions on Medical Imaging **40**(1), 371–380 (2020)

[27] Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B. : The cityscapes dataset for semantic urban scene understanding. In : Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)

[28] Courtecuisse, H., Allard, J., Kerfriden, P., Bordas, S.P., Cotin, S., Duriez, C. : Real-time simulation of contact and cutting of heterogeneous soft-tissues. Medical image analysis **18**(2), 394–410 (2014)

[29] Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L. : Imagenet : A large-scale hierarchical image database. In : 2009 IEEE conference on computer vision and pattern recognition, pp. 248–255. Ieee (2009)

[30] Deng, R., Shen, C., Liu, S., Wang, H., Liu, X. : Learning to predict crisp boundaries. In : ECCV (2018)

[31] Doersch, C., Gupta, A., Efros, A.A. : Unsupervised visual representation learning by context prediction. In : Proceedings of the IEEE international conference on computer vision, pp. 1422–1430 (2015)

[32] Dollár, P., Zitnick, C.L. : Fast edge detection using structured forests. IEEE transactions on pattern analysis and machine intelligence **37**(8), 1558–1570 (2014)

[33] Dubuisson, M., Jain, A. : A modified hausdorff distance for object matching. In : ICPR (1994)

[34] Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A. : The PASCAL Visual Object Classes Challenge 2011 (VOC2011) Results. http ://www.pascal-network.org/challenges/VOC/voc2011/workshop/index.html

[35] de Farias Macedo, M.C., Júnior, A.L.A., dos Santos Souza, A.C., Giraldi, G.A. : High-quality on-patient medical data visualization in a markerless augmented reality environment. Journal on Interactive Systems **5**(3) (2014)

[36] Fischler, M.A., Bolles, R.C. : Random sample consensus : a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM **24**(6), 381–395 (1981)

[37] Forsyth, D., Ponce, J. : Computer vision : A modern approach. Prentice hall (2011)

[38] François, T., Calvet, L., Sève-d'Erceville, C., Bourdel, N., Bartoli, A. : Image-based incision detection for intraoperative 3d model update in augmented reality assisted laparoscopic surgery. In : M. de Bruijne, P.C. Cattin, S. Cotin, N. Padoy, S. Speidel, Y. Zheng, C. Essert (eds.) MICCAI, pp. 647–656. Springer International Publishing, Cham (2021)

[39] François, T., Calvet, L., Zadeh, S.M., Saboul, D., Gasparini, S., Samarakoon, P., Bourdel, N., Bartoli, A. : Detecting the occluding contours of the uterus to automatise augmented laparoscopy : score, loss, dataset, evaluation and user study. International journal of computer assisted radiology and surgery **15**(7), 1177–1186 (2020)

[40] François, T., Debize, C., Calvet, L., Collins, T., Pizarro, D., Bartoli, A. : Uteraug : Augmented reality in laparoscopic surgery of the uterus. Démonstration présentée lors de la conférence ISMAR en octobre 2017 (2017)

[41] Fukushima, K., Miyake, S. : Neocognitron : A self-organizing neural network model for a mechanism of visual pattern recognition. In : Competition and cooperation in neural nets, pp. 267–285. Springer (1982)

[42] Garcia-Peraza-Herrera, L.C., Li, W., Fidon, L., Gruijthuijsen, C., Devreker, A., Attilakos, G., Deprest, J., Vander Poorten, E., Stoyanov, D., Vercauteren, T., et al. : Toolnet : holistically-nested real-time segmentation of robotic surgical tools. In : 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 5717–5722. IEEE (2017)

[43] Garry, R. : Laparoscopic surgery. Best Practice & Research Clinical Obstetrics & Gynaecology **20**(1), 89–104 (2006)

[44] Gay-Bellile, V., Bartoli, A., Sayd, P. : Direct estimation of nonrigid registrations with image-based self-occlusion reasoning. IEEE Transactions on Pattern Analysis and Machine Intelligence **32**(1), 87–104 (2008)

[45] Goodfellow, I., Bengio, Y., Courville, A. : Deep learning. MIT press (2016)

[46] Grard, M., Chen, L., Dellandréa, E. : Bicameral structuring and synthetic imagery for jointly predicting instance boundaries and nearby occlusions from a single image. arXiv (2019)

