Détection et description de points clés par apprentissage

Nicolas Loiseau-Witon

N°d'ordre NNT : 2023ISAL0101

THESE de DOCTORAT DE L'UNIVERSITE DE LYON
opérée au sein de
(Institut National des Sciences Appliquées, INSA - Lyon)

Ecole Doctorale N° 160
(ELECTRONIQUE, ELECTROTECHNIQUE, AUTOMATIQUE)

Spécialité/ discipline de doctorat :
Traitement du Signal et de l'Image

Soutenue publiquement le 10/11/2021, par :
Nicolas Loiseau--Witon

Détection et description de points clés par apprentissage

Devant le jury composé de :

Angelini, Elsa	PU	Université de Paris	Examinatrice
Antonini, Marc	DR	Université de Nice-Sophia	Examinateur
Aubreton, Olivier	MDC/HDR	Université de Bourgogne	Rapporteur
Hétroy, Franck	PU	Université de Strasbourg	Rapporteur
Kéchichian, Razmig	MC	Université de Lyon	co-encadrant
Valette, Sébastien	DR	Université de Lyon	Directeur de thèse

Département FEDORA – INSA Lyon - Ecoles Doctorales

SIGLE	ECOLE DOCTORALE	NOM ET COORDONNEES DU RESPONSABLE
CHIMIE	**CHIMIE DE LYON** **https://www.edchimie-lyon.fr** Sec. : Renée EL MELHEM Bât. Blaise PASCAL, 3e étage secretariat@edchimie-lyon.fr	**M. Stéphane DANIELE** C2P2-CPE LYON-UMR 5265 Bâtiment F308, BP 2077 43 Boulevard du 11 novembre 1918 69616 Villeurbanne **directeur@edchimie-lyon.fr**
E.E.A.	**ÉLECTRONIQUE, ÉLECTROTECHNIQUE, AUTOMATIQUE** **https://edeea.universite-lyon.fr** Sec. : Stéphanie CAUVIN Bâtiment Direction INSA Lyon Tél : 04.72.43.71.70 secretariat.edeea@insa-lyon.fr	**M. Philippe DELACHARTRE** INSA LYON Laboratoire CREATIS Bâtiment Blaise Pascal, 7 avenue Jean Capelle 69621 Villeurbanne CEDEX Tél : 04.72.43.88.63 **philippe.delachartre@insa-lyon.fr**
E2M2	**ÉVOLUTION, ÉCOSYSTÈME, MICROBIOLOGIE, MODÉLISATION** **http://e2m2.universite-lyon.fr** Sec. : Sylvie ROBERJOT Bât. Atrium, UCB Lyon 1 Tél : 04.72.44.83.62 secretariat.e2m2@univ-lyon1.fr	**M. Philippe NORMAND** Université Claude Bernard Lyon 1 UMR 5557 Lab. d'Ecologie Microbienne Bâtiment Mendel 43, boulevard du 11 Novembre 1918 69 622 Villeurbanne CEDEX **philippe.normand@univ-lyon1.fr**
EDISS	**INTERDISCIPLINAIRE SCIENCES-SANTÉ** **http://ediss.universite-lyon.fr** Sec. : Sylvie ROBERJOT Bât. Atrium, UCB Lyon 1 Tél : 04.72.44.83.62 secretariat.ediss@univ-lyon1.fr	**Mme Sylvie RICARD-BLUM** Institut de Chimie et Biochimie Moléculaires et Supramoléculaires (ICBMS) - UMR 5246 CNRS - Université Lyon 1 Bâtiment Raulin - 2ème étage Nord 43 Boulevard du 11 novembre 1918 69622 Villeurbanne Cedex Tél : +33(0)4 72 44 82 32 **sylvie.ricard-blum@univ-lyon1.fr**
INFOMATHS	**INFORMATIQUE ET MATHÉMATIQUES** **http://edinfomaths.universite-lyon.fr** Sec. : Renée EL MELHEM Bât. Blaise PASCAL, 3e étage Tél : 04.72.43.80.46 infomaths@univ-lyon1.fr	**M. Hamamache KHEDDOUCI** Université Claude Bernard Lyon 1 Bât. Nautibus 43, Boulevard du 11 novembre 1918 69 622 Villeurbanne Cedex France Tél : 04.72.44.83.69 **hamamache.kheddouci@univ-lyon1.fr**
Matériaux	**MATÉRIAUX DE LYON** **http://ed34.universite-lyon.fr** Sec. : Yann DE ORDENANA Tél : 04.72.18.62.44 yann.de-ordenana@ec-lyon.fr	**M. Stéphane BENAYOUN** Ecole Centrale de Lyon Laboratoire LTDS 36 avenue Guy de Collongue 69134 Ecully CEDEX Tél : 04.72.18.64.37 **stephane.benayoun@ec-lyon.fr**
MEGA	**MÉCANIQUE, ÉNERGÉTIQUE, GÉNIE CIVIL, ACOUSTIQUE** **http://edmega.universite-lyon.fr** Sec. : Stéphanie CAUVIN Tél : 04.72.43.71.70 Bâtiment Direction INSA Lyon mega@insa-lyon.fr	**M. Jocelyn BONJOUR** INSA Lyon Laboratoire CETHIL Bâtiment Sadi-Carnot 9, rue de la Physique 69621 Villeurbanne CEDEX **jocelyn.bonjour@insa-lyon.fr**
ScSo	**ScSo*** **https://edsciencessociales.universite-lyon.fr** Sec. : Mélina FAVETON INSA : J.Y. TOUSSAINT Tél : 04.78.69.77.79 melina.faveton@univ-lyon2.fr	**M. Christian MONTES** Université Lumière Lyon 2 86 Rue Pasteur 69365 Lyon CEDEX 07 **christian.montes@univ-lyon2.fr**

*ScSo : Histoire, Géographie, Aménagement, Urbanisme, Archéologie, Science politique, Sociologie, Anthropologie

Détection et description de points clés par apprentissage en vue d'un recalage à grande échelle

Résumé : Les hôpitaux génèrent de plus en plus d'images médicales en 3D. Ces volumes nécessitent un recalage automatique, en vue d'être analysés de manière systématique et à grande échelle. Les points clés sont utilisés pour réduire la durée et la mémoire nécessaires à ce recalage et peuvent être détectés et décrits à l'aide de différentes méthodes classiques, mais également à l'aide de réseaux neuronaux, comme cela a été démontré de nombreuses fois en 2D.

Cette thèse présente les résultats et les discussions sur les méthodes de détection et de description de points clés à l'aide de réseaux neuronaux 3D. Deux types de réseaux ont été étudiés pour détecter et/ou décrire des points caractéristiques dans des images médicales 3D. Les premiers réseaux étudiés permettent de décrire les zones entourant directement les points clés, tandis que les seconds effectuent les deux étapes de détection et de description des points clés en une seule fois.

Mots clés : Détection, Description, Point clé, Image médicale 3D, Recalage.

Detection and description of keypoints by learning for large-scale registration

Abstract : Hospitals are increasingly generating 3D medical images that require automatic registration for systematic and large-scale analysis. Key points are used to reduce the time and memory required for this registration, and can be detected and described using various classical methods, as well as neural networks, as demonstrated numerous times in 2D.

This thesis presents results and discussions on methods for detecting and describing key points using 3D neural networks. Two types of networks were studied to detect and/or describe characteristic points in 3D medical images. The first networks studied describe the areas directly surrounding key points, while the second type performs both detection and description of key points in a single step.

Keywords : Detection, Description, Keypoint, 3D medical image, registration.

Univ. de Lyon, CNRS, France

Table des Matières

Table des Figures

Abréviations

2D Bidimensionnel 5–7, 9, 17, 48, 49, 55, 75
3D Tridimensionnel 5, 6, 8, 20, 46, 48, 49, 58, 89
ACP Analyse en composantes principales (angl. «*Principal component analysis*») 17
ALIKE Accurate and lightweight keypoint detection and descriptor extraction 98, 100, 101
AOS Fractionnements additifs d'opérateurs 24
BRIEF Binary Robust Independent Elementary Features . . . xx, 25, 27
BRISK Binary Robust Invariant Scalable Keypoints xx
CPU Processeur (angl. «*central processing unit*») 74
D&DL3D Détection et Description multi-échelle Léger en 3D . 98, 104, 105
DICOM Imagerie numérique et communications en médecine (angl. «*Digital imaging and communications in medicine*») 58
DKD Differentiable keypoint detection 100, 102, 103
DMMA Distance moyenne entre les marqueurs anatomiques 77
DoG Difference de gaussiennes 23, 44, 48
ES Entrée/sorties 80
ESPCN Réseau neuronal convolutionnel efficace sous-pixel (angl. «*Efficient Sub-pixel Convolutional Neural Network*») 89
FGR Fast groupwise registration (fran. «*recalage rapide de groupes*»)13, 14
FPR Fast pairwaise registration (fran. «*recalage rapide par paires*») 13, 14
FPR95 Taux de faux positifs 95 (angl. «*False Positive Rate 95*»)17, 77, 78
FROG Fast registration of image groups13–15, 19, 62, 75, 77, 81, 82, 88, 109
GAN Réseaux antagonistes génératifs (angl. «*Generative adversarial networks*») 82
GPU Unité de traitement graphique (angl. «*Graphics Processing Unit*») 74, 87, 95

IA Intelligence artificielle 28

ICP Iterative Closest Point (fran. «*Algorithme du plus proche voisin itératif*») 11

IRM Imagerie par résonance magnétique . . . . xv, 9, 58, 61, 82, 109

L2DMK Learning to Detect and Match Keypoints with Deep Architectures41

LIFT Learned Invariant Feature Transform . . . . . xviii, 39–41

LoG Laplacien du gaussien 23

mAP Moyenne des précisions sur chaque élément (angl. «*mean average precision*») 71–73

MMA Précision moyenne de l'appariement (angl. «*Mean Matching Accuracy*») 19

MP3D MagicPoint3D 93, 94

Nifti Initiative technologique en informatique de la neuro-imagerie (angl. «*Neuroimaging Informatics Technology Initiative*») 58

NMS Suppression non maximal (angl. «*Non-Maximal Suppression*»)21, 100, 101, 103

ORB Oriented fast and Rotated BRIEF xx

PACS Systèmes d'archivage et de communication d'images (angl. «*Picture Archiving and Communication Systems*») xv

PM Précision moyenne 47, 48

R2D2 Repeatable and Reliable Detector and Descriptor xviii, xxi, 95, 96, 104

RANSAC Consensus sur l'échantillon aléatoire (angl. «*Random Sample Consensus*») 14

ReLU Unité linéaire rectifiée (angl. «*Rectified Linear Unit*»)31, 43, 50, 95

RNC Réseaux de neurones convolutifs (angl. «*Convolutionnal Neural Network*») . . . . . xx, 28, 29, 31, 33, 37, 39, 78

SfM Structure acquise à partir de mouvement (angl. «*Structure from Motion*») 40–42, 55, 56

SGD Descente de gradient stochastique (angl. «*stochastic gradient descent*») 76, 78, 79

SIFT Scale-Invariant Feature Transform xvii, xx, 21, 24–26, 39, 41, 44, 51, 56

SP3D SuperPoint3D 89, 91, 93–95, 103, 104

SURF Speeded Up Robust Features . . . . . xvii, xx, 24, 26, 51, 75, 78

SVM Séparateur à vaste marge (angl. «*Support Vector Machine*») 29

t-SNE T-distributed Stochastic Neighbor Embedding (fran. «*intégration de voisins stochastiques distribués en t*») 17, 77, 81

TAD Taux d'appariement des descripteurs 77, 78
Tanh Tangente hyperbolique 74
TDM Tomodensitométrique (angl. «*CT-Scan*») . . . . xv, 9, 58, 60, 109
TILDE Temporally Invariant Learned DEtector . . . . . xviii, 39–42
UH Unité Hounsfield xv
VGG Very deep convolutional networks for large-scale recognition (fran. «*réseau convolutionnel très profond pour la reconnaissance à grande échelle*») 43, 74, 89, 91, 92, 95

Introduction

La vision par ordinateur est une branche de l'intelligence artificielle qui vise à donner aux machines la capacité de comprendre et d'interpréter les images du monde réel. Elle trouve une application très pertinente dans l'analyse de millions d'images acquises par les hôpitaux chaque année, et stockées dans les systèmes d'archivage et de communication d'images (angl. «*Picture Archiving and Communication Systems*», PACS). Elles pourraient être davantage exploitées dans de nombreuses applications, allant au-delà du «*simple*» examen clinique du patient qui en est fait actuellement. Des tests automatiques pourraient être développés pour analyser systématiquement ces images, permettant un diagnostic préventif de diverses maladies et un suivi plus précis de l'évolution des patients au cours du temps. Ces tests automatiques doivent être réalisés de manière totalement anonyme, afin de garantir la confidentialité des patients.

L'imagerie médicale désigne un ensemble de processus et techniques qui permettent l'acquisition et la restitution d'images d'un corps humain, et donc la visualisation de différentes structures anatomiques (par exemple les organes ou les os). Différentes modalités d'imagerie médicale existent (voir [Guy et ffytche, 2000]), telles que : l'imagerie par résonance magnétique (IRM), l'imagerie tomodensitométrique (angl. «*CT-Scan*», TDM), l'imagerie ultrasonore, etc. Ces différentes méthodes d'imageries sont incontournables dans de nombreuses situations, et constituent donc des champs d'analyse et de recherche importants.

Dans ce manuscrit, nous allons nous concentrer principalement sur l'analyse d'images TDM. Cette méthode d'imagerie utilise des rayons X qui émettent un rayonnement ionisant néfaste pour le corps humain à forte dose. Par exemple, un scanner crânien peut générer une quantité de rayonnement équivalente à celle de l'irradiation naturelle sur une période de six mois. C'est pourquoi, plutôt que de n'utiliser les images TDM qu'une seule fois lors de l'analyse initiale faite par le praticien, une analyse systématique et automatique de ces images par différents algorithmes peut être bénéfique pour le patient. Cette approche systématique peut permettre la détection d'éventuelles pathologies potentiellement non soupçonnées et donc non intentionnellement recherchées par le praticien lors de son analyse des images. Les images TDM utilisent l'unité Hounsfield (UH), qui mesure la capacité d'absorption d'un

faisceau de rayons X par différents tissus ou matériaux. Les valeurs de l'UH sont de -1000 UH pour l'air pur, 0 UH pour l'eau, et $300+$ UH pour les os et le métal. À l'aide de ces différentes intensités, les radiologues peuvent observer les différentes structures du corps humain, en fonction de la pathologie recherchée.

La recherche en imagerie médicale et en vision par ordinateur est donc un domaine en constante évolution, voir [Elyan et al., 2022], qui vise à améliorer la qualité et la précision des images, ainsi qu'à développer de nouvelles méthodes d'analyse et de traitement. Une approche importante dans ce contexte est le recalage d'images ; elle consiste à aligner et à superposer des images de patients provenant de différentes modalités ou acquises à des moments différents, afin de faciliter leur comparaison et leur analyse. Le but de ce processus de recalage d'images est de déterminer un champ de déformation qui, pour chaque point dans une image source, associe les coordonnées dans une image cible. Il est nécessaire que les points appariés entre l'image source et l'image cible représentent une même réalité anatomique. Cette mise en correspondance des images peut être réalisée en comparant chaque pixel des deux images ou en se basant sur des points clés extraits à partir de ces images. Les points clés, également appelés points d'intérêt, sont des zones distinctes et informatives qui facilitent cette mise en correspondance, tout en réduisant le nombre d'informations à comparer.

L'objectif de cette étude est de développer des méthodes pour la détection et la description de points clés dans des images médicales, qui seront ensuite utilisées pour le recalage d'images en imagerie médicale. D'une part, nous nous intéresserons à la description et à l'appariement des points caractéristiques entre deux images, puis d'autre part à la détection de ces points clés (ou points caractéristiques) au sein des images. Ces deux aspects seront réalisés à l'aide d'apprentissage par réseau neuronal, ce qui pose également un nouveau verrou dans le domaine de l'imagerie médicale : le nombre d'images médicales utilisables pour réaliser ces tâches de vision par ordinateur.

Recalage

Selon la thèse de [Roche, 2001], «*le recalage est un processus qui vise à faire correspondre deux images ou plus dans un espace commun*». La recherche des correspondances peut être calculée entre chaque pixel d'une image source et d'une image cible, ou bien à l'aide de points clés extraits de ces deux images. Le recalage peut être appliqué à des images provenant d'un même patient (ce que l'on appelle un recalage intra-individu), ou bien elles peuvent provenir de patients différents (on parlera alors de recalage interindividus). Toujours selon Roche, le recalage intrapatient requiert en général des transformations géométriques simples, utilisant un faible nombre de paramètres, tels que les transformations rigides et affines. Ce recalage provenant d'un même patient est généralement utilisé pour l'analyse de structures anatomiques (comme les hypertrophies ou atrophies), ou encore l'analyse du déplacement respiratoire ou cardiaque. Il permet donc une analyse longitudinale du patient, c'est-à-dire le suivi du sujet au cours du temps, voir [Scahill et al., 2003]. Le recalage interpatient quant à lui, utilise le plus souvent des transformations déformables, dîtes non rigides, qui ont beaucoup plus de degrés de liberté que les transformations rigides, voir [Noblet, 2006]. Ce recalage permet diverses

applications, telles que la segmentation automatique par transport de points clés ([Bralet et al., 2021]), ou l'analyse statistique de variations anatomiques, normales ou pathologiques.

Deux catégories de recalage existent aujourd'hui, le recalage par champ de déformation dense et le recalage parcimonieux, voir [Agier, 2017]. Le recalage dense a pour but de déterminer un champ de déformation pour chaque pixel de l'image source vers l'image cible. Ce type de recalage est très coûteux en termes de mémoire machine et de temps de calcul : il nécessite plusieurs heures de calculs pour recaler 20 volumes de taille $512 \times 512 \times 512$ ([Bhattacharjee et al., 2021]), contrairement au second type de recalage qui nécessitera moins d'une heure pour le même nombre d'images ([Agier, 2017]).

Le recalage parcimonieux est un type de recalage basé sur des points d'intérêt détectés et décrits dans les images à recaler. Ces points clés (ou points d'intérêt) sont définis comme des zones distinctives et informatives qui facilitent la reconnaissance, la comparaison et le suivi des objets ou des scènes. Les points sont ensuite appariés (comme sur la Figure 1) selon la similarité entre leurs descripteurs. Les paires de points clés ainsi obtenues sont utilisées pour le recalage.

L'un des aspects importants du recalage parcimonieux provient de la faible empreinte mémoire nécessaire à l'alignement de dizaines d'images de patients, ainsi que du temps de calcul nécessaire nettement inférieur en comparaison au recalage dense.

Pour qu'un point d'intérêt soit considéré comme répété, il doit être présent dans un certain nombre d'images afin qu'il puisse être identifié et apparié aux mêmes endroits dans les autres images ; les descripteurs de ces points clés doivent également être suffisamment similaires entre eux et distincts des autres descripteurs. Il est donc logique de vouloir réduire le nombre de points d'intérêt pour diminuer l'empreinte mémorielle et temporelle nécessaire pour le traitement des images. Cela signifie qu'il faut supprimer les points peu répétés ou ceux dont les descripteurs ne permettent pas un bon appariement. Plus ces points seront détectés et associés précisément, moins il sera nécessaire d'en détecter un nombre important. C'est précisément l'objectif de ce travail.

Détection de points clés

En premier lieu, lorsque l'on parle de points clés, il faut bien les distinguer des points de repère anatomiques (voir Figure 1 et Figure 2). Les premiers ont pour but de décrire au mieux l'image dont ils sont extraits, ils doivent également permettre une bonne description de leur voisinage. Les seconds, les points de repère anatomiques, sont des points spécialement placés par des professionnels du domaine (ici des praticiens hospitaliers) ; ces points de repère permettent la localisation précise de diverses zones.

La détection de points caractéristiques est un problème majeur dans la vision artificielle par ordinateur. Son application continue à se développer dans de nombreux domaines comme la segmentation d'images (voir [Wachinger et al., 2015]) ou la détection d'objets (voir [Jaiswal et al., 2021]) et nécessite donc une constante amélioration de cette détection de points. Les méthodes de détection peuvent être divisées en deux sous-ensembles : les méthodes classiques (ou traditionnelles) et les méthodes par apprentissage.