[47] Grigorescu, C., Petkov, N., Westenberg, M.A. : Contour detection based on non classical receptive field inhibition. IEEE Transactions on image processing **12**(7), 729–739 (2003)

[48] Grill-Spector, K., Kushnir, T., Edelman, S., Itzchak, Y., Malach, R. : Cue-invariant activation in object-related areas of the human occipital lobe. Neuron **21**(1), 191–202 (1998). DOI https://doi.org/10.1016/S0896-6273(00)80526-7. URL https://www.sciencedirect.com/science/article/pii/S0896627300805267

[49] Griwodz, C., Gasparini, S., Calvet, L., Gurdjos, P., Castan, F., Maujean, B., Lillo, G.D., Lanthony, Y. : Alicevision Meshroom : An open-source 3D reconstruction pipeline. In : Proc. 12th ACM Multimed. Syst. Conf. - MMSys '21. ACM Press (2021). DOI 10.1145/3458305.3478443

[50] Han, L., Wang, H., Liu, Z., Chen, W., Zhang, X. : Vision-based cutting control of deformable objects with surface tracking. IEEE/ASME Transactions on Mechatronics (2020)

[51] Haouchine, N., Dequidt, J., Berger, M.O., Cotin, S. : Monocular 3d reconstruction and augmentation of elastic surfaces with self-occlusion handling. IEEE transactions on visualization and computer graphics **21**(12), 1363–1376 (2015)

[52] Haouchine, N., Dequidt, J., Peterlik, I., Kerrien, E., Berger, M.O., Cotin, S. : Image-guided simulation of heterogeneous tissue deformation for augmented reality during hepatic surgery. In : 2013 IEEE international symposium on mixed and augmented reality (ISMAR), pp. 199–208. IEEE (2013)

[53] Hariharan, B., Arbeláez, P., Bourdev, L., Maji, S., Malik, J. : Semantic contours from inverse detectors. In : 2011 International Conference on Computer Vision, pp. 991–998. IEEE (2011)

[54] Hasan, M.K., Calvet, L., Rabbani, N., Bartoli, A. : Detection, segmentation, and 3d pose estimation of surgical tools using convolutional neural networks and algebraic geometry. Medical Image Analysis **70**, 101994 (2021)

[55] Hattab, G., Arnold, M., Strenger, L., Allan, M., Arsentjeva, D., Gold, O., Simpfendörfer, T., Maier-Hein, L., Speidel, S. : Kidney edge detection in laparoscopic image data for computer-assisted surgery. International journal of computer assisted radiology and surgery **15**(3), 379–387 (2020)

[56] He, K., Zhang, X., Ren, S., Sun, J. : Deep residual learning for image recognition. In : Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778 (2016)

[57] Hubel, D.H., Wiesel, T.N. : Receptive fields of single neurones in the cat's striate cortex. The Journal of physiology **148**(3), 574–591 (1959)

[58] ISCAS : Miccai endoscopic vision challenges (2019). URL endovis.grand-challenge.org

[59] Kelm, A.P., Zölzer, U. : Walk the lines : Object contour tracing cnn for contour completion of ships. arXiv preprint arXiv :2004.06587 (2020)

[60] Kim, M., Woo, S., Kim, D., Kweon, I.S. : The devil is in the boundary : Exploiting boundary representation for basis-based instance segmentation. In : Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 929–938 (2021)

[61] Kokkinos, I. : Pushing the boundaries of boundary detection using deep learning. arXiv preprint arXiv :1511.07386 (2015)

[62] Koo, B., Ozgur, E., Roy, B.L., Buc, E., Bartoli, A. : Deformable registration of a preoperative 3d liver volume to a laparoscopy image using contour and shading cues. In : MICCAI (2017)

[63] Krizhevsky, A., Sutskever, I., Hinton, G.E. : Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems **25**, 1097–1105 (2012)

[64] Larsson, G., Maire, M., Shakhnarovich, G. : Learning representations for automatic colorization. In : European conference on computer vision, pp. 577–593. Springer (2016)

[65] Le Cun, Y., Jackel, L.D., Boser, B., Denker, J.S., Graf, H.P., Guyon, I., Henderson, D., Howard, R.E., Hubbard, W. : Handwritten digit recognition : Applications of neural network chips and automatic learning. IEEE Communications Magazine **27**(11), 41–46 (1989)