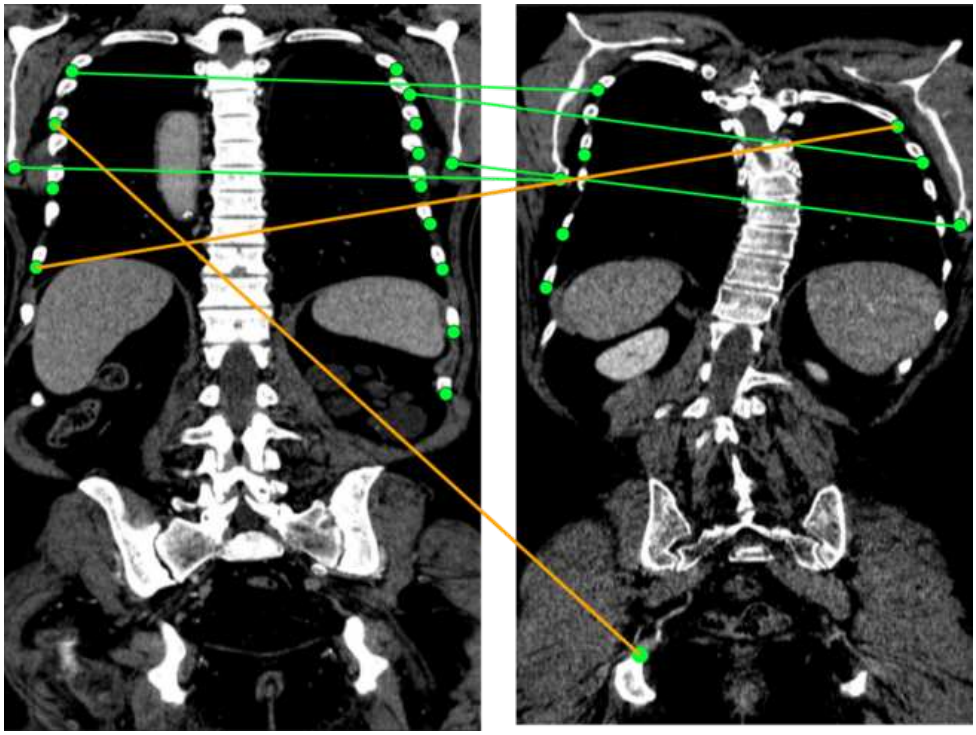


FIGURE 1 – Exemple de points d'intérêt détectés et appairés dans des images médicales 2D, extraites d'images volumiques tomodensitométriques. Les traits verts reliant les deux images représentent les points bien appairés, contrairement aux traits oranges qui représentent des points mal appariés.

Les méthodes de détection classiques ont connu un grand développement depuis les années 90. Parmi les méthodes les plus connues figurent Harris ([Harris et Stephens, 1988]), Scale-Invariant Feature Transform (SIFT) ([Lowe, 2004]) et Speeded Up Robust Features (SURF)([Bay et al., 2006]). La méthode Harris a été introduite en 1988 et utilise la dérivée seconde de la fonction de la luminosité pour trouver les points d'intérêt. La méthode SIFT a été proposée en 1999, elle détecte les points clés en analysant les images à différents niveaux d'échelle. Elle a été suivie par la méthode SURF en 2006, qui utilise les approximations des dérivées secondes de la gaussienne pour accélérer la détection des points clés tout en améliorant la robustesse.

Avec l'avènement des réseaux de neurones profonds, de nouvelles méthodes de détection de points clés ont émergé. Ces méthodes par apprentissage ont la particularité de ne pas nécessiter d'extraction de caractéristiques traditionnelles, mais plutôt de les apprendre à partir des données d'entraînement. Parmi les méthodes les plus connues, on retrouve Temporally Invariant Learned DEtector (TILDE) (voir [Verdie et al., 2014]) qui est un réseau étudié pour détecter des points clés de manière répétable au travers d'images subissant des changements d'illuminations drastiques (comme les changements de saisons ou météorologiques). Une extension en a été proposée, nommée Learned Invariant Feature Transform (LIFT) (voir [Yi et al., 2016]), afin de créer une chaîne de traitement permettant la détection des points clés, leurs orientations et enfin leurs descriptions.

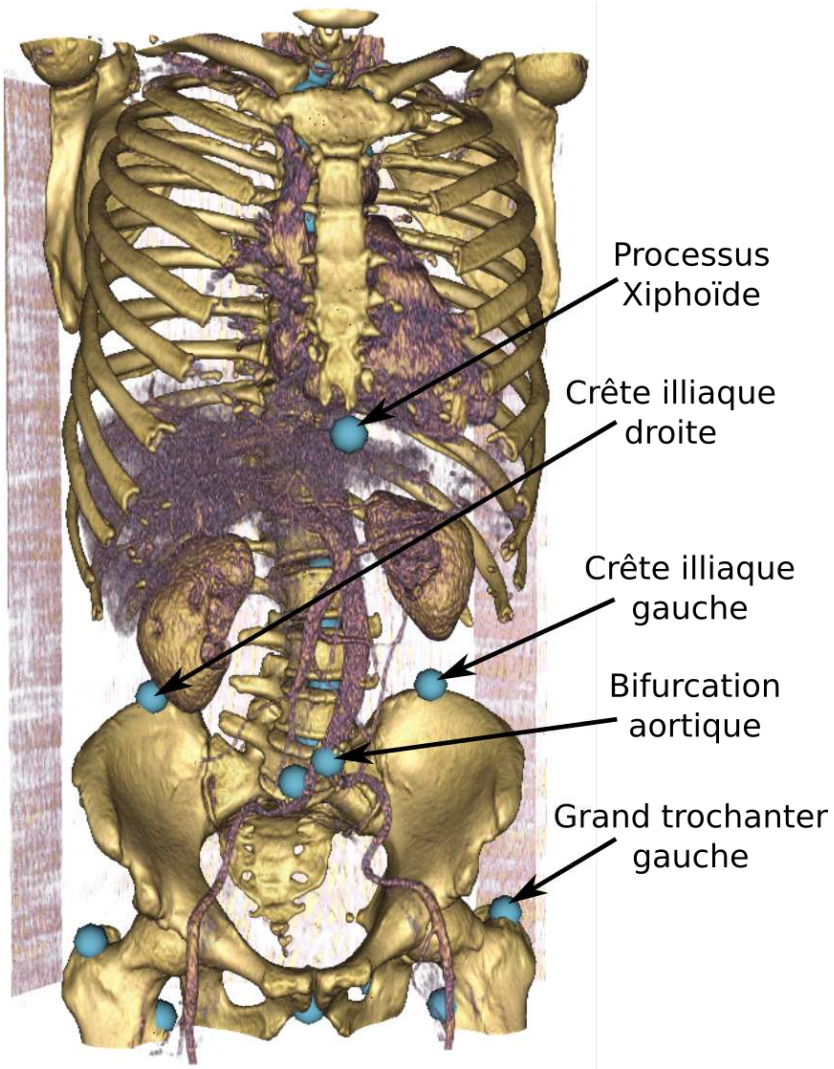


FIGURE 2 – Exemple de points de repère anatomiques placés sur un patient. Chaque point de repère représente un point saillant remarquable et identifiable du corps humain.

En 2016, [Altwaijry et al., 2016] proposait un réseau permettant d'apprendre à détecter des points dans des échelles multiples, au contraire de LIFT qui applique son détecteur à des échelles d'images différentes lors de l'inférence du réseau. Suite à cela, SuperPoint (voir [DeTone et al., 2017]) a vu le jour avec une implémentation effectuant la chaîne de traitement complète de détection et description en un seul entraînement de bout en bout. SuperPoint utilise des images en pleine résolution, et ne nécessite pas d'extraire des parcelles d'images. Enfin, des réseaux tels que D2-Net et Repeatable and Reliable Detector and Descriptor (R2D2) (voir [Dusmanu et al., 2019 ; Revaud, Weinzaepfel et al., 2019]) qui utilisent directement la sortie du réseau encodeur d'image pour détecter et décrire des points clés au sein des images en une seule passe ont vu le jour.

Tous ces réseaux ont fait émerger l'intérêt de l'apprentissage dans cette tâche qu'est la détection de points clés, en obtenant des résultats surpassant largement les méthodes

classiques. C'est pourquoi nous nous sommes tournés vers l'apprentissage d'un détecteur dans le domaine de l'imagerie médicale 3D.

Description et appariement de points clés

La vision par ordinateur et le recalage d'images impliquent la description de caractéristiques et leur appariement. Pour être efficace, un descripteur de points clés doit être résistant aux transformations d'images comme l'échelle, la rotation, le bruit et les variations d'illumination. L'appariement de ces points clés est également une étape cruciale pour des applications de la vision par ordinateur, comme la reconstruction 3D, la reconnaissance d'objets, la structure du mouvement et le suivi d'objets.

Un des descripteurs les plus connus pour la description de points caractéristiques est SIFT (voir [Lowe, 2004]), cependant sa complexité de calcul le rend peu adapté aux applications en temps réel. Une alternative populaire est le descripteur SURF (voir [Bay et al., 2006]), qui utilise une méthode de calcul plus rapide basée sur l'utilisation des ondelettes de Haar dans les images intégrales précalculées ;ces ondelettes fournissent des résultats plus précis, tout en conservant sa robustesse pour la description de caractéristiques.

Le descripteur Binary Robust Independent Elementary Features (BRIEF), proposé par [Calonder et al., 2012], est une alternative proposée à SIFT qui est moins complexe, mais moins robuste aux changements d'orientation. Il est cependant efficace pour les applications en temps réel. Pour remédier à cette sensibilité, [Rublee et al., 2011] ont proposé Oriented fast and Rotated BRIEF (ORB) qui inclut le calcul et la compensation de l'orientation, en vue de mieux représenter les informations des régions des images. Le descripteur Binary Robust Invariant Scalable Keypoints (BRISK) (voir [Leutenegger et al., 2011]) est également une alternative intéressante à BRIEF, basé sur l'identification de la direction caractéristique de chaque élément. Cependant, tous ces descripteurs binaires ont une capacité distinctive limitée, car ils sont construits sur un ensemble de comparaisons d'intensité par paire où chaque point d'échantillonnage représente un seul pixel. Cette limitation ne permet pas de capturer toutes les informations contenues dans une région d'image voir [Calonder et al., 2012]).

À l'instar des détecteurs, des descripteurs par apprentissage utilisant des réseaux de neurones convolutifs (angl. «*Convolutionnal Neural Network*», RNC) ont été proposés. Les premières approches, encore très employées, utilisaient des réseaux siamois, qui nécessitent la formation de paires ou de triplets lors de l'entraînement, voir [Bromley et al., 1993 ; Vijay Kumar B et al., 2015]. La création de sous-ensembles d'éléments peut être très coûteuse en termes de temps et de mémoire, et si elle est mal exécutée, elle peut conduire à de mauvais résultats. Pendant l'entraînement du descripteur, il est important de choisir les paires ou les triplets qui conduisent à une bonne formation, c'est-à-dire ceux qui donnent le résultat le plus élevé pour la fonction de perte, mais cela peut être très coûteux en termes de calculs et donc de temps d'entraînement. Afin de pallier cette limitation, il a été proposé d'optimiser directement la précision de la recherche de correspondances en optimisant le réseau avec des listes d'images plutôt qu'avec de simples paires ou triplets, comme décrit dans les références bibliographiques [Cakir et al., 2019 ; He et al., 2018]. Cette approche consiste à réduire la distance existant

entre une image cible et les images correspondantes au sein d'une liste d'images associée. Ces approches ont évolué vers des réseaux qui exécutent toute la chaîne de traitement de détection et de description, comme Superpoint, D2-Net et R2D2, voir [DeTone et al., 2017 ; Dusmanu et al., 2019 ; Revaud, Weinzaepfel et al., 2019].

Un réseau de description de points clés dans des images médicales 3D est apparu en 2018, voir [Blendowski et Heinrich, 2018]. Ce réseau a pour objectif de créer un descripteur binaire pour caractériser les points clés issus d'images médicales volumétriques. Ce descripteur est basé sur une extension en 3D des architectures d'arbres binaires [Zhang et al., 2017].

Il est remarquable que très peu de réseaux de détection et de description de points clés aient été introduits en 3D, ce qui souligne l'importance méthodologique de la 3D. En effet, l'introduction de la troisième dimension dans les images nécessite des calculs supplémentaires considérables, entraînant des temps de calcul et d'apprentissage très importants. Ceci m'est en lumière un nouveau verrou scientifique majeur.

Objectifs et Contributions

Ce travail de recherche s'inscrit dans la continuité des travaux menés par [Agier, 2017]), qui portaient sur le recalage de groupes d'images médicales 3D par points d'intérêt. Ce manuscrit présente trois contributions à la détection et à la description de points clés dans des images médicales volumiques.

Les deux premières contributions de ce travail ont pour but de se concentrer sur la phase de description des points clés en utilisant un réseau 3D.

En 2D, la plupart des réseaux de détection ou de description de points clés nécessitent l'utilisation de plusieurs centaines de milliers, voire de millions d'images. Cependant, en imagerie médicale 3D, nous n'avons pas accès à une telle quantité d'images en raison de différentes raisons telles que les réglementations régissant leur usage et leur diffusion ainsi que le secret médical qui les entoure (consentement des patients, anonymisation des images ou encore coût financier important). De plus, le temps nécessaire à la lecture d'un grand nombre d'images pendant l'entraînement est un autre aspect important. En effet, le temps de chargement en mémoire d'une image 2D de taille 600^2 est d'environ 5 ms, tandis que pour un volume de taille 400^3, cela prend environ 3 secondes, soit un facteur de 1000. Ce problème est d'autant plus lié aux réseaux de neurones, qui se voient fortement complexifiés et alourdis, en raison de l'ajout d'une dimension supplémentaire.

De plus, lors de l'apprentissage, il est important de prendre en compte le nombre limité de points de repère anatomiques (ou marqueurs anatomiques) disponibles sur un nombre restreint d'images, car leur positionnement par des experts est très chronophage. Par conséquent, ce nombre limité de marqueurs sur un nombre d'images limité ne permet pas de réaliser des entraînements efficaces en comparaison aux points clés qui peuvent être détectés par dizaines de milliers. Toutefois, ces marqueurs anatomiques fournissent des informations précieuses sur les correspondances réelles au sein des images après une mise en correspondance dans un espace commun et seront utilisés en tant que tels.

La troisième contribution se concentre principalement sur l'extension en 3D des réseaux Superpoint et R2D2, qui seront utilisés comme références pour la création d'un nouveau réseau utilisant une fonction de perte optimisant directement la répétabilité entre les points clés détectés. Les performances de chaque réseau seront évaluées selon différents critères et comparées les unes aux autres.

Structure du manuscrit de thèse

La structure de ce manuscrit s'articule ainsi : le Chapitre 1 présentera en premier lieu un rapide aperçu du recalage d'images et des méthodes pour le mettre en place ; puis il se concentrera sur les méthodes d'évaluation des détecteurs et descripteurs de points clés. Finalement, le premier chapitre se focalisera sur l'état de l'art en matière de détection et description de points clés par méthodes classiques (ou traditionnelles), ainsi que les méthodes par apprentissage, en portant un intérêt particulier sur la méthode de SURF 3D et des méthodes 2D par apprentissage.

Le Chapitre 2 exposera les différentes données que nous avons utilisées durant la thèse, ainsi que les méthodes utilisées pour augmenter ces mêmes données. Cette augmentation de données permet d'accroître artificiellement le nombre de données médicales à notre disposition.

Dans les chapitres suivants, nous développerons les deux contributions portées par ce manuscrit : Le Chapitre 3 présentera l'apprentissage de descripteurs de points clés en vue d'une mise en correspondance plus fine de ces points clés, en tant que première contribution. La deuxième contribution dans le Chapitre 4, exposera l'extension 3D de deux réseaux 2D, ainsi qu'un nouveau réseau permettant l'apprentissage direct de la répétabilité des points clés.

1

État de l'art : recalage d'images et détection/description de points clés

Sommaire

Ce premier chapitre recense d'importants éléments dans deux axes majeurs : la mise en correspondance d'images, et les problématiques de détection et de descriptions de points clés. Nous détaillerons également les différentes méthodes classiques bidimensionnelles (2Ds) et tridimensionnelles (3Ds) existantes. Dans la dernière partie de ce chapitre, nous aborderons quelques notions d'apprentissage par réseaux de neurones afin de décrire les méthodes de détection et description de points clés par apprentissage.

Dans la première section, nous allons effectuer une description du domaine du recalage d'images. Après une présentation des différentes méthodes permettant le recalage d'images, nous soulèverons les problématiques associées et les axes de recherche qui permettraient l'amélioration de cette étape de mise en correspondance d'images. Nous évoquerons également le besoin d'une étape de détection et de description des points clés qui soit la plus précise et juste possible.

Dans la seconde section, nous détaillerons la nécessité de ces étapes de détection et description de points clés, ainsi que les méthodes permettant d'analyser leur précision et justesse. Nous analyserons les différentes méthodes d'extractions de points clés en commençant tout d'abord par les méthodes classiques, tant sur des images 2D que des images volumiques 3D.

Dans la dernière section, nous parlerons des réseaux de neurones, puis nous analyserons les différents extracteurs de points clés basés apprentissage. Tout au long de ce manuscrit, nous utiliserons le terme «*extracteur*» pour faire référence à une méthode qui permet à la fois de détecter et de décrire des points clés.

1.1 Images et Mise en correspondance d'images

Une image, selon le dictionnaire Larousse, peut-être vue comme «*la reproduction d'un objet matériel par la photographie ou diverses techniques apparentées, telles que : la radioscopie d'un organe ou la réflexion d'ondes ultrasonores*».

FIGURE 1.1 – Exemple d'assemblage de six images de la même scène (paysage) prise à différents points de vue (ou différents angles). Cet assemblage est effectué à l'aide d'un recalage, afin de créer un panorama.

Dans le domaine informatique, une image correspond à un signal numérique, qui peut-être interprété comme une matrice 2D de valeurs d'intensités (dans le cas d'images en niveaux de gris) ou 3D de couleurs (dans le cas d'images en niveau rouge, vert et bleu) captées par la technique d'imagerie utilisée. Ces valeurs numériques sont appelées pixels dans les images 2D et voxels dans les images volumiques 3D. Chaque image peut être modélisée par l'action d'une fonction f à laquelle on associe des valeurs $f(x, y)$ selon la localisation spatiale ; de même pour les images 3D, nous pouvons ajouter une troisième dimension et associer des valeurs $f(x, y, z)$.

En traitement d'image, le recalage d'images est une technique permettant leur mise en correspondance. Selon [Roche, 2001], le recalage est un problème qui n'est que partiellement résolu et qui «*consiste essentiellement à établir une relation géométrique entre les objets représentés par deux images*». Ce terme de recalage d'images est utilisé le plus souvent comme synonyme des expressions «*mise en correspondance d'images*» ou encore «*alignement d'images*». Cette technique est utilisée dans diverses applications et traitements d'images, comme : la création de panoramas en utilisant plusieurs images, l'exposition longue durée en photo de nuit ou encore la superposition d'images d'une même scène à des heures différentes. Une autre utilisation possible est le recalage d'images médicales (comme [Agier et al., 2016 ; Cootes et al., 2010]), ce qui va nous intéresser plus particulièrement ici. Ces différentes techniques et applications nécessitent l'analyse des images à l'échelle du pixel (2D) et du voxel (3D), ou bien des régions localisées autour de ces points. C'est précisément cette analyse des images qui nous concerne dans ce manuscrit.

Il est important de noter qu'aucun processus d'analyse d'image n'est parfait et que les résultats obtenus peuvent varier en fonction de la méthode utilisée et des types de données sur lesquelles cette méthode est appliquée. Par conséquent, les types de données et d'application déterminent également l'erreur acceptable pour une utilisation précise.

FIGURE 1.2 – Exemple de superposition d'images provenant de la même vue (scène), avec à gauche un ensemble de 5 d'images captées la nuit par le même appareil photo. À droite, on peut voir l'empilement de l'ensemble des 5 images en les moyennant pixel à pixel, afin d'améliorer la luminosité.

1.1.1 Applications du recalage en 2D

Prenons une paire d'images 2D $(\mathcal{I}, \mathcal{J})$, représentant une même scène acquise sous différents angles (ou points de vue) et par différentes techniques. Dans notre cas classique de recalage par paires, $\mathcal{I}$ sera désignée comme l'image requête ou encore image mouvante, et $\mathcal{J}$ comme l'image cible ou image fixe. Le recalage d'images, en général, est un traitement informatique qui consiste à estimer la transformation T qui aligne l'image $\mathcal{I}$ (la source), vers la seconde image $\mathcal{J}$ (l'image fixe), afin de superposer les différentes structures communes à travers les images. Nous allons voir quelques-unes des applications de mise en correspondance d'images.

Panorama - L'assemblage de photos (ou «*Image stitching*» en anglais) est probablement l'application la plus connue du grand public, en ce qui concerne le recalage d'images 2D. Cette combinaison de plusieurs images numériques se recouvrant vise à produire un panorama, dont on peut voir un exemple dans la Figure 1.1. Dans cet exemple, on remarque que les images possèdent un pourcentage de recouvrement, les frontières de ces recouvrements sont marquées par des tracés rouges entre chaque point de vue ; des algorithmes vont utiliser ces recouvrements possibles entre chaque image pour créer une image panoramique.

Superposition d'images - L'empilement d'images, dont on peut voir un exemple dans la Figure 1.2, est une autre des utilisations connaissant un essor important ces dernières années, tout particulièrement en astrophotographie. La superposition de photos permet de calculer la somme ou la moyenne des pixels à travers différentes images, ceci afin d'améliorer la qualité d'une image en réduisant le bruit ou encore d'en augmenter la luminosité. Pour que ces opérations soient réalisables sur les pixels à travers les images, il est important que les images soient recalées et correspondent bien les unes aux autres, ceci afin de corriger les micromouvements que pourrait effectuer le photographe durant l'acquisition des images.

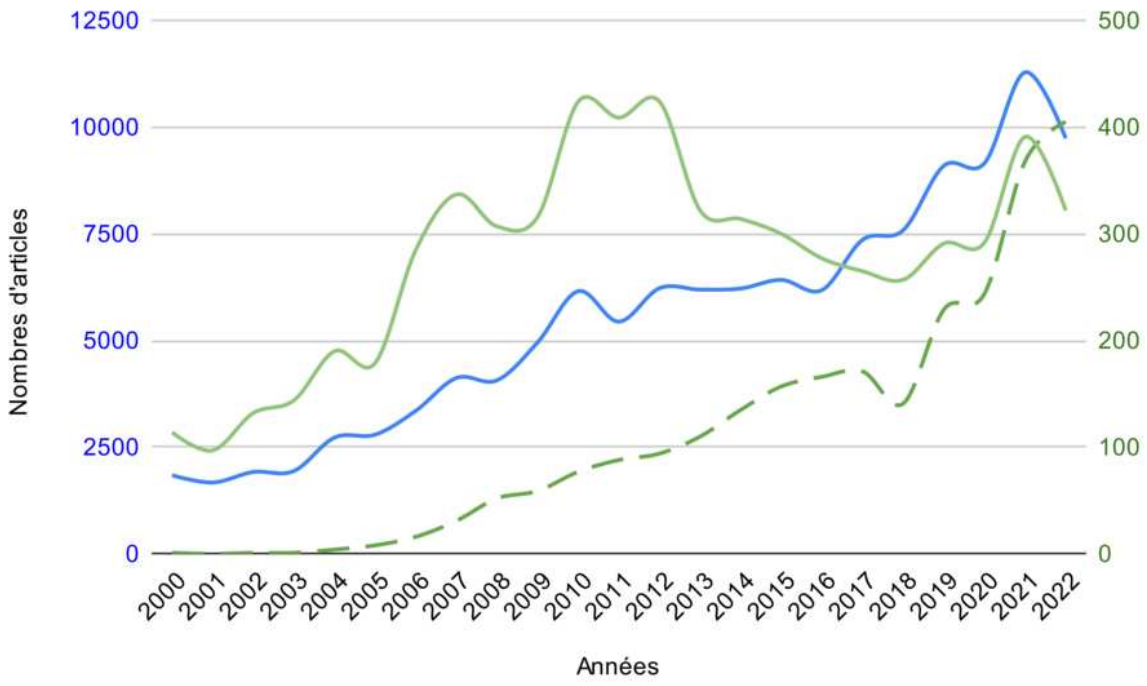


FIGURE 1.3 – Dénombrement des publications parues chaque année de 2000 à 2022 sur le site «*https ://ieeexplore.ieee.org/*» répondant aux critères suivants : «*Medical Images*» en bleu, «*Keypoints*» en vert pointillé et «*Medical Image Registration*» en vert. L'axe des ordonnées représentant le nombre de publications selon la couleur associée.

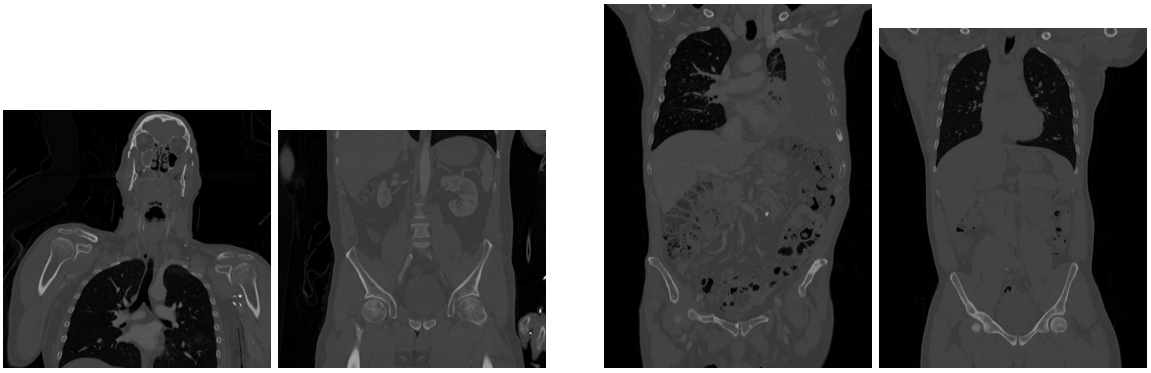
1.1.2 Recalage en imagerie médicale

Dans cette nouvelle section, nous allons voir les besoins du recalage d'images dans le domaine de l'imagerie médicale, puis les transformations géométriques qui peuvent être appliquées à ces images.

L'analyse des données médicales 3D connaît un essor important depuis plusieurs années, comme le montre le graphique de la Figure 1.3. Cet essor peut s'expliquer par le nombre croissant d'images médicales générées tous les ans. Un grand nombre de ces nouvelles approches d'analyses de données médicales se base sur des réseaux de neurones et nécessite donc des bases de données cohérentes et importantes. Afin d'obtenir ces bases de données cohérentes, la mise en correspondance d'images est un outil indispensable permettant d'établir les correspondances spatiales existantes entre deux ou plusieurs images différentes.

Nous allons maintenant analyser les différents liens existant entre les images médicales à mettre en correspondance.

Intra-individu et inter-individu(s) - Un recalage d'images médicales peut se faire au travers d'images acquises chez un même individu, ou bien d'images acquises chez plusieurs patients (voir Figure 1.4).



(a) Paire d'images intra-patient **(b)** Paire d'images inter-patients

FIGURE 1.4 – Coupes 2D extraites de volumes 3D. La paire d'images **(a)** provient d'un même patient, avec des zones imagées différentes. Les deux images **(b)** représentent quant à elles deux patients différents, avec des zones imagées similaires. Images originaires d'une base de données des hôpitaux de Saint-Étienne.

Les applications sont différentes en fonction du mode de mise en correspondance des images. Pour des images intra-individus, les applications seront tournées vers le suivi du patient, telles que l'évolution des masses musculaires, graisseuses ou encore tumorales, ainsi que l'analyse des mouvements pulmonaires ou cardiaques. Concernant les images interindividus, les applications seront plutôt focalisées sur l'analyse de populations, les diagnostics comparatifs entre patients ou encore l'analyse computationnelle d'un groupe d'individus.

Modalités - Il existe plusieurs techniques d'acquisition en imagerie médicale, dont les plus connues sont : l'échographie, l'IRM, la radiographie et enfin celle qui va nous intéresser tout particulièrement, l'imagerie TDM (voir Figure 1.5). Toutes ces modalités nécessitent des critères différents pour effectuer un recalage, et donc des critères différents pour l'extraction et la description des points clés en vue de ces recalages.

1.1.3 Définition du recalage, choix des primitives et optimisation

Définition du recalage

Le recalage d'images vise à déterminer les transformations spatiales qui permettent d'optimiser la correspondance des intensités des images dans l'espace, afin d'obtenir la meilleure concordance possible (voir Figure 1.6).

Reprenons nos deux images 2D $(\mathcal{I}, \mathcal{J})$. Elles représentent une même scène prise sous différents angles et par différentes techniques. Ces deux images peuvent être représentées par deux matrices, où $\mathcal{I}(x, y)$ et $\mathcal{J}(x, y)$ correspondent chacun à leurs valeurs d'intensité respectives.

Selon l'état de l'art dressé par [L. G. Brown, 1992], la correspondance entre ces deux images peut être exprimée selon l'équation :

$$\mathcal{J}(x, y) = g(\mathcal{I}(f(x, y))) \tag{1.1}$$

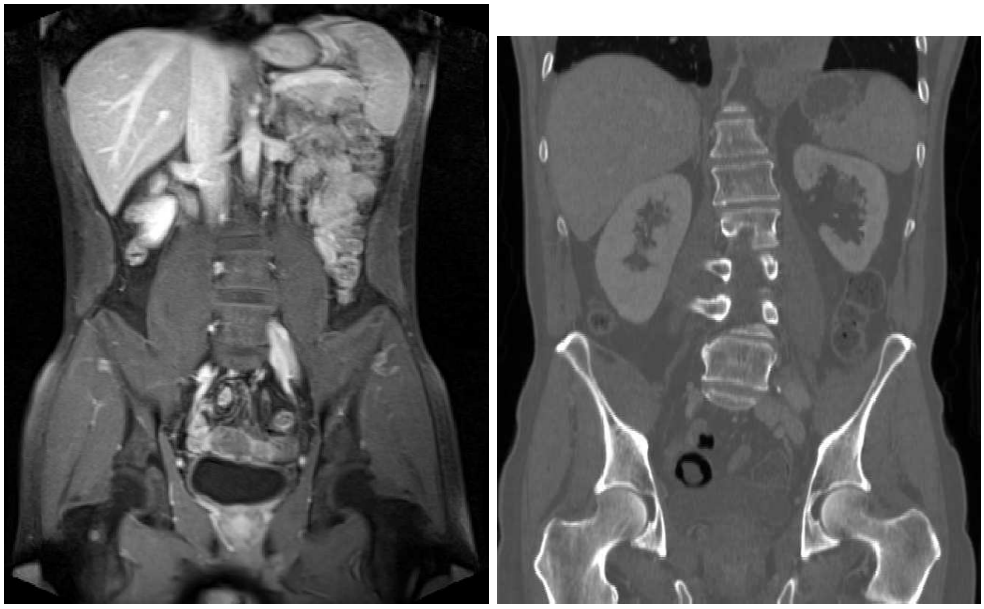


FIGURE 1.5 – Exemple d'image d'IRM (à gauche) et d'image TDM (à droite).

où f est une transformation spatiale bidimensionnelle et g une transformation d'intensité unidimensionnelle. L'équation 1.1 utilise l'image $\mathcal{I}$ comme base de comparaison, appelée «*image de référence*». Le but est d'aligner la seconde image, généralement appelée «*image cible*» ou «*flottante*», sur cette image de référence. Pour ce faire, différentes transformations seront appliquées à l'image source $\mathcal{J}$.

Le recalage revient donc à trouver une transformation optimale $T_{optimale}$ telle que $T(\mathcal{J})$ soit similaire à $\mathcal{I}$ selon un critère de similarité C. Nous pouvons donc écrire l'équation suivante :

$$T_{optimale} = \arg\min_{t \in T} C(\mathcal{I}, t(J)) \tag{1.2}$$

ou $\arg\min$ correspond à une méthode d'optimisation permettant d'obtenir la transformation la plus précise possible.

Critères de mise en correspondance

Selon [Noblet, 2006], il existe deux types d'approches de mise en correspondance d'images : les approches basées en intensités et les approches géométriques.

Approche par intensité - Le recalage basé sur l'intensité (ou encore recalage iconique), repose sur l'exploitation de l'intensité des pixels dans les images. Ces approches sont entièrement automatiques et utilisent l'ensemble des informations au sein des images. Les approches iconiques tentent d'optimiser un critère de ressemblance ou de mesurer la similarité en comparant les intensités localement. Si l'on prend deux intensités représentant exactement le même élément dans les deux images $\mathcal{I}, \mathcal{J}$, elles doivent être alignées aux mêmes coordonnées.

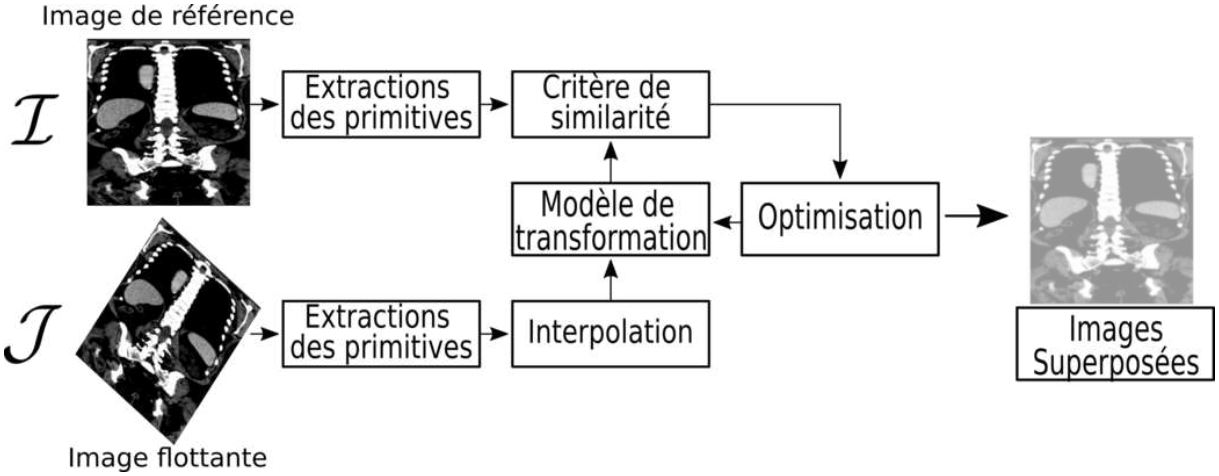


FIGURE 1.6 – Schéma présentant le procédé utilisé lors d'une mise en correspondance d'images. Avec $\mathcal{I}$ (image du haut) l'image de référence sur laquelle on cherche à faire correspondre l'image source $\mathcal{J}$ (image du bas).

Cette approche de correspondance nécessite un grand nombre de calculs et une mise en correspondance pixel par pixel.

Le critère de mise en correspondance, dans ce type d'approche, va dépendre de la relation qui existe entre les différentes images à recaler. Dans le cas d'images monomodales, on peut imaginer que deux localisations similaires sur deux images différentes auront des intensités lumineuses similaires. On peut alors utiliser la somme des carrés des différences d'intensités des images $(\mathcal{I}, t(\mathcal{J}))$, avec $t(\mathcal{J})$ l'image transformée $\mathcal{J}$ par t vers $\mathcal{I}$. En effet, si deux images identiques (deux images représentant le même objet sous un angle différent par exemple) $(\mathcal{I}, t(\mathcal{J}))$ sont correctement alignées, les différences d'intensités $(\mathcal{I}(x, y), t(\mathcal{J})(x', y'))$ de chaque paire de pixels seront nulles. Malheureusement, les clichés pouvant être pris à des périodes différentes, l'intensité lumineuse est rarement la même. On utilisera plutôt une relation affine dont le critère optimal est la corrélation croisée.

Approches géométriques - Les méthodes géométriques (voir Figure 1.7) se basent quant à elles, sur l'extraction de caractéristiques dans chacune des images. Ces caractéristiques, aussi appelées primitives, peuvent être de différentes natures telles que : des points, des surfaces, des coins, des blobs, etc. De manière informelle, un blob est une région d'une image dans laquelle certaines propriétés sont constantes ou approximativement constantes. La méthode la plus courante pour la détection des blobs est la convolution. Il faut ensuite décrire au mieux ces primitives afin de mettre en correspondance deux primitives similaires dans deux images distinctes. Ces éléments extraits peuvent être intrinsèques à l'objet d'étude (le patient par exemple) ou bien extrinsèques (la table d'opération). Les caractéristiques se distinguent encore en deux catégories : celles qui sont apposées manuellement par un expert du domaine, que l'on nommera points de repère (ou «*landmarks*» en anglais), et celles qui sont déterminées par un algorithme automatique.

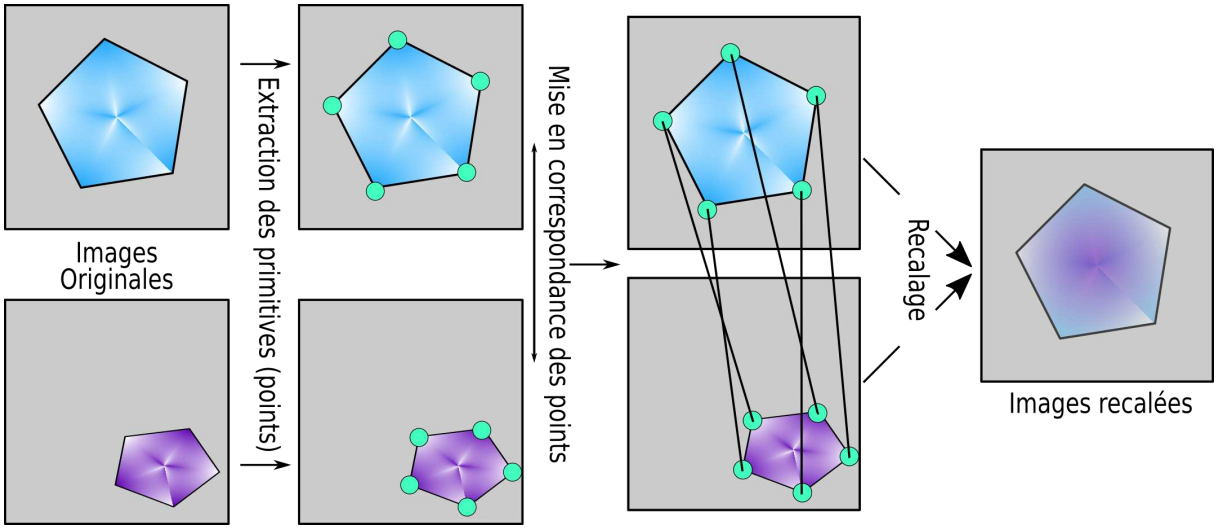


FIGURE 1.7 – Schéma représentant l'approche mise en œuvre par les méthodes géométriques. Des primitives sont détectées et décrites, puis mises en correspondance en vue de recaler ces images.

Il existe ici un certain nombre de critères de mise en correspondance des primitives. Les méthodes ne se basant que sur les primitives extraites des images, telles que : la méthode des moindres carrés [Sammut et Webb, 2010] ou l'algorithme Iterative Closest Point (fran. «*Algorithme du plus proche voisin itératif*», ICP) [Borgefors, 1986]. Une autre possibilité est d'associer un descripteur à chaque primitive extraite [Mikolajczyk et Schmid, 2005]. Ce descripteur aura pour vocation de décrire le plus précisément possible la primitive extraite ou la zone entourant cette primitive.

1.1.4 Transformations affines, rigides, déformables et notations

Les transformations [Hartley et Zisserman, 2004a] affines, rigides et déformables sont des bijections couramment utilisées pour aligner ou superposer des images médicales. Chacune de ces transformations a des avantages et des inconvénients en fonction de l'application et des images sur lesquelles elles sont mises en œuvre.

Les transformations rigides (voir Figure 1.8) sont les plus simples à utiliser et les plus rapides à calculer, elles sont une version simplifiée des transformations affines que nous verrons par la suite. Elles consistent à appliquer une combinaison de translations, rotations et miroirs à une image. La translation correspond à deux degrés de liberté, soit le mouvement horizontal et vertical, tandis que la rotation représente un degré de liberté avec l'angle de rotation. Les transformations rigides imposent la conservation des rapports de longueur entre deux points de l'image initiale.

Les transformations affines permettent des opérations similaires aux transformations rigides, en ajoutant les combinaisons de mise à l'échelle et de cisaillement. Ces deux types de transformations sont généralement utilisées pour aligner des images montrant le même point de vue, ou encore des images intrapatients.

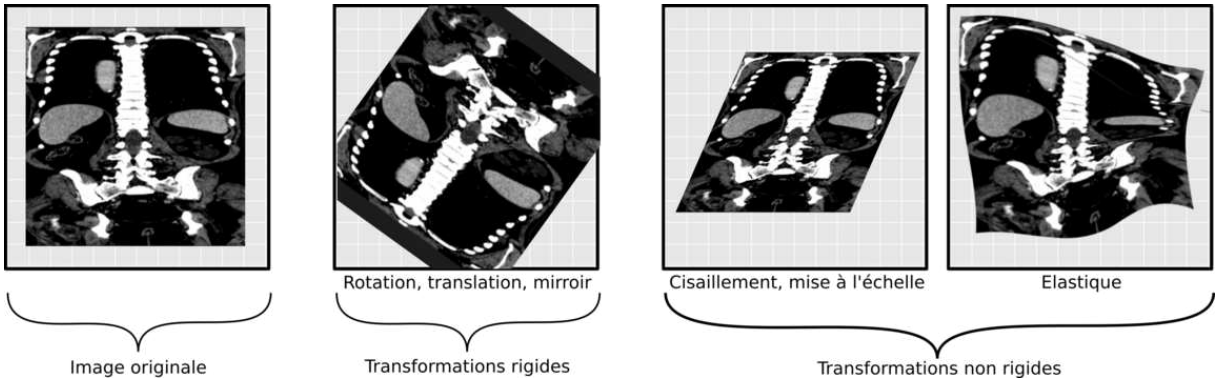


FIGURE 1.8 – Exemple des différentes transformations rigides, affines et déformables applicables.