[66] Lee, C.Y., Xie, S., Gallagher, P., Zhang, Z., Tu, Z. : Deeply-supervised nets. In : Artificial intelligence and statistics, pp. 562–570. PMLR (2015)

[67] Leibetseder, A., Petscharnig, S., Primus, M.J., Kletz, S., Münzer, B., Schoeffmann, K., Keckstein, J. : Lapgyn4 : a dataset for 4 automatic content analysis problems in the domain of laparoscopic gynecology. In : Proceedings of the 9th ACM Multimedia Systems Conference, MMSys, pp. 357–362 (2018)

[68] Lepetit, V., Moreno-Noguer, F., Fua, P. : Epnp : An accurate o (n) solution to the pnp problem. International journal of computer vision **81**(2), 155 (2009)

[69] Lim, J.J., Zitnick, C.L., Dollár, P. : Sketch tokens : A learned mid-level representation for contour and object detection. In : Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3158–3165 (2013)

[70] Litjens, G., Kooi, T., Bejnordi, B.E., Setio, A.A.A., Ciompi, F., Ghafoorian, M., Van Der Laak, J.A., Van Ginneken, B., Sánchez, C.I. : A survey on deep learning in medical image analysis. Medical image analysis **42**, 60–88 (2017)

[71] Liu, G., Haralick, R.M. : Optimal matching problem in detection and recognition performance evaluation. Pattern Recognition **35**(10), 2125–2139 (2002)

[72] Liu, Y., Cheng, M., Hu, X., Bian, J., Zhang, L., Bai, X., Tang, J. : Richer convolutional features for edge detection. TPAMI **41**(8), 1939–1946 (2019)

[73] Liu, Y., Cheng, M.M., Fan, D.P., Zhang, L., Bian, J., Tao, D. : Semantic edge detection with diverse deep supervision. arXiv preprint arXiv :1804.02864 (2018)

[74] Lloyd, S. : Least squares quantization in pcm. IEEE transactions on information theory **28**(2), 129–137 (1982)

[75] Lopez-Molina, C., Baets, B.D., Sola, H.B. : Quantitative error measures for edge detection. Pattern Recognition **46**(4), 1125–1139 (2013)

[76] Lowe, D.G. : Object recognition from local scale-invariant features. In : Proceedings of the seventh IEEE international conference on computer vision, vol. 2, pp. 1150–1157. Ieee (1999)

[77] Lowe, D.G. : Distinctive image features from scale-invariant keypoints. International journal of computer vision **60**(2), 91–110 (2004)

[78] Maar, D. : Vision : A Computational Investigation into the Human Representation and Processing of Visual Information. The MIT Press (2010). DOI 10.7551/mitpress/9780262514620.001.0001

[79] Mage, G. : Chirurgie cœlioscopique en gynécologie. Elsevier Health Sciences (2013)

[80] Magnier, B., Abdulrahman, H., Montesinos, P. : A review of supervised edge detection evaluation methods and an objective comparison of filtering gradient computations using hysteresis thresholds. J. Imaging **4**(6), 74 (2018)

[81] Mahmoud, N., Grasa, Ó.G., Nicolau, S.A., Doignon, C., Soler, L., Marescaux, J., Montiel, J. : On-patient see-through augmented reality based on visual slam. International journal of computer assisted radiology and surgery **12**(1), 1–11 (2017)

[82] Maier-Hein, L., Mountney, P., Bartoli, A., Elhawary, H., Elson, D., Groch, A., Kolb, A., Rodrigues, M., Sorger, J., Speidel, S., et al. : Optical techniques for 3d surface reconstruction in computer-assisted laparoscopic surgery. Medical image analysis **17**(8), 974–996 (2013)

[83] Malik, J. : Cs280 : Computer vision. Accès 23 septembre 2021 [en ligne] (2015). URL https://inst.eecs.berkeley.edu/~cs280/sp15/

[84] Martin, D., Fowlkes, C., Tal, D., Malik, J. : A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In : Proc. 8th Int'l Conf. Computer Vision, vol. 2, pp. 416–423 (2001)

[85] Martin, D.R., Fowlkes, C.C., Malik, J. : Learning to detect natural image boundaries using local brightness, color, and texture cues. PAMI **26**(5), 530–549 (2004)

[86] Mendizabal, A., Tagliabue, E., Brunet, J.N., Dall'Alba, D., Fiorini, P., Cotin, S. : Physics-based deep neural network for real-time lesion tracking in ultrasound-guided breast biopsy. In : Computational Biomechanics for Medicine, pp. 33–45. Springer (2019)

[87] Mitchell, T. : Machine learning (1997)

[88] Molino, N., Bao, Z., Fedkiw, R. : A virtual node algorithm for changing mesh topology during simulation. ACM Transactions on Graphics (TOG) **23**(3), 385–392 (2004)

[89] Mottaghi, R., Chen, X., Liu, X., Cho, N.G., Lee, S.W., Fidler, S., Urtasun, R., Yuille, A. : The role of context for object detection and semantic segmentation in the wild. In : IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2014)

[90] Nicolau, S., Soler, L., Mutter, D., Marescaux, J. : Augmented reality in laparoscopic surgical oncology. Surgical oncology **20**(3), 189–201 (2011)

[91] Ozgur, E., Koo, B., Roy, B.L., Buc, E., Bartoli, A. : Preoperative liver registration for augmented monocular laparoscopy using backward-forward biomechanical simulation. IJCARS **13**(10) (2018)

[92] Parashar, S., Pizarro, D., Bartoli, A., Collins, T. : As-rigid-as-possible volumetric shape-from-template. In : ICCV (2015)

[93] Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A. : Context encoders : Feature learning by inpainting. In : Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2536–2544 (2016)

[94] Paulus, C.J., Haouchine, N., Kong, S., Soares, R.V., Cazier, D., Cotin, S. : Handling topological changes during elastic registration. IJCARS **12**(3), 461–470 (2017)

[95] Pfeiffer, M., Riediger, C., Weitz, J., Speidel, S. : Learning soft tissue behavior of organs for surgical navigation with convolutional neural networks. International journal of computer assisted radiology and surgery **14**(7), 1147–1155 (2019)

[96] Pizarro, D., Bartoli, A. : Feature-based deformable surface detection with self-occlusion reasoning. International Journal of Computer Vision **97**(1), 54–70 (2012)

[97] Poma, X.S., Riba, E., Sappa, A. : Dense extreme inception network : Towards a robust cnn model for edge detection. In : Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 1923–1932 (2020)

[98] Prokopetc, K., Collins, T., Bartoli, A. : Automatic detection of the uterus and fallopian tube junctions in laparoscopic images. In : International conference on information processing in medical imaging, pp. 552–563. Springer (2015)

[99] Puerto-Souza, G.A., Cadeddu, J.A., Mariottini, G.L. : Toward long-term and accurate augmented-reality for monocular endoscopic videos. IEEE Transactions on Biomedical Engineering **61**(10), 2609–2620 (2014)

[100] Ramamonjisoa, M., Lepetit, V. : Sharpnet : Fast and accurate recovery of occluding contours in monocular depth estimation. arXiv (2019)

[101] Ronneberger, O., Fischer, P., Brox, T. : U-net : Convolutional networks for biomedical image segmentation. In : MICCAI 2015, pp. 234–241 (2015)

[102] Rumelhart, D.E., Hinton, G.E., Williams, R.J. : Learning representations by back-propagating errors. nature **323**(6088), 533–536 (1986)

[103] Shelhamer, E., Long, J., Darrell, T. : Fully Convolutional Networks for Semantic Segmentation. arXiv :1605.06211 [cs] (2016). URL http://arxiv.org/abs/1605.06211

[104] Shen, W., Wang, X., Wang, Y., Bai, X., Zhang, Z. : Deepcontour : A deep convolutional feature learned by positive-sharing loss for contour detection. In : CVPR (2015)

[105] Silberman, N., Hoiem, D., Kohli, P., Fergus, R. : Indoor segmentation and support inference from rgbd images. In : European conference on computer vision, pp. 746–760. Springer (2012)

[106] Simonyan, K., Zisserman, A. : Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv :1409.1556 (2014)

[107] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R. : Dropout : a simple way to prevent neural networks from overfitting. The journal of machine learning research **15**(1), 1929–1958 (2014)

[108] Stauder, R., Ostler, D., Kranzfelder, M., Koller, S., Feußner, H., Navab, N. : The TUM lapchole dataset for the M2CAI 2016 workflow challenge. arXiv (2016)