Les transformations déformables sont les plus complexes à utiliser et les plus lentes à calculer, due au fait qu'elles cherchent à calculer un champ de déformation non linéaire possédant un grand nombre de degrés de libertés; elles regroupent un certain nombre de modèles de fonctions [de Boor, 2001 ; Zagorchev et Goshtasby, 2006]. Ces transformations prennent en compte les distorsions de l'image et les variations de la densité des tissus, ce qui permet d'obtenir un alignement très précis des images. Cependant, ces transformations sont plus complexes à mettre en place et prennent également plus de temps à exécuter. Ce type de transformation est principalement utilisé pour le recalage interpatient, en vue d'aligner les différentes sections du corps humain entre elles.

Il est important de noter que le choix de la transformation utilisée dépendra des images à aligner et de l'application pour laquelle elles sont utilisées. Les transformations rigides et affines sont généralement suffisantes pour des applications de visualisation, tandis que les transformations non rigides sont nécessaires pour des applications de quantification.

1.1.5 Recalage par paires et recalage de groupes

Il existe un grand nombre de méthodes de recalages, voir [Fu et al., 2020 ; Maintz et Viergever, 1998 ; Sotiras et al., 2013]. Nous ne détaillerons ici que les méthodes de recalages décrites dans [Agier, 2017], car ce sont celles que nous utiliserons dans la suite de ce manuscrit. Une première méthode de recalage, fast pairwaise registration (fran. «*recalage rapide par paires*», FPR), consiste en la mise en correspondance rigide de paires d'images à l'aide de points clés extraits. La seconde méthode, nommée fast groupwise registration (fran. «*recalage rapide de groupes*», FGR), est basée sur la précédente et sert à recaler des groupes d'images. Enfin, nous décrirons une méthode que nous utiliserons tout au long de ce manuscrit : fast registration of image groups (FROG). Nous avons utilisé cette méthode spécifiquement, car selon R. Agier elle donne d'excellents résultats en un temps réduit tout en étant basée sur des points clés, ce qui va nous permettre d'évaluer ces derniers.

Fast pairwise registration - Cette première méthode de recalage permet la mise en corres-

pondance rigide de deux images. Elle est basée sur l'utilisation de points d'intérêts extraits par l'algorithme SURF (que nous détaillerons dans la suite de cet état de l'art) au sein de cette paire d'images. L'algorithme FPR est découpé en plusieurs grandes phases (voir Figure 1.9(a)) :

- Extraction des points caractéristiques à l'aide du détecteur et descripteur SURF.
- Mise en correspondance des points clés extraits et formation de paires de points.
- Estimation de la matrice de transformation à l'aide de l'algorithme consensus sur l'échantillon aléatoire (angl. «*Random Sample Consensus*», RANSAC), [Fischler et Bolles, 1981].

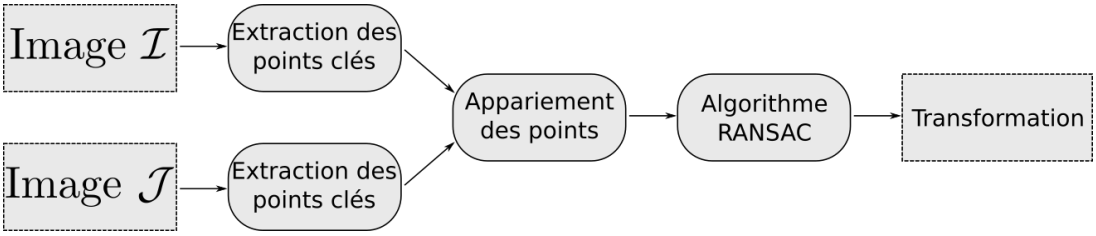
RANSAC est une technique qui permet d'évaluer les paramètres d'un modèle mathématique tout en étant résistant à la présence de données aberrantes (appelées «*outliers*», par opposition aux données «*inliers*»). Cette approche itérative non déterministe n'offre pas de garanties d'optimalité pour le résultat, mais plutôt une probabilité d'optimalité qui augmente avec le nombre d'itérations.

Cette première méthode permet la mise en correspondance de paires d'images interpatientes, de manière efficace, fiable et rapide. Cependant, cette technique ne permet pas de mettre en correspondance des images sans zones communes de recouvrement.

Fast groupwise registration - Cette méthode permet le recalage de groupes d'images, plutôt que simplement des paires d'images comme dans la méthode FPR (voir Figure 1.9(b)). L'algorithme FGR est étudié, selon [Agier et al., 2016], pour effectuer un recalage «*sans référence par graphe complet*», c'est-à-dire qu'il consiste à calculer les $\frac{n(n-1)}{2}$ mises en correspondance possibles entre les n images présentées à l'algorithme. Cette méthode se découpe en 5 étapes :

- Extraction de points d'intérêts par l'algorithme SURF.
- Mise en correspondance de ces points caractéristiques entre toutes les paires d'images possibles, c'est-à-dire $\frac{n(n-1)}{2}$ opérations.
- Estimation des paramètres des transformations avec l'algorithme RANSAC pour chaque paire d'images.
- Élagage du graphe des mises en correspondance en ôtant les arêtes du graphe selon un critère sur le nombre d'«*inliers*» de l'algorithme RANSAC. Les données dont la distance est inférieure à un intervalle de confiance donné sont considérées comme inliers, les autres comme outliers.
- Calcul des positions des images par résolution des équations laplaciennes.

Fast registration of image groups - FROG est le nom de la technique mise en place par [Agier, 2017 ; Agier et al., 2020] permettant la mise en correspondance d'un ensemble d'images à l'aide de transformations déformables, contrairement à la technique FGR qui permet la mise en correspondance rigide de grands groupes d'images. Comme les deux méthodes précédentes, FROG utilise SURF comme extracteur de points clés et des descripteurs associés.



(a) Algorithme mis en place dans la méthode FPR.

(b) Algorithme mis en place dans la méthode FGR.

FIGURE 1.9 – Schémas représentants les algorithmes : FPR (Figure 1.9(a)) et FGR (Figure 1.9(b)).

L'algorithme va ensuite apparier les points clés et utiliser ces appariements entre toutes les paires d'images afin de calculer un ensemble de demi-transformées entre l'espace local de chaque image et un espace commun abstrait. Contrairement à l'algorithme FGR, FROG cherche à recaler les images dans un espace commun en utilisant des transformées déformables grâce aux B-Splines cubiques.

1.1.6 Conclusion sur le recalage

Le but de cette thèse n'est pas d'améliorer les méthodes de recalage, mais se concentre sur l'étape qui précède directement le recalage : l'étape de détection et description des points clés, c'est-à-dire l'extraction des points clés. C'est pourquoi nous utiliserons FROG pour effectuer nos comparaisons.

1.2 Méthodes d'évaluation des détecteurs et descripteurs

Comme nous l'avons vu précédemment, la détection et la description des points d'intérêts permettent, après une mise en correspondance entre les images, d'obtenir une information géométrique de distance. Ceci va permettre de réaliser un recalage cohérent et très rapide en comparaison des méthodes denses.

Mais comment définir un bon point clé ? Lesquels sont les plus pertinents ? De même, comment définir un bon descripteur ?

L'analyse des méthodes de détections et de descriptions de points est un vaste champ de recherche [Cuiyin et al., 2021 ; Hassaballah et al., 2019 ; Krig, 2014 ; Mikolajczyk et Schmid, 2005 ; Schmid et al., 1998 ; Schmid et al., 2000].

Les méthodes d'analyse des extracteurs sont nombreuses et peuvent prendre différentes formes : une analyse visuelle, une analyse sur le temps de calcul et des analyses

sur différentes métriques.

Dans cette nouvelle partie, nous analyserons les méthodes d'évaluation des détecteurs, des descripteurs de points clés et également les méthodes évaluant les extracteurs. Finalement, nous examinerons les résultats de certains extracteurs de points clés.

1.2.1 Évaluation des détecteurs

À présent, nous allons examiner diverses méthodes permettant d'évaluer les points clés extraits par un détecteur. Afin de décrire ces différentes méthodes, nous utilisons les ensembles de points et de descripteurs extraits des images I_1 et I_2 pour former respectivement les ensembles P_1 et D_1, ainsi que P_2 et D_2. La correspondance exacte entre chacun des pixels des deux images est supposée connue, avec $T(P_1) = P_2$, T étant une transformation connue entre les deux images.

Répétabilité - Une première méthode, décrite dans [Schmid et al., 1998 ; Schmid et al., 2000], nécessite de connaître les correspondances exactes de chaque pixel entre deux images. Un point p_1 de l'ensemble P_1 est considéré comme répété, si le point correspondant est aussi détecté dans l'image I_2.

Le pourcentage de points détectés et répétés indique le taux de répétabilité. Généralement, un point p_1 détecté dans I_1 n'est pas détecté exactement à la même position dans I_2, mais plutôt dans un voisinage de cette position. La taille de ce voisinage est désignée par ϵ et le seuil de répétabilité à l'intérieur de ce voisinage est appelée $\epsilon_{\text{repetabilite}}$.

L'ensemble de paires de points (P_1, P_2) qui correspondent dans les deux images, sous le seuil $\epsilon_{\text{repetabilite}}$, est défini par $C(\epsilon) = \{(P_1, P_2) \mid dist(P_1, T(P_2)) < \epsilon\}$. Le taux de répétabilité $r(\epsilon)$ pour deux images I_1, I_2 s'écrit :

$$r(\epsilon) = \frac{|D(\epsilon)|}{\min(n_1, n_2)} \tag{1.3}$$

Où $n_1 = |P_1|$ et $n_2 = |P_2|$, le résultat étant compris entre 0 et 1.

1.2.2 Évaluation des descripteurs

Dans cette nouvelle sous-partie, nous allons examiner diverses méthodes d'évaluation des descripteurs extraits autour des points clés.

Rappel, précision et moyenne des meilleurs appariements - Ces trois métriques, décrites dans [Hassaballah et al., 2019], font partie des métriques standards pour évaluer la performance des descripteurs.

Le taux de rappel (ou «*Recall*» en anglais) mesure le nombre de correspondances correctes entre deux ensembles de données, exprimé comme le ratio entre le nombre de

correspondances correctes et le nombre total de correspondances, selon l'équation suivante :

$$\text{Rappel} = \frac{|\text{Appariements corrects}|}{|\text{Correspondances}|} \tag{1.4}$$

La précision mesure la capacité d'un système à trouver des correspondances correctes entre deux ensembles de données, exprimée comme le ratio entre le nombre de correspondances correctes et le nombre total de correspondances proposées, selon l'équation suivante :

$$\text{Précision} = \frac{|\text{Appariements corrects}|}{|\text{Appariements supposés}|} \tag{1.5}$$

Le nombre moyen d'appariements corrects (ou «*average number of best matches*») correspond au nombre total d'appariements corrects dans chaque image, sur le nombre total d'images, selon l'équation suivante :

$$\text{NbMoyenAppariementsCorrects} = \frac{|\text{Appariements corrects}|}{|\text{Images}|} \tag{1.6}$$

Où $|.|$ correspond à la cardinalité de l'ensemble.

FPR95 - La métrique appelée Taux de faux positifs 95 (angl. «*False Positive Rate 95*», FPR95), définie dans la référence [M. Brown et al., 2010], permet d'évaluer la performance d'un système. Elle mesure le taux de faux positifs lorsque le rappel des vrais positifs atteint 0.95. Pour la calculer, on utilise les distances entre les descripteurs de n paires de points clés sélectionnés aléatoirement, dont la moitié correspond à des paires correspondantes et l'autre moitié à des paires non correspondantes. Si la FPR95 est faible, cela indique que le système obtient de bons résultats.

Analyse en composantes principales (PCA) et intégration stochastique de voisinage distribuée en T (t-SNE) - Les deux techniques [Abdi et Williams, 2010 ; van der Maaten et Hinton, 2008] permettent une visualisation simple et efficace des descripteurs de points clés, dans un espace 2D. Ces deux techniques sont utilisées pour réduire la dimensionnalité de données. Cette réduction de dimension permet de projeter un ensemble de données de haute dimension $d - Dimensions$ vers un ensemble plus simple à $n - dimensions$, où $n < d$ (voir Figure 1.10).

La technique de réduction de dimensions appelée Analyse en composantes principales (angl. «*Principal component analysis*», ACP) est à la fois linéaire et non supervisée. Elle vise à réduire la dimensionnalité d'un ensemble de données fortement corrélées en transformant le vecteur original en un nouvel ensemble appelé composantes principales. Cette réduction se fait en trois étapes distinctes :

- Normalisation des données et calcul de la matrice de covariance.
- Calcul du vecteur propre et de la valeur propre.
- Pour réduire d'une dimension d d'un vecteur à une dimension n, les n plus importantes valeurs propres sont sélectionnées.

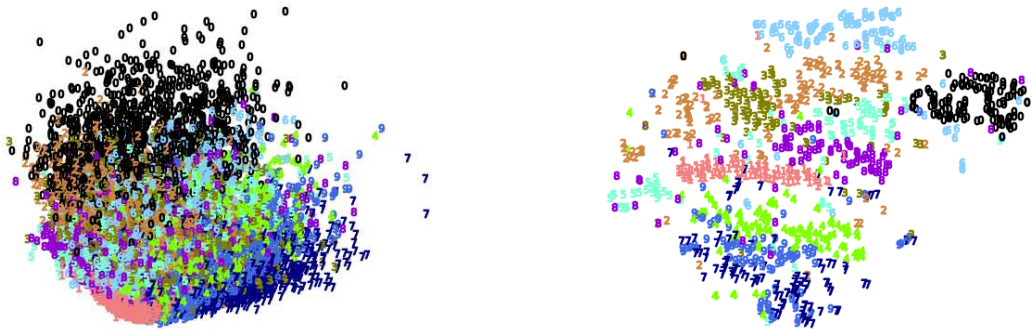


FIGURE 1.10 – Visualisation de l'ensemble des données de mnist par l'analyse en composantes principales (à gauche) et l'intégration stochastique de voisinage distribuée (à droite). Chaque couleur représente un chiffre différent allant de 0 à 9.

L'algorithme t-distributed Stochastic Neighbor Embedding (fran. «*intégration de voisins stochastiques distribués en t*», t-SNE) permet également de réduire les dimensions en créant une distribution de probabilité qui reflète les similitudes entre les voisins dans un espace de grande dimension d et de dimension plus réduite n. L'algorithme est découpé en 3 étapes :

- Il calcule les similarités entre les points dans l'espace de grande dimension en utilisant une fonction de similarité gaussienne.
- Il crée une distribution de probabilité pour les similarités entre les points dans l'espace de petite dimension.
- Il utilise une méthode d'optimisation pour minimiser la différence entre les deux distributions de probabilité.

L'algorithme va ensuite répéter ces différentes étapes jusqu'à ce que les distributions de probabilité convergent vers une solution stable. Finalement, l'algorithme projette les données de grande dimension dans l'espace de plus petite dimension en utilisant les distributions de probabilité optimisées.

1.2.3 Évaluation globale des extracteurs

Il existe différentes méthodes pour obtenir les appariements entre les descripteurs, l'une des plus connues étant la distance du plus proche voisin («*nearest neighbor distance ratio*» en anglais (NNDR)).

Pour chaque descripteur d'une image I_1, les deux descripteurs les plus proches dans l'image I_2 sont définis en tant que les deux meilleures correspondances, puis le rapport de distance entre eux est calculé. Si le ratio de distance est inférieur à une valeur de seuil spécifique, alors la meilleure correspondance est sélectionnée, sinon les deux sont rejetées.

Selon [Hassaballah et al., 2019], «*Une valeur de seuil de* $0,8$ *élimine 90% des correspondances fausses tout en rejetant moins de 5% des bonnes correspondances.*».

M-Score - [DeTone et al., 2017 ; Yi et al., 2016] définissent cette métrique comme une mesure de la performance globale du détecteur et du descripteur de points d'intérêts combinés. Elle calcule le rapport entre les correspondances exactes récupérables par l'ensemble du processus et le nombre de points caractéristiques proposés par le processus dans des régions d'images similaires. Cette métrique est calculée de manière symétrique pour la paire d'images et est ensuite moyennée.

Précision moyenne de l'appariement (ou «*Mean Matching Accuracy*» (MMA)) - Cette métrique, définie par [Dusmanu et al., 2019], où la précision de l'appariement est la moyenne du pourcentage d'appariements corrects dans une paire d'images, en considérant plusieurs seuils d'erreur en pixels. Le score précision moyenne de l'appariement (angl. «*Mean Matching Accuracy*», MMA), est généralement calculé pour un seuil de 3 pixels.

Estimation de l'homographie - Cette mesure permet de définir la capacité d'un algorithme à estimer l'homographie reliant une paire d'images en comparant l'homographie estimée $\widehat{\mathcal{H}}$ à l'homographie réelle $\mathcal{H}$. Il n'est pas facile de comparer directement les matrices $\mathcal{H}$ de 3 x 3, car les différentes entrées de la matrice ont des échelles différentes. Cette mesure compare donc la qualité de la transformation de l'homographie en utilisant les quatre coins d'une image pour l'appliquer à l'autre image. Les quatre coins de la première image sont définis comme c_1, c_2, c_3 et c_4. L'homographie réelle $\mathcal{H}$ est ensuite appliquée pour obtenir les coins de la première image dans la seconde image c'_1, c'_2, c'_3, c'_4, et l'homographie estimée $\widehat{\mathcal{H}}$ pour obtenir les coins estimés dans la deuxième image $\hat{c'_1}$, $\hat{c'_2}$, $\hat{c'_3}$, $\hat{c'_4}$. Un seuil ϵ est ensuite utilisé pour définir une homographie correcte.

1.2.4 Évaluation des extracteurs par le recalage

Pour calculer cette mesure, il est nécessaire d'extraire les points clés en détectant et décrivant ces derniers, ainsi que d'utiliser FROG pour faire correspondre les différentes images. Une fois les images recalées dans un espace commun, la moyenne des distances entre chaque point de repère anatomique, avec le même label (correspondant à la même zone dans plusieurs images), peut être calculée. Cette méthode permet d'évaluer la qualité du recalage et donc de l'extracteur utilisé. Lorsque l'extracteur SURF est utilisé, cette métrique donne une valeur moyenne de 8 mm pour 20 images contenant chacune 40 points de repère anatomiques. Ces valeurs sont mesurées sur l'ensemble de données VISCERAL (voir [Krenn et al., 2017]). Une valeur petite indique un ajustement de meilleure qualité.

FIGURE 1.11 – Exemple de détection de points par le détecteur de Harris.

1.3 Détection et description de points clés par méthodes classiques

Selon [Krig, 2014], «*un détecteur de points est construit pour une classe de points d'intérêts spécifique. Le détecteur doit donc être paramétré en conséquence pour permettre un filtrage des résultats, en vue d'obtenir un ensemble de points candidats pour une caractéristique spécifique*». Il faut également tenir compte du descripteur qui lui sera associé, car chaque détecteur de points d'intérêts fonctionnera mieux avec un descripteur cohérent, selon les caractéristiques spécifiées.

Dans la suite de cette partie, nous nous intéresserons tout d'abord à la définition d'un point clé, dont nous verrons quelques méthodes permettant de les extraire. Puis, nous aborderons l'étape de description des points clés précédemment extraits, à travers l'analyse de quelques méthodes.

1.3.1 Détection de points clés

Nous l'avons vu dans la partie précédente, il convient d'extraire des primitives dans les images en vue d'un recalage d'images. Ces primitives peuvent varier selon le détecteur et prendre différentes formes, telles que : des points, des courbes, des surfaces, etc. En général, une bonne primitive doit-être simple à détecter et idéalement rapide à calculer ; c'est pourquoi se tourner vers les points clés (exemple Figure 1.11) (encore nommés «*points d'intérêts*» ou «*points caractéristiques*») semble être un bon compromis. Un point est donc défini comme d'intérêt, s'il qualifie une zone dans laquelle il est possible d'extraire un descripteur discriminant des autres points de l'image.

Les points clés extraits dans des images médicales 3D sont utilisés dans diverses applications telles que : le recalage d'image comme nous l'avons vu précédemment ou la segmentation d'images [Kéchichian et al., 2014].

D'après [Schmid et al., 1998], il existe trois catégories de détecteurs pour extraire les primitives :

- Les détecteurs basés sur les contours : dans ce type de méthode, on va tout d'abord chercher à localiser les contours dans l'image. Suite à cela, le détecteur va extraire les

coordonnées où un changement brutal de l'intensité lumineuse apparaît. Ce sont ces changements brutaux qui vont indiquer qu'un élément important doit être caractérisé. La méthode [Horaud et al., 1990] extrait des segments de lignes dans les contours de l'image, puis ces segments sont associés et les points sont extraits aux intersections de ces segments.

- Les détecteurs basés sur l'intensité : ces méthodes calculent une mesure qui permet d'indiquer si un point est d'intérêt en se basant directement sur le signal d'intensité. Le détecteur de Harris, [Harris et Stephens, 1988], est probablement le plus connu des détecteurs de cette catégorie. Il utilise une fonction d'autocorrélation du signal en analysant les changements du signal dans plusieurs directions. SIFT ([Lowe, 2004]) quant à lui, utilise des différences de gaussiennes pour extraire les points clés.
- Les détecteurs utilisant des modèles paramétriques : Selon [Schmid et al., 1998], ces détecteurs permettent une grande précision, mais sont limités à des types spécifiques de points d'intérêts, comme les coins en L.

Lors de la détection des points clé, les différents algorithmes peuvent extraires un très grand nombre de points clé, c'est pourquoi une méthode nommée suppression non maximal (angl. «*Non-Maximal Suppression*», NMS) est utilisée afin de filtrer la prédiction du détecteur.

Il existe un grand nombre de détecteurs de points clés, que nous diviserons en deux catégories : les méthodes traditionnelles et les méthodes par apprentissages. Dans la prochaine section, nous présenterons les détecteurs les plus utilisés dans ces deux catégories.

1.3.2 Description de différents détecteurs mathématiques

Nous allons décrire quatre des détecteurs classiques les plus connus, Harris [Harris et Stephens, 1988], SIFT [Lowe, 2004], SURF [Bay et al., 2006] et KAZE [Fernández Alcantarilla et al., 2012].

Détecteur de Harris - Cet algorithme utilise la matrice d'autocorrélation des dérivées, pour détecter les régions de l'image qui ont une forte variation de gradient, comme les coins et les bords.

Pour détecter les points clés, le détecteur de Harris calcule d'abord les dérivées de l'image dans les directions horizontales et verticales. Il calcule ensuite la matrice de corrélation des dérivées. Cette matrice est ensuite utilisée pour calculer un score pour chaque pixel de l'image, appelé le score de Harris. Les pixels qui ont des scores élevés sont considérés comme des points clés.

En détail, l'algorithme va chercher à déterminer les changements d'intensité dans le voisinage immédiat (ou fenêtre) d'un point de coordonnée (x, y), lorsque ce voisinage se déplace dans diverses directions. Considérons la fonction :

$$E(u,v) = \sum_{x,y} \underbrace{w(x,y)}_{\text{fenêtre rectangulaire}} \underbrace{[I(x+u,y+v)}_{\text{décalage d'intensité}} - \underbrace{I(x,y)}_{\text{intensité}}]^2 \tag{1.7}$$

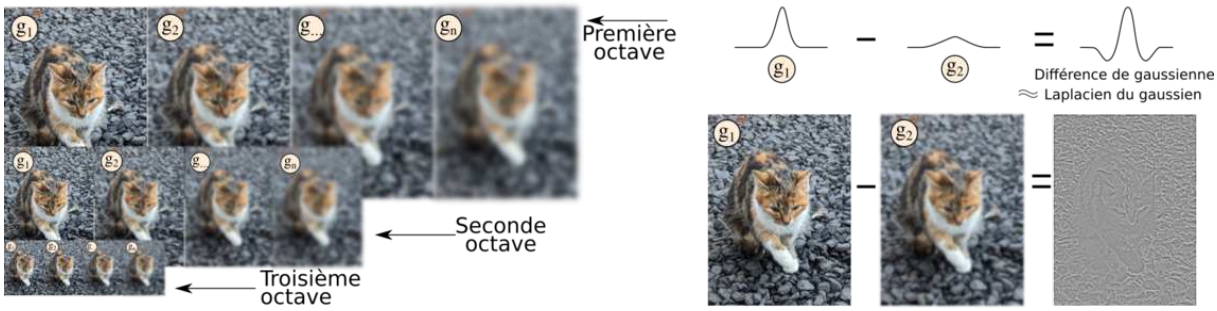


FIGURE 1.12 – Exemple de l'espace échelle calculé par SIFT à gauche et de la DoG entre deux gaussiennes d'une image d'origine.

où $w(x, y)$ correspond à une fenêtre rectangulaire à la position (x, y) agissant comme un masque, en assurant que seule la fenêtre rectangulaire est utilisée. $I(x, y)$ correspond à l'intensité de la fenêtre d'origine (x, y), $I(x + u, y + v)$ à l'intensité de la fenêtre décalée et enfin $E(u, v)$ représente la différence entre l'image d'origine et celle décalée.

Il est possible d'approximer l'équation 1.7 grâce au développement de séries de Taylor, ce qui nous donne la formule suivante :

$$E(u,v) \approx \begin{bmatrix} u & v \end{bmatrix} M \begin{bmatrix} u \\ v \end{bmatrix}, \text{ où } M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \tag{1.8}$$

Avec I_x et I_y les dérivées des images dans les directions x et y. Finalement, le détecteur de Harris va créer un score de réponse pour chaque pixel, en permettant de déterminer si une fenêtre peut contenir ou non un coin. Ce score est calculé selon l'équation :

$$R = det(M) - k(trace(M))^2 \tag{1.9}$$

avec $det(M) = \lambda_1 \lambda_2$, $trace(M) = \lambda_1 + \lambda_2$ et λ_1, λ_2 les valeurs propres de la matrice M. Selon la valeur R, l'algorithme permettra d'extraire ou non un point clé, avec :

- Si $R < 0$: le point se trouve sur un bord.
- Si $R > 0$: le point correspond à un coin.
- Si R est petit : le point correspond à une zone plane.

Scale-Invariant Feature Transform (SIFT) - Ce détecteur de points clés [Lowe, 2004] est grandement utilisé en traitement d'image et en reconnaissance d'images. Il permet de détecter et de décrire les points clés d'une image de manière robuste aux transformations affines, comme les rotations, les changements de perspective et les changements d'échelles.

Pour détecter les points clés, SIFT utilise une technique de détection multirésolution pour analyser l'image à différentes échelles. Cette technique est basée sur une pyramide de gaussiennes de l'image à analyser, à différentes échelles. Cet espace échelle (voir Figure 1.12)

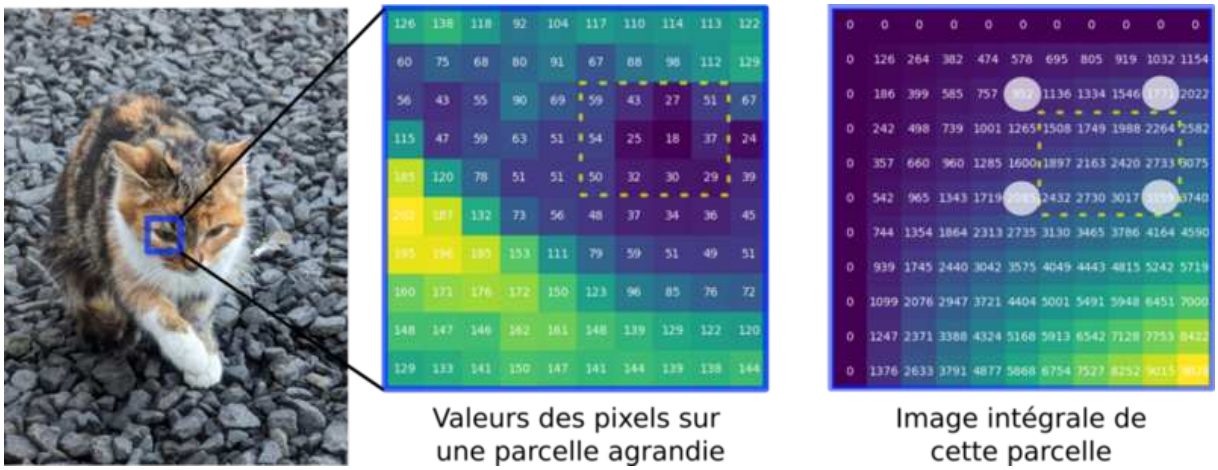


FIGURE 1.13 – Exemple de calcul d'image intégrale. L'image du centre représente une parcelle de l'image d'origine (à gauche). L'image de droite représente l'image intégrale de la parcelle extraite. Dans cet exemple, le calcul de la somme des pixels de la zone entourée de pointillés jaunes peut se faire en 3 opérations simples : SommeRectangulaire(zonejaune) $= 3359 + 952 - 1991 - 2085$.

est calculé en utilisant la fonction suivante :

$$L(x, y, \sigma) = \underbrace{\frac{1}{2\pi\sigma^2} e^{\frac{-(x^2+y^2)}{2\sigma^2}}}_{\text{filtre gaussien}} * \underbrace{I(x, y)}_{\text{l'image elle même}} \tag{1.10}$$

où σ correspondant au paramètre d'échelle.

Suite au calcul de cette espace-échelle, SIFT va utiliser la difference de gaussiennes (DoG) comme approximation du laplacien du gaussien (LoG), beaucoup moins coûteuse à calculer (voir Figure 1.12). Cette DoG peut s'écrire de la façon suivante :

$$DoG(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma) \tag{1.11}$$

En fin de compte, tout pixel dont la luminosité constitue un extrême parmi les 8 pixels adjacents dans l'image et les 18 autres pixels voisins dans l'espace d'échelle est considéré comme un point clé à détecter.

Speeded Up Robust Features (SURF) - Cet algorithme est basé sur des images intégrales précalculées en amont de la détection. Ces images intégrales permettent un calcul extrêmement rapide de la somme des pixels dans des zones rectangulaires. L'image intégrale est de la même taille que l'image originale. Chaque pixel de l'image intégrale contient la somme de tous les pixels situés au-dessus et à gauche de ce pixel dans l'image originale (voir Figure 1.13).

Ces images précalculées permettent de calculer la somme des pixels dans une zone rectangulaire en temps constant, grâce à la formule suivante :

$$\text{SommeRectangulaire}(x_1, y_1, x_2, y_2) = It(x_1, y_1) + It(x_2, y_2) - It(x_1, y_2) - It(x_2, y_1) \tag{1.12}$$

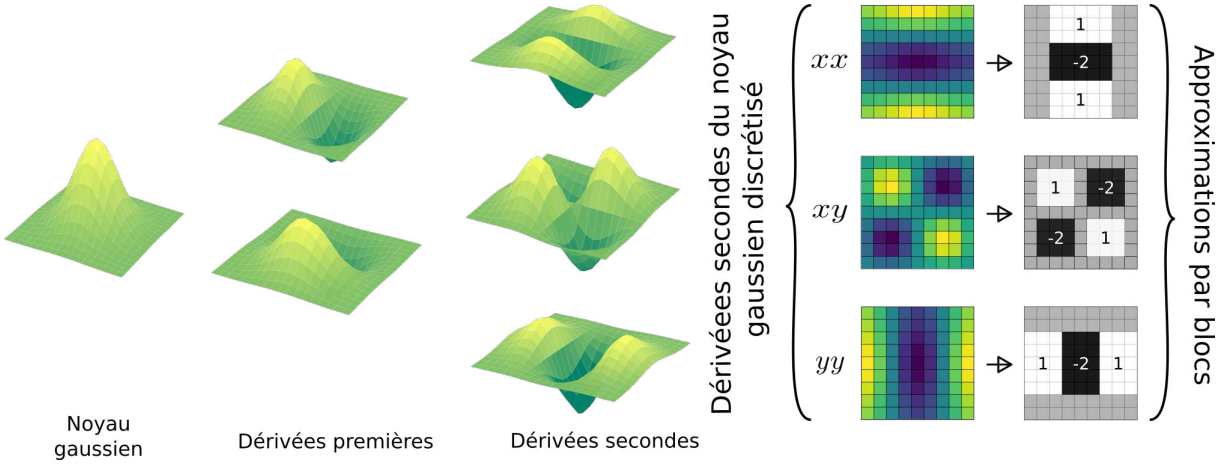


FIGURE 1.14 – De gauche à droite, les représentations 3D du noyau gaussien et des différentes dérivées (première et seconde) de ce noyau, les dérivées secondes du noyau gaussien discrétisés et enfin l'approximation par blocs de ces dérivées secondes.

où It correspond à l'image intégrale, et (x_1, y_1, x_2, y_2) les angles d'une zone rectangulaire.

Cette somme rectangulaire va permettre de calculer les blocs des approximations de dérivées secondes de noyau gaussiens, dans les directions y^2, x^2 et xy (voir Figure 1.14). Ces approximations seront ensuite utilisées pour construire la matrice hessienne de l'image, puis les extrema du déterminant de cette hessienne seront considérés comme des points clés.

KAZE - Le terme KAZE est dérivé du mot japonais signifiant "vent". Ce détecteur, à l'instar de SIFT, utilise une approche multirésolution. Cette approche utilise des espaces d'échelles non linéaires contrairement à SIFT et SURF, qui utilisent eux, un espace échelle composé de gaussiennes de l'image. La raison étant que, selon [Fernández Alcantarilla et al., 2012], «*le flou gaussien ne respecte pas les limites naturelles des objets et lisse de manière similaire les détails et le bruit lors de l'évolution de l'image originale dans l'espace échelle*».

KAZE va donc tout d'abord construire l'espace échelle non linéaire au moyen de techniques de fractionnements additifs d'opérateurs (AOS) et de la notion physique de diffusion non linéaire, de cette manière le lissage peut-être contrôlé localement. Cette méthode permet ainsi de réduire le bruit tout en préservant les contours des régions dans les images.

Une fois que l'espace d'échelle est créé, il est essentiel de détecter les points clés, qui sont des points présentant certaines caractéristiques (tels que l'indépendance de la position, la robustesse face aux transformations de l'image et l'indépendance à l'échelle). Lors du calcul de la matrice de la Hesse, un pixel sera identifié comme un point clé s'il représente la valeur maximale parmi ses voisins.

1.3.3 Description des descripteurs classiques

La seconde étape de l'extraction des points clés consiste en l'obtention du descripteur de ces derniers. Selon [Grand-Brochier, 2011 ; Mikolajczyk et Schmid, 2005], il y a une multitude

de descripteurs et de méthodes de mesure de distances disponibles pour évaluer les différences entre ces descripteurs. Nous pouvons citer comme exemples de descripteurs : les moments, les transformées et les histogrammes. Ce dernier type de descripteur est le plus connu, avec par exemple les descripteurs [Bay et al., 2006 ; Lowe, 2004], que nous décrirons dans la suite de cette section. Les histogrammes d'intensité lumineuse, proposés par [Swain et Ballard, 1991] permettent une description des voisinages des points clés rapide et peu coûteuse en termes d'espace mémoire et de temps de calcul. Dans ce manuscrit, nous nous focaliserons sur les descripteurs calculés sur les images à intensité de gris.

Cette étape de description des points clés permet de les associer les uns aux autres de manière aussi précise que possible, en utilisant leurs descripteurs respectifs. Cette étape d'appariement des points clés est importante dans de nombreuses applications liées à la vision par ordinateur et à la reconnaissance d'images, comme : la reconstruction 3D (en associant les points correspondants, il est possible d'estimer la structure tridimensionnelle d'une scène), la création de panoramas, la stabilisation d'images, etc.

Différentes méthodes d'appariement des points clés existent, dont les plus connues sont : le «*brute force*», les KD-tree [Bentley, 1975] ou encore les approches par plus proches voisins, la finalité étant la même pour ces différentes techniques. Pour un couple d'images $[\mathcal{I}, \mathcal{J}]$, on extrait deux ensembles de points $[P_\mathcal{I}, P_\mathcal{J}]$ et deux ensembles de descripteurs $[D_\mathcal{I}, D_\mathcal{J}]$ associés aux positions des points clés. La première méthode revient simplement à itérer à travers chacun des descripteurs $D_\mathcal{I}$ et $D_\mathcal{J}$, selon l'Algorithme 1.

Données : $[D_\mathcal{I}, D_\mathcal{J}]$
Résultat : Une liste de paires de descripteurs P
$Paire(x, y)$ une paire de descripteurs
$Paires = []$ une liste de paire de descripteurs :
pour *chaque descripteurs* d_i *dans* $D_\mathcal{I}$ **faire**
  $d_{min} = \infty$
  pour *chaque descripteurs* d_j *de* $D_\mathcal{J}$ **faire**
    d = DistanceEuclidienne(D_i, D_j)
    si $d < d_{min}$ **alors**
      $d_{min} = d$
      $P_{tmp} = Paire(d_i, d_j)$
    ajouter(P_{tmp}, Paires)

Algorithme 1 : Algorithme «*brute force*» d'appariement de descripteurs de points clés.

Nous allons à présent examiner sur les descripteurs associés aux détecteurs que nous avons précédemment abordés, ainsi que le descripteur BRIEF, [Calonder et al., 2012].

Scale-Invariant Feature Transform (SIFT) - Après avoir détecté les points clés et afin d'obtenir des points invariants par rapport aux rotations de l'image, SIFT va calculer l'orientation des

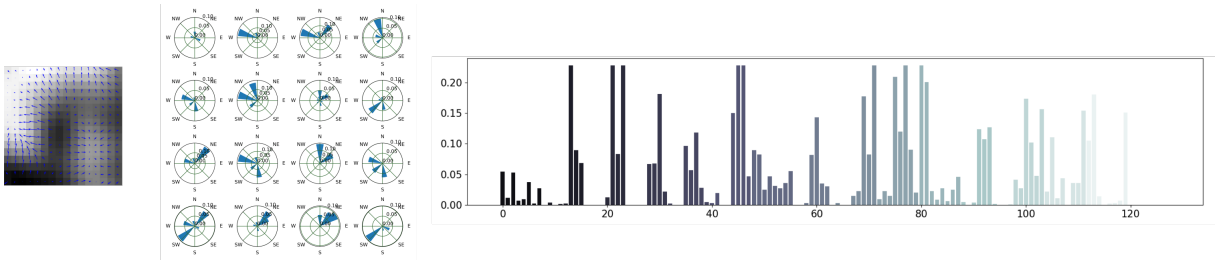


FIGURE 1.15 – De gauche à droite, la parcelle extraite de l'image d'origine avec le calcul des gradients de la parcelle (représentés par des flèches). Au centre, les histogrammes des 4×4 sous-blocs de la parcelle d'image, sous forme de diagramme polaire comprenant 8 directions. À droite, la concaténation des 16 diagrammes polaires formant le descripteur final.

caractéristiques extraites. Cette étape s'effectue sur un voisinage du point par le calcul des magnitudes et des gradients de ce voisinage. SIFT va ensuite construire un histogramme comprenant 36 valeurs, chacune représentant 10 degrés d'orientation. Les valeurs maximales de cet histogramme (supérieures à un pourcentage) permettront d'obtenir l'orientation du point clé.

Finalement, pour calculer le descripteur associé à chaque point clé, SIFT va utiliser un voisinage de 16×16 pixels, qui sera subdivisé en blocs de 4×4. Un nouvel histogramme comprenant 8 valeurs est calculé pour chaque sous-bloc. Enfin, par concaténation des valeurs de ces 16 histogrammes, nous obtenons un vecteur de description de taille 128, dont un exemple est illustré Figure 1.15.

Speeded Up Robust Features (SURF) - Ce descripteur va faire appel aux ondelettes de Haar (introduite par Haar Aldréd, en 1910), en vue de construire le vecteur de description associé à chaque point clé. Ce descripteur a été développé en vue d'approximer celui de SIFT en termes de répétabilité et distinction, mais également dans le but de diminuer le temps de calcul et d'améliorer l'étape de mise en correspondance des descripteurs.

Dans ce but, SURF se propose d'utiliser une seconde fois l'image intégrale précédemment calculée. Le descripteur considère une fenêtre de taille $20s \times 20s$ au voisinage du pixel détecté, avec s le facteur d'échelle du point d'intérêt. Cette région de pixels est ensuite subdivisée en 16 sous-régions de 5×5 pixels. Les coefficients des ondelettes de Haar sont ensuite calculés pour les directions d_x et d_y, comme le montre la Figure 1.16. Ces réponses vont être sommées dans chaque sous-région et inscrites dans un vecteur v selon l'équation suivante :

$$v = (\sum d_x, \sum |d_x|, \sum d_y, \sum |d_y|) \tag{1.13}$$

Finalement, chacun de ces vecteurs sera assemblé en vue de former un vecteur de description de dimension 64 en regard des vecteurs SIFT de dimension 128.