[109] Sturm, P., Triggs, B. : A factorization based algorithm for multi-image projective structure and motion. In : European conference on computer vision, pp. 709–720. Springer (1996)

[110] Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A. : Inception-v4, inception-resnet and the impact of residual connections on learning. In : Proceedings of the AAAI Conference on Artificial Intelligence, vol. 31 (2017)

[111] Takikawa, T., Acuna, D., Jampani, V., Fidler, S. : Gated-scnn : Gated shape cnns for semantic segmentation. In : Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 5229–5238 (2019)

[112] Török, P., Harangi, B. : Digital image analysis with fully connected convolutional neural network to facilitate hysteroscopic fibroid resection. Gynecologic and obstetric investigation **83**(6), 615–619 (2018)

[113] Twinanda, A.P., Shehata, S., Mutter, D., Marescaux, J., de Mathelin, M., Padoy, N. : Endonet : A deep architecture for recognition tasks on laparoscopic videos. IEEE Trans. Med. Imaging **36**(1), 86–97 (2017)

[114] Vevaldi, A. : Universal, unsupervised, and understandable image representations. Medical Imaging Summer School MISS (2018)

[115] de Vos, B.D., Berendsen, F.F., Viergever, M.A., Sokooti, H., Staring, M., Išgum, I. : A deep learning framework for unsupervised affine and deformable image registration. Medical image analysis **52**, 128–143 (2019)

[116] Wang, G., Wang, X., Li, F.W.B., Liang, X. : Doobnet : Deep object occlusion boundary detection from an image. In : ACCV (2018)

[117] Wang, P., Yuille, A.L. : DOC : deep occlusion estimation from a single image. In : ECCV (2016)

[118] Wu, C. : Towards linear-time incremental structure from motion. In : 2013 International Conference on 3D Vision-3DV 2013, pp. 127–134. IEEE (2013)

[119] Wu, J., Westermann, R., Dick, C. : A survey of physically based simulation of cuts in deformable bodies. Comput. Graph. Forum **34**(6), 161–187 (2015)

[120] Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K. : Aggregated residual transformations for deep neural networks. In : Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1492–1500 (2017)

[121] Xie, S., Tu, Z. : Holistically-nested edge detection. IJCV **125**(1-3), 3–18 (2017)

[122] Yang, H., Li, Y., Yan, X., Cao, F. : Contourgan : Image contour detection with generative adversarial network. Knowledge-Based Systems **164**, 21–28 (2019)

[123] Yang, J., Price, B.L., Cohen, S., Lee, H., Yang, M. : Object contour detection with a fully convolutional encoder-decoder network. In : CVPR (2016)

[124] Yang, S., Li, X., Jia, X., Wang, Y., Zhao, H., Lee, J. : Deep learning-based intelligent defect detection of cutting wheels with industrial images in manufacturing. Procedia Manufacturing **48**, 902–907 (2020)

[125] Yu, Z., Feng, C., Liu, M., Ramalingam, S. : Casenet : Deep category-aware semantic edge detection. In : CVPR (2017)

[126] Yu, Z., Liu, W., Zou, Y., Feng, C., Ramalingam, S., Kumar, B.V.K.V., J.Kautz : Simultaneous edge alignment and learning. In : ECCV (2018)

[127] Yuan, Y., Xie, J., Chen, X., Wang, J. : Segfix : Model-agnostic boundary refinement for segmentation. In : European Conference on Computer Vision, pp. 489–506. Springer (2020)

[128] Zadeh, S.M., Francois, T., Calvet, L., Chauvet, P., Canis, M., Bartoli, A., Bourdel, N. : Surgai : deep learning for computerized laparoscopic image understanding in gynaecology. Surgical endoscopy **34**(12), 5377–5383 (2020)

[129] Zhang, R., Isola, P., Efros, A.A. : Colorful image colorization. In : European conference on computer vision, pp. 649–666. Springer (2016)

[130] Zhou, P., Price, B., Cohen, S., Wilensky, G., Davis, L.S. : Deepstrip : High-resolution boundary refinement. In : Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10558–10567 (2020)

[131] Zitnick, L. : The dark ages : A history of object recognition before deep learning. Medical Imaging Summer School 2018 (MISS 2018), Favignana (2018)