Binary Robust Independent Elementary Features (BRIEF) - [Calonder et al., 2012] est le premier détecteur binaire de points clés calculé directement sur la comparaison de l'intensité des pixels. Selon [Hassaballah et al., 2019], l'utilisation de descripteurs binaires présente quelques

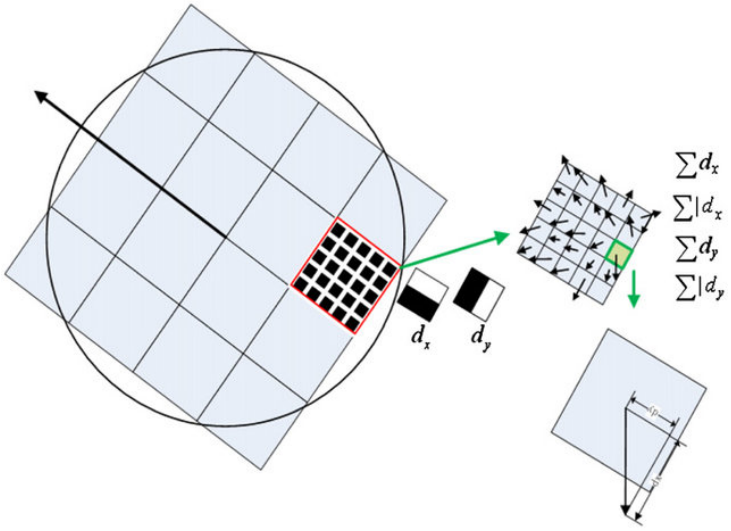


FIGURE 1.16 – Représentation schématique du calcul d'un descripteur SURF, avec la parcelle extraite de l'image source subdivisé en 4×4 sous-blocs.
Source: Analysis and Evaluation of Keypoint Descriptors for Image Matching Hassaballah et al. (2019).

avantages tels que : des performances comparables aux descripteurs non binaires ([Bay et al., 2006 ; Lowe, 2004]), un temps de calcul et un espace mémoire nécessaire au stockage des caractéristiques nettement diminué. Concernant l'étape de mise en correspondance de ce type de descripteurs, la distance de Hamming [Li et Jain, 2009] est utilisée plutôt que la distance euclidienne, voir équation 1.15 et équation 1.14. Ce calcul de distance peut être exécuté très rapidement sur les processeurs modernes en une seule opération suivie d'une somme de bits.

$$\text{Distance}_{\text{euclidienne}}(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{1.14}$$

$$\text{Distance}_{\text{hamming}}(x, y) = \sum_{i=1}^{n} (x_i \oplus y_i) \tag{1.15}$$

Dans un premier temps, BRIEF va lisser l'image grâce à un filtre gaussien. Ensuite, le descripteur sera calculé dans un voisinage autour d'un point clé détecté, selon l'équation suivante :

$$T(p; x, y) = \begin{cases} 1 \text{ si } p(x) < p(y) \\ 0 \text{ sinon} \end{cases} \tag{1.16}$$

Cette équation correspond à un test de calcul de différence d'intensité lumineuse, avec $p(x)$ l'intensité à la localisation x, dans la parcelle d'image p. Les paires de localisations $n(x, y)$ sont choisies de différentes manières, telles que : un tirage aléatoire échantillonné uniformément, un échantillonnage aléatoire utilisant une distribution gaussienne, etc. Ces motifs d'échantillonnages

FIGURE 1.17 – Représentation des cinq approches pour choisir les paires d'emplacements sélectionnés. Ces exemples montrent 128 tests dans chaque image.
Source: Calonder et al. (2012).

sont au nombre de cinq et représentés dans la Figure 1.17, avec de gauche à droite :

- Distribution aléatoire : cet échantillonnage permet de recouvrir plus efficacement l'image, sans tentative d'influencer la distribution.
- Distribution gaussienne : les paires sont tirées aléatoirement, mais suivant une distribution gaussienne centrée sur le plus d'intérêt.
- Distribution gaussienne courte : les paires sont également tirées aléatoirement, mais cette méthode suppose que les paires courtes sont plus chargées en informations.
- Distribution gaussienne polaire : cet échantillonnage est similaire à la deuxième hypothèse d'échantillonnage, mais tente d'ajouter plus d'ordre en tirant x et y sur une grille polaire.
- Dans cette dernière distribution, $x_i = (0, 0)$ et y_i sont tirés sur une grille polaire.

Nous avons examiné quelques méthodes traditionnelles d'extraction de points clés, qui ont été développées manuellement (handcrafted en anglais) par les experts en science des données et en analyse d'images. Étant donné que l'analyse d'images est bien adaptée à l'apprentissage automatique, l'utilisation de réseaux de neurones pour apprendre de manière automatique de nouvelles caractéristiques à extraire dans les images est devenue une solution naturelle.

La principale différence entre les méthodes classiques et celles apprises réside dans la façon dont les caractéristiques que l'on souhaite mettre en évidence sont définies. Les méthodes classiques se concentrent sur la recherche d'éléments prédéfinis par les concepteurs des extracteurs, tandis que les méthodes d'apprentissage utilisent un grand nombre de paramètres appris par le réseau, qui permettent de découvrir de nouveaux éléments à mettre en évidence.

Nous allons maintenant détailler quelques détecteurs et descripteurs de points utilisant les méthodes d'apprentissage. Mais avant de décrire ces réseaux de neurones, nous allons détailler les différentes étapes mises en place dans les réseaux neuronaux, ainsi que les données utilisées par les RNC. Les RNC sont des types de réseaux de neurones artificiels acycliques (feed-forward), dans lesquels le motif de connexion entre les neurones est inspiré par le cortex visuel des animaux.

1.4 Réseaux de neurones à convolutions et données d'apprentissage

L'apprentissage informatique automatique par réseaux de neurones, est une branche de l'intelligence artificielle (IA) qui vise à imiter la méthode d'apprentissage utilisée par les êtres humains pour acquérir des connaissances. Chez les Hommes, ce processus est automatique tout au long de la vie ; à l'inverse, les machines sont incapables d'un tel apprentissage automatique.

Les premiers essais d'apprentissage automatique par des programmes informatiques remontent aux années 1950, avec l'invention de réseaux de type séparateur à vaste marge (angl. «*Support Vector Machine*», SVM) [Cortes et Vapnik, 1995] ou Perceptron [Rosenblatt, 1958]. Ces SVM ont pour but de résoudre des problèmes simples de discrimination ou de régression.

La publication d'un premier réseau de neurones artificiels profond à convolution nommé *néocognitron*, par l'informaticien japonais K. Fukushima [Fukushima, 1980], a permis de démontrer les performances élevées de la reconnaissance d'images, en particulier dans le cas de la reconnaissance des caractères manuscrits. Ensuite, Geoffrey Hinton, un chercheur canadien, ainsi que David Rumelhart et Ronald Williams, tous deux chercheurs américains (voir [Rumelhart et McClelland, 1987]), ont réussi à réaliser la première mise en œuvre fonctionnelle du mécanisme de rétropropagation du signal d'erreur à travers les réseaux de neurones artificiels profonds. Cela constituait un obstacle majeur à l'amélioration des performances des réseaux de neurones. Par la suite, avec l'augmentation continue de la puissance de calcul des ordinateurs, Yann LeCun ([LeCun et al., 1989]) a publié un article décrivant une architecture de réseau neuronal simple et efficace qui utilise le mécanisme de rétropropagation pour la reconnaissance de chiffres manuscrits.

Au fil des années, les chercheurs ont apporté de nombreuses avancées et améliorations à l'architecture des RNC de base. En 2012, [Krizhevsky et al., 2012] a démontré la puissance de l'apprentissage profond en entraînant un RNC profond sur un grand nombre de données d'images et en obtenant des performances de pointe dans le défi ImageNet, [J. Deng et al., 2009], une référence largement utilisée en vision par ordinateur.

Ce sont ces RNC que nous utiliserons dans la suite de ce manuscrit. Nous allons donc maintenant présenter quelques composants clés de ces réseaux et de leur apprentissage.

1.4.1 Réseau neuronal convolutif

Nous allons ici expliquer quelques opérations importantes (illustrées Figure 1.19) utilisées dans les RNC telles que : la convolution, le pooling et différentes fonctions d'activations utiles pour la suite du manuscrit.

Convolution et pooling - La couche convolutionnelle (ou convolution) est une opération mathématique simple en plusieurs étapes. Elle consiste dans un premier temps à définir une taille de fenêtre de filtre, qui représente une caractéristique que nous aimerions voir apparaître

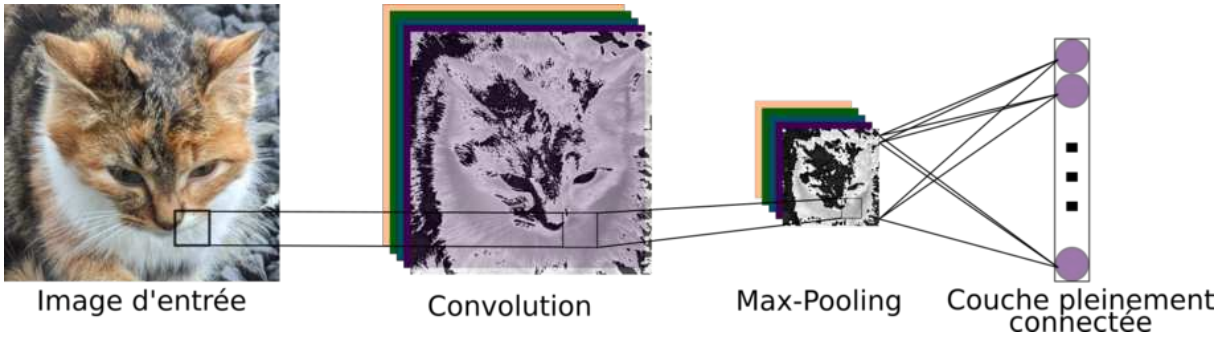


FIGURE 1.18 – Exemple de quelques couches utilisées dans les RNC.

dans l'image. Ce filtre va ensuite se déplacer progressivement, d'un certain nombre de pixels, sur toute l'image. À chaque portion d'image rencontrée par le filtre, un calcul de convolution va être effectué afin d'obtenir une carte d'activation (ou «*feature map*» en anglais), qui indiquera où sont localisées les caractéristiques dans l'image. La couche de pooling (ou «*regroupement*» en français) est une opération permettant de sous-échantillonner une matrice ; cette couche succède généralement à une couche de convolution. Il existe plusieurs opérations de pooling différentes, dont les plus populaires sont le max-pooling et l'average-pooling. La première opération va permettre de sélectionner la valeur maximale au sein d'une fenêtre donnée, la seconde opération sélectionnera la valeur moyenne de cette fenêtre.

Différents paramètres permettent d'ajuster ces deux opérations : la dimension de la fenêtre, le pas de déplacement de cette fenêtre, le padding associé à l'image d'entrée et enfin le paramètre de dimension de filtre (qui ne concerne que l'opération de convolution).

La dimension du filtre correspond au nombre de filtres qui seront appliqués à l'image par l'opération de convolution. Si l'on prend une parcelle d'image de taille $5 \times 5 \times 1$ avec 1 canal et 5 filtres de taille $3 \times 3 \times 1$, l'application de ces filtres et leur concaténation, va permettre de faire émerger une nouvelle carte de caractéristiques d'une taille $3 \times 3 \times 5$ avec ici 5 canaux.

En travaillant avec des images, chaque pixel est représenté par une valeur qui décrit sa couleur ou sa luminosité. Les couleurs sont souvent décrites en utilisant un modèle de couleur RGB (rouge, vert, bleu), où chaque couleur est représentée par un canal séparé. Ainsi, pour une image couleur, nous avons trois canaux : un canal rouge, un canal vert et un canal bleu. Dans les réseaux de neurones, chaque canal est généralement traité comme un élément à part entière d'une couche et peut être analysé séparément. Lors du passage d'une image dans une couche de convolution, chaque canal est présenté à chaque filtre séparément. Les résultats sont ensuite combinés pour produire la sortie de la couche convolutionnelle.

La dimension de la fenêtre correspond à la taille en pixels que va prendre le filtre à appliquer à l'image. Cette fenêtre va être décalée sur l'image entière en fonction de deux paramètres :

- le pas (ou «*stride*» en anglais) : Le pas définit la distance en pixels entre chaque sous-échantillonnage. Si le pas est défini à 1, alors le filtre effectue une opération de convolution sur tous les pixels de l'image. Si le pas est défini à 2, alors le filtre saute une ligne ou une

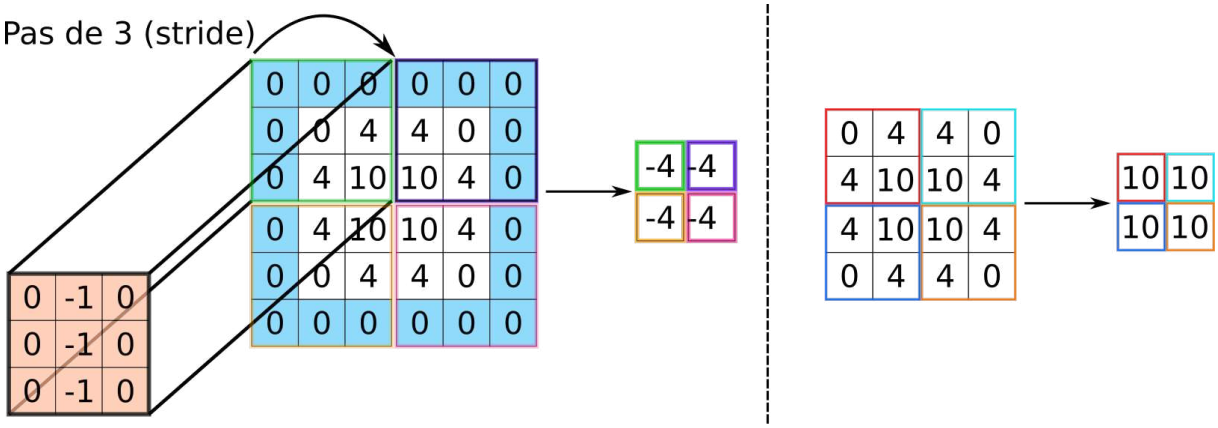


FIGURE 1.19 – L'illustration présente l'application d'une opération de convolution sur une matrice 2D de pixels à gauche, ainsi que l'application de l'opération de pooling à droite (l'exemple montre l'application du max-pooling). Dans l'image de gauche, la flèche en haut à gauche représente le pas (ou stride en anglais) auquel le noyau de convolution est appliqué, et les 0 en bleu représentent le zéro-padding appliqué à l'image d'entrée.

colonne entre chaque opération de convolution, ce qui réduit la dimension de l'image en sortie de la couche Conv.

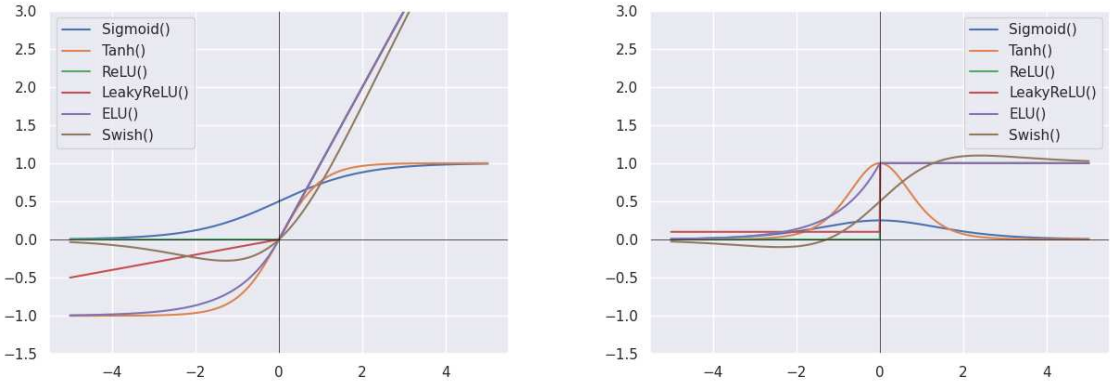
- le padding (ou «*remplissage*» en français) : ce paramètre permet de gérer les dimensions des images ou des matrices de données en entrée dans les couches Convolutionnelles d'un modèle de réseau neuronal. Il consiste à ajouter une certaine valeur (généralement zéro) aux bords de l'image pour éviter une réduction excessive de la dimension. Cela permet de conserver les informations importantes présentes aux bords de l'image. De plus, le padding peut aider à contrôler la taille de la sortie de la couche de convolution et donc à mieux contrôler la taille globale du modèle.

Couche pleinement connectée - Cette couche est généralement la dernière d'un RNC. Elle permet de connecter tous les nœuds d'une couche pour former un vecteur de sortie, en appliquant une combinaison linéaire à travers les éléments. Ce vecteur de sortie peut renvoyer un vecteur de taille N désiré, N correspondant au nombre de classes dans un réseau de classification d'images ou la taille d'un vecteur de description de points clés dans notre cas.

Fonctions d'activations - Les fonctions d'activation (voir Figure 1.20(a)) jouent un rôle crucial dans la transformation non linéaire des données. Cette transformation permet une évolution spatiale de la représentation des données, tout en permettant à un réseau de neurones suffisamment large d'approximer n'importe quelle fonction, voir [Ding-Xuan, 2020].

Pour choisir la bonne fonction d'activation à utiliser lors de l'apprentissage, il convient de considérer la transformation directe qu'elle applique aux données. Voici quelques exemples de ces transformations :

- $ReLU(x) = max(x, 0)$: la plus connue et la plus simple à utiliser parmi ces fonctions



(a) Fonctions d'activations **(b)** Dérivées premières des fonctions d'activations

FIGURE 1.20 – Graphiques représentant quelques fonctions d'activations (graphique de gauche) et leurs dérivées premières respectives (graphique de droite).

d'activation est sans doute unité linéaire rectifiée (angl. «*Rectified Linear Unit*», ReLU). Elle permet de filtrer les données et de laisser les valeurs positives avancer dans les couches suivantes du réseau de neurones, voir [Abien Fred, 2018].

- $Sigmoïd(x) = \frac{1}{1+\exp(-x)}$: Cette fonction renvoie une valeur entre $[0, 1]$ et est très utilisée dans les réseaux de classification binaire.
- $tanh(x) = \frac{\exp(x)-\exp(-x)}{\exp(x)+\exp(-x)}$: tanh est une valeur entre $[-1, 1]$ et correspond à la fonction tangente hyperbolique, voir [Shiv Ram et al., 2021].

Il est également important de considérer les dérivées de ces fonctions d'activations (voir Figure 1.20(b)). Ce sont les dérivées de ces fonctions qui entrent en compte lors du processus d'apprentissage, car les poids (les paramètres qui contrôlent la contribution de chaque entrée pour la sortie d'une unité de neurone) vont être actualisés en fonction du gradient de la fonction (gradient correspondant essentiellement à la dérivée d'une fonction). Prenons en exemple la dérivée de la fonction sigmoïd, qui tend vers 0 à $+/-\infty$, or si le gradient est trop petit (proche de 0), les poids sont actualisés avec de très petits nombres : il en résulte un apprentissage très lent. Ce problème est connu sous le nom de problème du gradient évanescent (ou «*vanishing gradient*» en anglais).

Descente de gradient - Nous allons maintenant expliquer le mécanisme nommé *descente de gradient* utilisé dans les réseaux neuronaux lors de l'apprentissage. Son objectif est de minimiser la fonction de coût («*loss function*» en anglais) qui mesure l'écart entre les prédictions du réseau et les valeurs réelles attendues. Le processus débute par l'initialisation aléatoire des poids et des biais du réseau. Lors de la phase de rétropropagation, les erreurs de prédiction sont propagées de la couche de sortie vers les couches précédentes, en utilisant la dérivée partielle de la fonction de coût par rapport à chaque poids et biais. Ces dérivées sont ensuite utilisées pour mettre à jour les poids et les biais afin de minimiser l'erreur. Cela se fait en ajustant les

paramètres dans la direction opposée du gradient de la fonction de coût, d'où le nom «*descente de gradient*». Ce processus est répété sur l'ensemble des données d'entraînement pendant plusieurs itérations, appelées époques («*epoch*» en anglais), jusqu'à ce que le réseau converge vers des valeurs de poids et de biais qui minimisent la fonction de coût. Ainsi, la descente de gradient permet au réseau de neurones d'apprendre à ajuster ses paramètres de manière itérative afin de produire des prédictions de plus en plus précises.

Fonction de perte - Parmi les fonctions de coût (ou fonction de perte ou encore fonction objective) les plus connues, nous retrouvons l'erreur moyenne absolue («*mean absolute error or L1 loss*» en anglais), l'erreur quadratique moyenne («*mean squared error or L2 loss*» en anglais), l'entropie croisée binaire («*binary cross-entropy*» en anglais) ou encore la perte charnière («*hinge loss*» en anglais) que nous allons décrire maintenant.

La fonction de perte charnière est une fonction de coût utilisée pour maximiser la marge de séparation entre les classes dans les problèmes de classification binaire, en pénalisant les exemples mal classés ou proches de la frontière de décision. Elle favorise ainsi un modèle qui effectue une classification précise et maintient une marge maximale entre les frontières de décision. Elle s'écrit comme suit :

$$l(y) = \max(0, 1 - y_{vrai} \cdot y_{pred}) \tag{1.17}$$

Avec y_{pred} la prédiction de classe attribuée par le réseau neuronal et y_{vrai} le résultat attendu.

Lot et mini-lot - Nous allons évoquer la taille des lots et mini-lots («*batch*» et «*mini-batch*» en anglais). Un lot de données est un ensemble d'échantillons d'entraînement que l'on utilise pour mettre à jour les poids du réseau. Plutôt que d'effectuer des mises à jour pour chaque échantillon individuellement, on calcule la moyenne des erreurs de prédiction pour l'ensemble des données et on met à jour les poids en conséquence. Cela permet d'amortir les variations dues à des échantillons individuels et de bénéficier d'une mise à jour plus stable.

Un mini-lot est une variation du lot où l'on divise l'ensemble d'entraînement en plusieurs sous-ensembles plus petits. Au lieu de traiter tous les échantillons en même temps, on alimente le réseau par lots plus petits, ce qui permet d'économiser des ressources computationnelles. Les mini-lots sont généralement choisis aléatoirement à partir de l'ensemble d'entraînement et permettent de réduire le bruit dû à des échantillons individuels tout en fournissant des mises à jour fréquentes pour les poids.

Réseau profond et réseau large - L'utilisation de réseaux profonds, c'est à dire qui contiennent de multiples couches successives, présente un avantage majeur, qui est la capacité à apprendre des caractéristiques à différents niveaux d'abstraction. Un réseau large quant à lui contient généralement peu de couches successives, mais un plus grand nombre de neurones par couches. Par exemple, dans un RNC profond entraîné à classer des images, la première

couche reconnaîtra les éléments de base tels que les bords, puis les couches suivantes reconnaîtront des formes de plus en plus complexes, comme les yeux et les nez. Enfin, la dernière couche apprendra des caractéristiques d'ordre supérieur, telles que les visages. Les couches multiples sont ainsi bénéfiques, car elles permettent de généraliser plus efficacement, en apprenant toutes les caractéristiques intermédiaires nécessaires entre les données brutes et la classification de haut niveau.

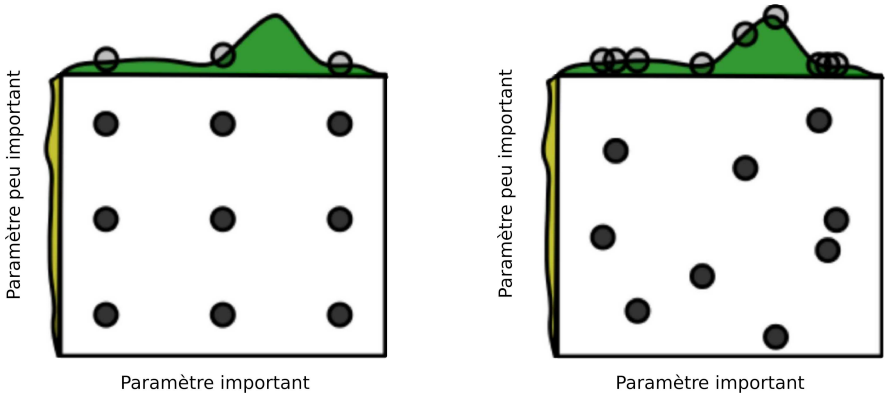
En général, l'utilisation d'un réseau profond plutôt qu'un réseau large et peu profond s'explique par le fait que l'on cherche à minimiser la taille du réseau tout en obtenant de bons résultats. En effet, augmenter la taille du réseau risque d'introduire davantage de paramètres, augmentant ainsi les risques de surapprentissage. Un réseau très large et très profond pourrait alors se contenter de mémoriser les sorties souhaitées sans réelle capacité de généralisation sur de nouvelles données.

Optimisation des hyper-paramètres - Lorsque l'on cherche à optimiser un réseau de neurones et donc à rechercher les meilleurs hyperparamètres, plusieurs méthodes existent. Les deux méthodes les plus employées sont : la première par grille et la recherche aléatoire, voir [Bergstra et Bengio, 2012] et Figure 1.21. Dans la recherche en grille, on sélectionne un ensemble de valeurs pour chaque paramètre, puis on crée un ensemble d'essais en assemblant toutes les combinaisons possibles ; cette solution est plus simple à mettre en œuvre et à paralléliser, mais elle souffre du fléau de la dimensionnalité. En effet, le nombre de combinaisons de valeurs augmente de manière exponentielle avec le nombre d'hyperparamètres. La recherche aléatoire, variante de la recherche en grille, tire indépendamment des valeurs aléatoires de l'espace à explorer, offrant les avantages pratiques de la recherche en grille tout en échangeant une légère réduction d'efficacité dans les espaces de faible dimension contre une amélioration significative dans les espaces de grande dimension ; elle sert de référence pour comparer les algorithmes d'optimisation d'hyperparamètres plus avancés.

Dans le cas d'un hyperparamètre peu important, la recherche aléatoire est susceptible de tester la fonction de coût de manière plus approfondie. Sur la Figure 1.21, le nombre de points gris est similaire pour les deux types de recherches, mais dans le cas de la recherche en grille, des ressources ont été utilisées inutilement pour tester des paramètres sans importance. Cependant, la recherche aléatoire peut encore manquer complètement l'optimum global, comme dans la deuxième illustration, voir Figure 1.21(b).

Apprentissage par transfert - Le transfert d'apprentissage («*transfer learning*» en anglais, voir [Tan et al., 2018]) est une technique dans le domaine de l'apprentissage automatique où les connaissances acquises à partir d'une tâche source sont utilisées pour améliorer les performances d'une tâche cible. Plutôt que de construire un modèle à partir de zéro pour la tâche cible, on utilise un modèle préentraîné sur une tâche similaire ou sur une grande quantité de données génériques.

Le processus de transfert d'apprentissage se déroule généralement en suivant plu-



(a) Recherche par grille. **(b)** Recherche aléatoire.

FIGURE 1.21 – Les Schémas 1.21(a) et 1.21(b) présentent les deux approches permettant une optimisation efficace des hyperparamètres durant l'entraînement d'un réseau de neurones. Points noirs : combinaisons de paramètres testés. Courbes vertes : fonction objective réelle en fonction du paramètre donné. Sur l'axe des y, la courbe est presque plate, ce qui signifie que ce paramètre a un faible impact sur la fonction objective. Cependant, sur l'axe des x, un maximum apparaît clairement, correspondant à la valeur optimale de ce paramètre. Points gris : projection de la combinaison de paramètres testée sur la courbe verte.
Source: Image extraite de Bergstra et Bengio (2012)

sieurs étapes. La première consiste à préentraîner un modèle source sur une tâche, comme la classification d'images à grande échelle, ce qui permet au modèle d'apprendre à extraire des caractéristiques pertinentes. Le modèle est ensuite réentraîné sur un ensemble de données spécifiques à une tache cible, et ainsi permettre de spécialiser le réseau à un domaine. Cette étape de réentraînement consiste à ajuster les poids en utilisant les données de la tâche cible, avec en général une quantité moins importante de données que lors de l'apprentissage sur la tâche source.

Le transfert d'apprentissage permet de bénéficier des connaissances et des caractéristiques extraites par le modèle préentraîné, ce qui peut conduire à une amélioration significative des performances sur la tâche cible, surtout lorsque les données d'entraînement sont limitées. Il permet également de réduire le temps et les ressources nécessaires pour entraîner un modèle à partir de zéro.

Inférence - En apprentissage automatique, la phase d'inférence désigne l'utilisation d'un modèle d'intelligence artificielle une fois qu'il a été entraîné sur un ensemble de données d'apprentissage et testé sur un ensemble de données de validation. Elle correspond au déploiement du modèle et à l'application de son évaluation dans des situations réelles en utilisant des données provenant du terrain.

1.5 Détecteurs et descripteurs de points par apprentissage

Dans cette Section, nous ne séparerons pas l'étape de détection de celle de description, car ces étapes sont très souvent liées les unes aux autres au sein des extracteurs basés sur

l'apprentissage. En effet, selon [Dusmanu et al., 2019], il existe deux approches différentes pour l'extraction de caractéristiques. La première désigne les méthodes d'extractions «*détecter-puis-décrire*» et la seconde désigne celles qui vont «*détecter-et-décrire*». Dans le premier type d'approche, l'extracteur va en premier lieu appliquer un détecteur de points [Bay et al., 2006 ; DeTone et al., 2017 ; Harris et Stephens, 1988 ; Lowe, 2004 ; Savinov et al., 2016 ; Yi et al., 2016] dans le but d'identifier un ensemble de points clés. Ensuite, le descripteur [Balntas et al., 2016 ; Tola et al., 2008 ; Yi et al., 2016], après avoir extrait une parcelle de l'image d'origine autour d'un point clé, va entrer en jeu pour décrire ce point.

Nous allons étudier quelques-uns de ces extracteurs par apprentissages «*détecter-et-décrire*» et «*détecter-puis-décrire*».

1.5.1 Réseau siamois et par triplets

Nous commencerons tout d'abord par présenter les réseaux siamois [Bromley et al., 1993 ; Gordo et al., 2017 ; Vijay Kumar B et al., 2015] (ou «*Siamese Networks*») et les réseaux par triplets (voir Figure 1.22) qui sont deux types de modèles de réseaux de neurones qui sont utilisés pour les tâches de reconnaissance d'objets et de comparaison de données. L'architecture générale de ces approches permet également de comprendre les structures employées par d'autres réseaux que nous décrirons par la suite.

Les réseaux siamois sont appelés ainsi, car ils ont une structure symétrique, où deux réseaux de neurones partagent les mêmes poids et sont formés en parallèle. Ils apprennent en comparant deux images pour savoir si elles représentent la même chose. Les réseaux prennent en entrée deux images et produisent en sortie une mesure de similitude.

Les réseaux par triplets [Balntas et al., 2016 ; Loiseau–witon et al., 2021 ; Schroff et al., 2015 ; Vijay Kumar B et al., 2015] (voir Figure 1.22) sont une extension des réseaux siamois, où trois images sont saisies en entrée (au lieu de deux). La première image est appelée "ancre", la deuxième est appelée "image positive" et la troisième est appelée "image négative". Le but est de former le réseau de manière à ce que la distance entre l'ancre et l'image positive soit plus petite que la distance entre l'ancre et l'image négative. Cette architecture est utile pour la reconnaissance d'objets et la classification d'images.

Les deux types de réseaux ont pour objectif de produire un vecteur de description capable de différencier les différentes sections d'images, que ce soit dans le cadre de la création d'un descripteur de points ou d'un système de reconnaissance faciale utilisant des images complètes. Afin de générer ce vecteur de description, la dernière couche de ces réseaux sera une couche pleinement connectée, dont la dimension doit être spécifiée.

Pour les réseaux siamois à deux branches, ou par paires (illustré dans la Figure 1.22), les vecteurs de description d_1 et d_2 des images $\mathcal{I}$ et $\mathcal{J}$, sont comparés en utilisant la norme L2. Les images peuvent provenir des mêmes classes ou bien de classes différentes. Dans le cadre des images représentant des chiffres écrits à la main [L. Deng, 2012], une classe représente l'ensemble des images d'un même chiffre. Sachant que $y = 1$ si les images $\mathcal{I}$ et

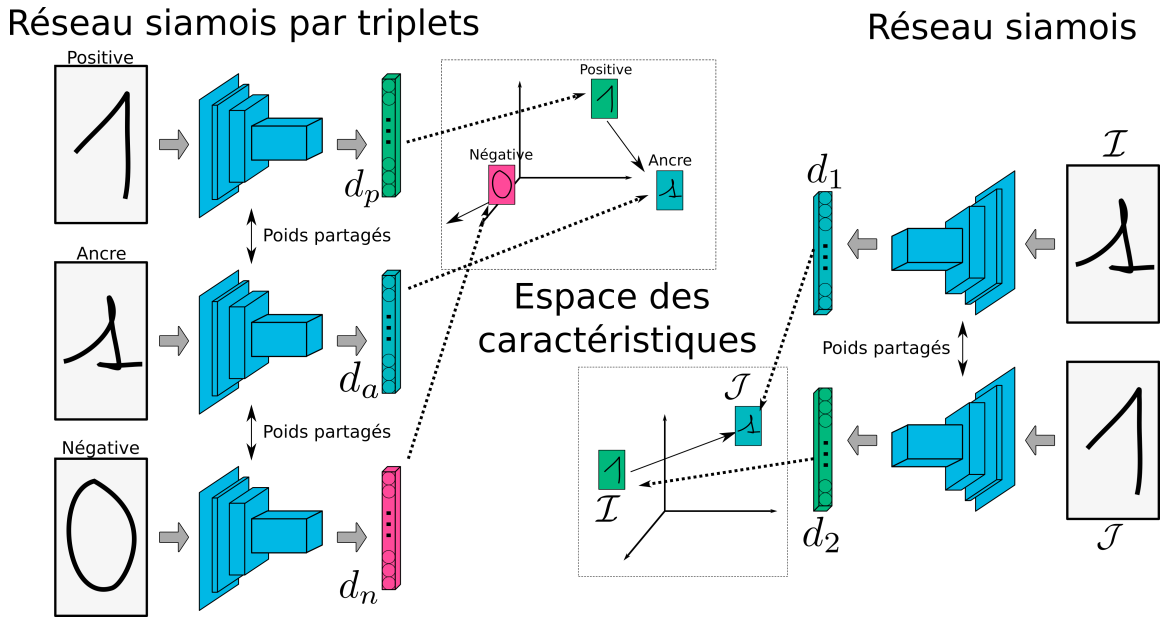


FIGURE 1.22 – Schéma de l'architecture des réseaux siamois à deux ou trois branches (réseau par triplets). Dans ce schéma, les éléments en bleu représentent un réseau de neurones dont les poids sont partagés lors du traitement de chaque image. Le résultat obtenu après le passage dans une couche pleinement connectée est un vecteur de description qui permet de décrire au mieux l'image donnée en entrée. Au centre de la figure, on peut observer l'espace des caractéristiques ou des descripteurs. Les réseaux siamois par triplets sont entraînés à produire des descripteurs pour des triplets d'images : l'image positive, l'image d'ancrage et l'image négative. L'objectif est de rapprocher les descripteurs de l'image positive et de l'ancrage dans cet espace, tandis que le descripteur de l'image négative doit être éloigné des deux autres. Les réseaux siamois visent à faire se rapprocher les descripteurs des images qui correspondent à un même élément, dans ce cas, le chiffre un.

$\mathcal{J}$ proviennent de la même classe, et $y = 0$ dans le cas contraire. La fonction de perte va pénaliser le réseau lorsque la paire d'images de la même classe se trouve à une distance au-delà d'une marge m ou lorsqu'une paire d'images se trouve à une distance inférieure à la marge. La fonction de perte se formule ainsi :

$$\mathcal{L}_{\text{contrastive}} = -\frac{1}{N}\sum_{i=1}^{N}\Big(y_i||d_{1_i} - d_{2_i}||_2 + (1-y_i)\max(m - ||d_{1_i} - d_{2_i}||_2, 0)\Big) \tag{1.18}$$

où N correspond au nombre de paires de descripteurs.

L'approche par triplets (illustrée sur la Figure 1.22) constitue une extension des réseaux siamois par paires, que nous avons expliquées précédemment. L'apprentissage par triplets implique la création de triplets d'images a, p, n, où a est appelée ancre, p est appelée positive et représente une image différente, mais appartenant à la même classe que a, et n est une image négative représentant une classe différente. L'objectif est d'optimiser les paramètres du RNC pour rapprocher les vecteurs de description de a et p dans l'espace des descripteurs, tout en éloignant le vecteur de description n de celui de a. La fonction de perte par triplet

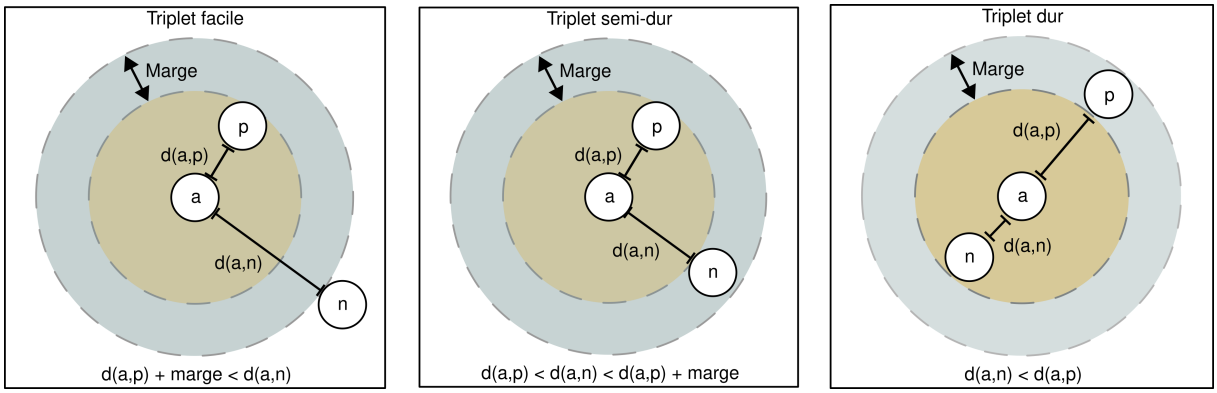


FIGURE 1.23 – Schéma représentant les trois catégories de triplets possibles.

est définie de la manière suivante :

$$\mathcal{L}(d_a, d_p, d_n) = \max\Big(||d_a - d_p||_2 - ||d_a - d_p||_2 + m, 0\Big) \tag{1.19}$$

où d_a, d_p, d_n correspondent respectivement aux descripteurs de l'image a, p, n et m correspond à la marge à appliquer. La minimisation de cette fonction de perte doit amener la distance entre a et p (notée $d(a,p)$) à tendre vers 0, et la distance entre a et n (notée $d(a,n)$) à tendre vers $d(a,p) + m$.

Il y a plusieurs approches possibles pour former les triplets ou les paires afin d'entraîner ces deux types de réseaux. L'approche la plus basique revient à itérer sur la totalité des images de l'ensemble des données à notre disposition. Cette solution n'est pas optimale pour deux raisons : le nombre de possibilités de paires ou de triplets très important, et l'utilité des triplets pour l'entraînement. En effet, un triplet apportant une contribution importante à l'entraînement au début peut ne plus être intéressant par la suite.

En utilisant l'ensemble de données MNIST (voir L. Deng (2012)) comme base d'apprentissage, qui est une collection d'images en noir et blanc de chiffres manuscrits de 0 à 9. Nous pouvons calculer un vecteur de description pour chaque image de la base de données, et apprendre à notre réseau à bien distinguer les éléments différents de ceux similaires, dans l'espace de description. Pour cela, nous pouvons poser $C = 10$ comme le nombre de classes (les chiffres de 0 à 9) et $N = 10$ le nombre d'images par classes utilisées lors d'une phase d'apprentissage (mini-lot ou «*mini-batch*» en anglais). La stratégie basique permet de produire $CN(N-1)(CN-N) = 100 * (9) * (10) = 9000$ triplets possibles (CN ancres, $N-1$ images positives par ancres et $CN - N$ négatives possibles).

Une autre stratégie plus optimale revient à sélectionner les triplets les plus avantageux dans un mini-lot d'apprentissage. Il convient donc d'analyser les différentes catégories de triplets, qui sont :

- Triplet facile : cette catégorie n'apporte rien à l'entraînement, car la sortie de la fonction de perte est 0.

- Triplet semi-dur : l'image négative est ici plus proche de l'ancre que l'image positive.
- Triplet dur : ici, l'image négative n'est pas plus proche de l'ancre par rapport à la positive, mais la perte est tout de même positive.

En utilisant ces trois catégories, nous excluons les triplets faciles du cycle d'apprentissage en identifiant les triplets durs et semi-durs, au sein d'un mini-lot.

1.5.2 TILDE : A Temporally Invariant Learned Detector et LIFT : Learn Invariant Feature Transform

Le détecteur TILDE (voir [Verdie et al., 2014]) a été développé dans le seul but de détecter des points clés robustes aux changements d'illuminations, de temporalité et de saisons. Pour la mise en place du jeu de données, TILDE utilise le détecteur SIFT afin d'extraire les points clés dans les différentes images et suppose le fait que chaque image d'une même scène est acquise du même point de vue (voir Figure 1.24(a)). Un point détecté dans une image doit donc être également détecté dans toutes les autres images. C'est pourquoi après avoir détecté les points dans toutes les images d'une même scène, ne seront conservés que les points clés apparaissant dans chaque image aux coordonnées similaires. Lors de l'apprentissage de ce détecteur, les points clés détectés par SIFT et répétés à travers les images seront utilisés comme des exemples positifs à présenter au réseau et donc que l'on aimerait pouvoir détecter de nouveau après l'entraînement.

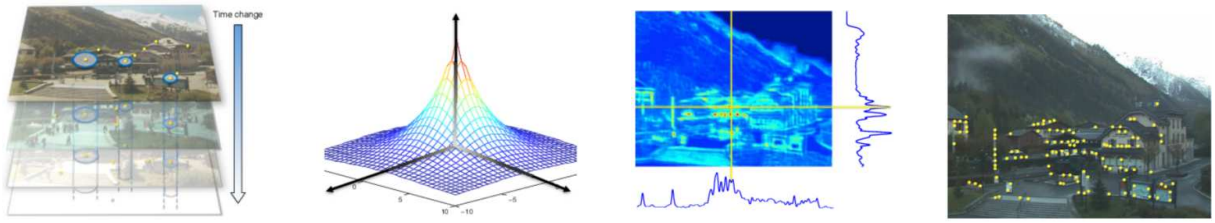
TILDE entraîne un régresseur linéaire par morceaux en vue de prédire une carte de réponse. Si une valeur de cette carte de réponse est supérieure à un seuil, alors il est considéré comme un point clé (voir Figures 1.24(c), 1.24(d)). La fonction objective du réseau TILDE se décompose en trois termes :

- Un terme de classification, $\mathcal{L}_c$, qui permet de séparer les localisations de l'image proches ou éloignées d'un point clé. Ce premier terme repose sur une perte à marge maximale («*max-margin loss*» en anglais).
- Un terme de régularisation de forme, $\mathcal{L}_s$, permettant d'obtenir des maxima locaux aux emplacements des points clés, tel que présenté sur la Figure 1.24(c).
- Un terme de régularisation temporel, $\mathcal{L}_t$, qui va imposer la répétabilité du régresseur

L'objectif final étant la minimisation de la fonction $\mathcal{L}$ regroupant ces trois termes, avec le paramètre ω provenant du régresseur et dont l'équation s'écrit :

$$\underset{\omega}{\text{minimize}}(\mathcal{L}_c(\omega) + \mathcal{L}_s(\omega) + \mathcal{L}_t(\omega)) \tag{1.20}$$

Les résultats présentés par Verdie et al. (2014) montrent qu'un régresseur utilisant des fonctions linéaires par morceaux donne de bien meilleurs résultats qu'avec d'autres régresseurs et des modèles mathématiques comme [Bay et al., 2006 ; Lowe, 2004].



(a) Empilement d'images provenant du même point de vue, avec en jaune, les points détectés par SIFT.

(b) Réponse désirée pour une parcelle d'image positive.

(c) Réponse du régresseur pour une image donnée.

(d) Points clés détectés dans l'image donnée.

FIGURE 1.24 – Vue globale de l'approche TILDE.
Source: Verdie et al. (2014).

LIFT est un réseau extracteur de type «*détecter-puis-décrire*» qui tente d'apprendre les phases de détection, d'estimation d'orientation et finalement de description en une seule chaîne de traitement. Ce réseau comprend trois RNC entraînés dans l'ordre inverse, c'est-à-dire en commençant par le descripteur, puis l'orientateur et finalement le détecteur.

Le détecteur LIFT est une amélioration de TILDE qui se veut être un détecteur robuste aux changements d'illuminations. TILDE a été appris sur un ensemble de différentes scènes comprenant chacune plusieurs images. Chaque image de la même scène est acquise du même point de vue, sans changement d'échelles, mais à des heures et jours différents. Contrairement à TILDE, LIFT a bénéficié de la création d'un nouvel ensemble de données d'entraînement formé d'images à différents points de vue.

Selon Wikipédia, en biologie, la structure acquise à partir de mouvement (angl. «*Structure from Motion*», SfM) «*désigne le phénomène par lequel une personne (et autres créatures vivantes) peut estimer la structure 3D d'un objet ou d'une scène en mouvement à partir de son champ de vision 2D (rétinien)*».

La seconde amélioration provient de la façon d'entraîner le détecteur. Pour permettre à S (la carte de réponse du détecteur) d'avoir des maxima à des endroits autres qu'un emplacement fixe récupéré par la SfM, LIFT veut traiter implicitement cet emplacement en tant que variable latente. La SfM est un algorithme permettant d'estimer la structure 3D d'un objet ou d'une scène en mouvement à partir de son champ de vision 2D, voir Section 2.1.1 pour de plus amples informations. La méthode peut potentiellement découvrir des points plus fiables et plus faciles à apprendre, contrairement à TILDE. À partir d'une carte de réponse S, cette nouvelle localisation x est obtenue grâce à la fonction suivante :

$$\mathrm{x} = \mathrm{softargmax}(S) = \frac{\sum_y \exp(\beta S(y)) y}{\sum_y \exp(\beta S(y))} \tag{1.21}$$

Avec y correspondant aux localisations dans S, et $\beta = 10$ l'hyperparamètre contrôlant le lissage de la fonction. Cette fonction *softargmax* est une approximation de l'opération *argmax* couramment utilisée et permettant de calculer l'indice de valeur maximale dans un tableau

ou un tenseur donné. Cependant, cette opération *argmax* n'est pas différentiable et ne peut donc pas être directement utilisée dans les algorithmes d'optimisation basés sur les gradients, tels que la rétropropagation.

Après avoir ajusté et extrait la nouvelle coordonnée x, une parcelle d'image (ou «*patch*» en anglais) est extraite autour de ce point et fournie à l'orientateur et au descripteur. Au moment de l'entraînement du détecteur LIFT, l'orientateur et le descripteur ont déjà été entraînés (nous reviendrons sur la façon de les entraîner par la suite). Afin d'entraîner le détecteur, TILDE se base sur une architecture siamoise à quatre branches, prenant un quadruplet de parcelles (P^1, P^2, P^3, P^4) en entrée et minimisant la somme de deux fonctions de perte. Dans ce quadruplet, P^1 et P^2 correspondent à des parcelles positives, c'est-à-dire provenant des mêmes localisations au sein d'images du même point de vue. P^3 est une parcelle provenant d'une localisation différente des points clés extraits par SIFT ou la SfM, mais non présente dans chaque image (appelée parcelle négative) et enfin P^4 qui ne représente aucune position distinctive dans l'image et qui est utilisée comme exemple négatif pour entraîner le détecteur.

La première fonction de perte $\mathcal{L}_{class}$ vise à maximiser la classification des scores au sein des parcelles d'images ne correspondant pas. La seconde partie de la fonction de perte $\mathcal{L}_{pair}$, permet de minimiser les distances entre les vecteurs de descriptions pour les paires de parcelles qui correspondent au même point physique dans l'image.

L'entraînement du second réseau requiert le descripteur appris que nous étudierons juste après. L'estimateur d'orientation se base sur la minimisation de la distance entre les vecteurs de descriptions de deux parcelles d'images différentes, mais provenant de la même localisation. Cette fonction de perte s'écrit de la façon suivante :

$$\mathcal{L}_{orientation}(P^1, P^2) = ||h_p(P^1) - h_p(P^2)||_2 \tag{1.22}$$

avec h_p représentant le descripteur de point.

La partie description est apprise par minimisation de la somme d'une fonction de perte entre des parcelles d'images correspondantes (P^1, P^2) et entre des parcelles ne correspondant pas aux mêmes positions (P^1, P^3). La fonction de perte peut être définie comme une fonction de perte charnière ou perte à marge maximale («*hinge loss*» en anglais) sur les distances euclidiennes entre les vecteurs de descriptions. La fonction peut s'écrire :

$$\mathcal{L}_{desc}(P^k, P^l) = \begin{cases} ||h_p(P^k) - h_p(P^l)||_2 & \text{pour les paires positives et,} \\ \max(0, C - ||h_p(P^k) - h_p(P^l)||_2) & \text{pour les paires négatives} \end{cases} \tag{1.23}$$

Avec $C = 4$ correspondant à la marge à appliquer entre deux éléments provenant de localisations différentes et $||.||_2$ la distance euclidienne.

La méthode TILDE n'inclut pas l'invariance d'échelle lors du processus d'apprentissage. Durant la phase d'inférence (la phase suivant l'étape d'entraînement), la répétabilité du détecteur à différentes échelles est permise en appliquant le réseau à différentes résolutions de l'image.

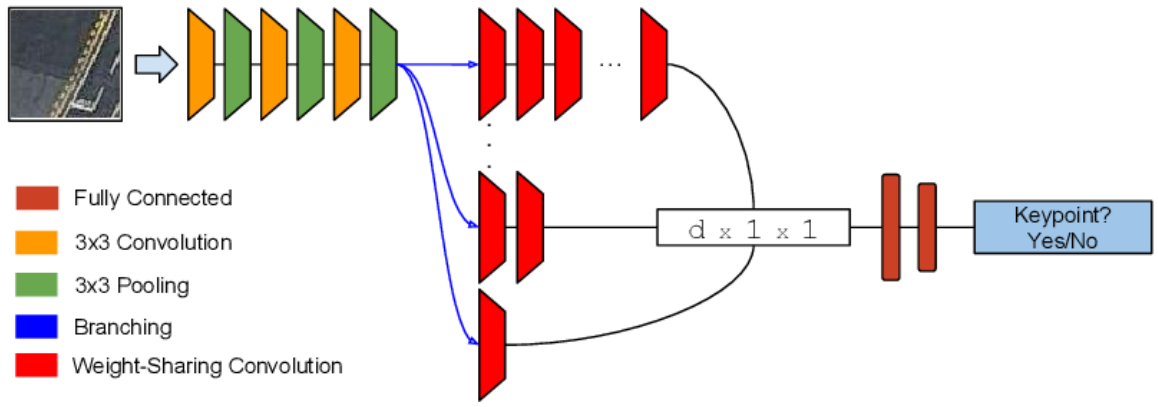


FIGURE 1.25 – Architecture du réseau L2DMK.
Source: Altwaijry et al. (2016).

1.5.3 L2DMK : Learning to Detect and Match Keypoints with Deep Architectures

Learning to Detect and Match Keypoints with Deep Architectures (L2DMK) [Altwaijry et al., 2016] est un extracteur multiéchelle du type *détecter-puis-décrire* et basé sur deux réseaux indépendants : un réseau de détection (voir architecture Figure 1.25) et un réseau de description basé sur des triplets, que nous ne redécrirons pas ici.

Dans cette approche, un ensemble de données est mis en place à l'aide de la SfM à partir d'images aériennes couvrant des zones de $15km^2$ autour de Boston, Massachusetts. Des parcelles d'images sont extraites dans cinq échelles différentes : $S = \{64, 96, 128, 192, 256\}$. L'ensemble des extraits d'images générés P est noté $P = p_i : (x_i, s_i, k_i)$ avec $k \in \{-1, 1\}$, et où p_i est une parcelle avec x_i un pixel ; s_i indique l'échelle de la parcelle et p_i correspond à l'étiquette (le label associé à la zone extraite).

Le réseau de détection (voir Figure 1.25) apprend une fonction non linéaire capable d'identifier si une zone de l'image contient un point clé. Le réseau est composé de différentes couches de convolution et pooling, mais également d'un mécanisme de ramification dépendant de l'échelle (flèches bleues dans la Figure 1.25), puis de deux couches entièrement connectées pour la classification.

Pendant l'inférence, le réseau est transformé en un modèle entièrement convolutionnel qui produit une carte de réponse, où chaque valeur représente un score de détection pour une région de l'image. Les points clés sont extraits en utilisant une technique appelée suppression non maximale.

1.5.4 SuperPoint

SuperPoint est un détecteur de points clés qui utilise un réseau de neurones à convolutions pour détecter des points clés précis dans une image. Ce détecteur est capable d'extraire des points clés rapidement et de manière fiable, même dans des conditions difficiles telles que

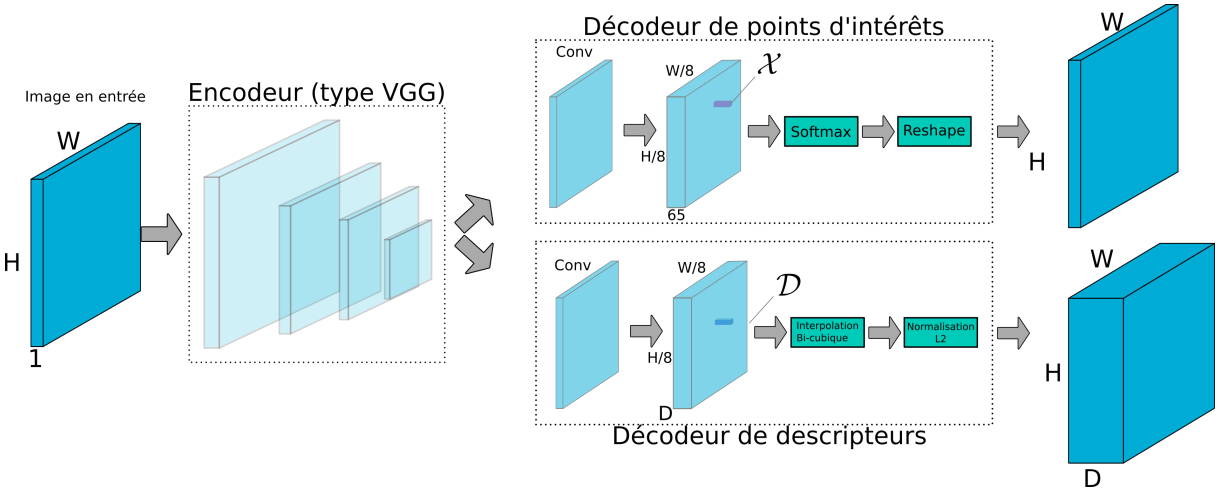


FIGURE 1.26 – Architecture du réseau superpoint.

des images floues ou présentant des changements d'illumination. Il utilise une approche similaire à celle de TILDE dans la création des données, mais bénéficie de la puissance d'un réseau neuronal profond et entièrement convolutif. Superpoint qui met en place la partie détection, mais également la description fait partie du type du type *détecter-et-décrire*.

Le réseau SuperPoint [DeTone et al., 2017] désigne un réseau dont l'entraînement comporte plusieurs étapes. Une première phase ne prend en compte que l'entraînement du détecteur et utilise des images générées de manière automatique par un programme, cette première phase d'entraînement est appelée *magicpoint*. La seconde étape, appelée *adaptation homographique*, consiste à utiliser le détecteur magicpoint appris sur des données synthétiques pour créer un nouvel ensemble de données d'entraînement sur des images naturelles, nous détaillerons cette génération de données dans la suite de cette thèse. La dernière phase consiste en un nouvel entraînement du réseau en joignant l'étape de description à celle de détection.

L'architecture du réseau SuperPoint est constituée de deux parties principales : un réseau de détection et un réseau de description (voir Figure 1.26). L'architecture est composée tout d'abord d'un encodeur de type very deep convolutional networks for large-scale recognition (fran. «*réseau convolutionnel très profond pour la reconnaissance à grande échelle*», VGG) [Simonyan et Zisserman, 2014] permettant de réduire la dimensionnalité des images en entrée. Cet encodeur comporte huit couches de convolution 3×3 de taille 64-64-64-64-128-128-128-128. Toutes les deuxièmes couches de convolution sont suivies d'une couche de *maxpooling* de taille 2×2 et enfin chaque couche de convolution du réseau est suivie d'une activation non linéaire ReLU et d'une normalisation par batch. Le résultat de l'encodeur est ensuite utilisé à la fois pour la partie détection et par la partie description.

Pour la partie détection du réseau, chaque valeur en sortie correspond à une probabilité d'être un point d'intérêt pour le pixel qui correspond dans l'image en entrée. La fonction de perte mise en place pour cette étape de détection est une fonction d'entropie croisée entre

la sortie du décodeur et la vérité terrain. Elle s'écrit :

$$\mathcal{L}_p(\mathcal{X}, \mathcal{Y}) = \frac{1}{H_c W_c} \sum_{h=1,w=1}^{H_c,W_c} (l_p(\mathsf{x}_{hw}; y_{hw})) \tag{1.24}$$

où H_c et W_c correspondent à la nouvelle taille de l'image suite au passage de l'encodeur. $\mathsf{x}_{hw} \in \mathcal{X}$ correspond à chaque vecteur à travers les canaux en sortie de l'encodeur (voir Figure 1.26) ; $\mathcal{Y}$ est la vérité terrain fournie au réseau (y_{hw} étant une entrée individuelle de $\mathcal{Y}$). Enfin :

$$l_p(\mathsf{x}_{hw}; y_{hw}) = -\log\left(\frac{\exp(x_{hwy})}{\sum_{k=1}^{65} \exp(\mathsf{x}_{hwk})}\right) \tag{1.25}$$

La partie description permet d'obtenir un descripteur par pixel dans l'image en entrée. Cette branche de superpoint est entraînée à l'aide d'une double fonction de perte charnière (angl. «*hinge loss*») décrite dans la Section 1.17. La première perte charnière rassemble les descripteurs correspondant à la même localisation, et la seconde perte charnière discrimine les descripteurs provenant de localisations différentes.

1.5.5 D2-net : A Trainable CNN for Joint Description and Detection of Local Features

Le réseau D2-Net [Dusmanu et al., 2019] est une méthode de détection et de description de points clés qui utilise une approche de «*détection-et-description*» (voir Figure 1.27). Au lieu d'adopter une méthode classique en deux étapes de «*détection-puis-description*», cette méthode partage la même représentation sous-jacente pour le détecteur et le descripteur.

La méthode d'apprentissage, de l'extracteur de points clés, appelée D2-Net est similaire à la méthode précédente SuperPoint en ce sens que la représentation entre la détection et la description des points clés est partagée. Toutefois, contrairement à SuperPoint, D2-Net utilise un seul entraînement conjoint qui optimise simultanément les deux parties, alors que SuperPoint divise l'apprentissage en deux branches indépendantes.

La Figure 1.27 montre que des vecteurs de description sont extraits à chaque emplacement (h, w) à travers tous les canaux. En ce qui concerne la détection des points clés, chaque canal $\mathcal{D}^k$ (avec k étant le nombre de canaux) peut être considéré comme une carte de réponse similaire aux DoG utilisés dans SIFT. Ainsi, la détection a lieu à travers toutes ces cartes de réponse.

D2-Net utilise deux techniques de détections de points, la première est utilisée lors de l'entraînement et la seconde lors de l'inférence du réseau. L'inférence d'un réseau de neurones consiste à utiliser le modèle de réseau de neurones entraîné pour effectuer des prédictions sur de nouvelles données. Durant la phase d'inférence, une première étape de suppression-non maximal est mise en place dans chacun des canaux. Puis, la détection prend place à travers

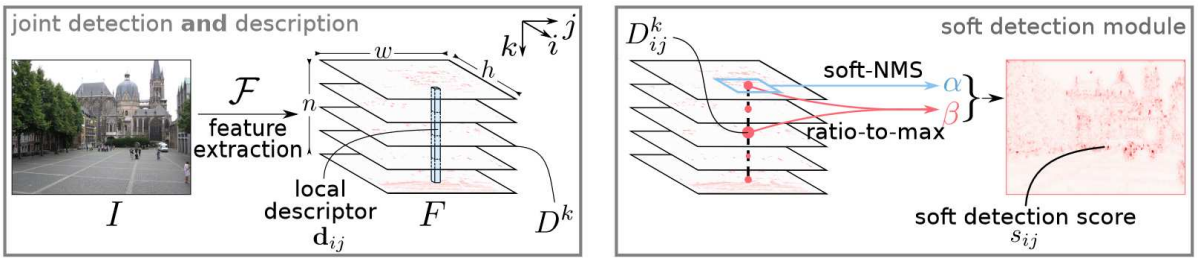


FIGURE 1.27 – Architecture de type «*détecter-et-décrire*» proposée par le réseau D2-Net.
Source: Dusmanu et al. (2019).

tous les canaux à une même localisation (h, w).

Pendant la phase d'entraînement, la méthode de détection précédente a été modifiée pour permettre la rétropropagation du gradient. Cela a été fait en introduisant la fonction soft local-max pour obtenir un score de probabilité α_{ij}^k, en calculant un score appelé «*ratio-to-max*» β_{ij}^k à travers les canaux, en maximisant le produit de ces deux scores pour toutes les cartes de caractéristiques k, puis en normalisant l'image pour obtenir le score de détection final, noté s_c. Cela est illustré dans la Figure 1.27.

Pour entraîner le modèle proposé, optimisant conjointement les étapes de détection et de description, une fonction de perte appropriée est mise en place. Dans ce but, une extension de la fonction de perte par triplets initialement utilisée pour l'étape de description est proposée (voir équation 1.19), afin de prendre également en compte l'étape de description. Dans le but de prendre en compte le détecteur et d'améliorer la répétabilité de ce dernier, un terme est ajouté à cette perte par triplets et peut s'écrire :

$$\mathcal{L}(I_1, I_2) = \sum_{c \in \mathcal{C}} \frac{s_c^{(1)} s_c^{(2)}}{\sum_q \in \mathcal{C} s_q^{(1)} s_q^{(2)}} * \text{perte par triplets} \tag{1.26}$$

Où $s_c^{(1)}$ et $s_c^{(2)}$ sont les scores de détection aux points A et B dans les images I_1 et I_2. L'ensemble des correspondances entre I_1 et I_2 est noté $\mathcal{C}$. Cette fonction de perte peut être vue comme une moyenne pondérée des fonctions de pertes des triplets, ce qui revient à augmenter le score de détection si la fonction de perte est inférieure à la marge et diminuer le score dans le cas contraire.

1.5.6 R2D2 : repeatable and reliable detector and descriptor

Selon [Revaud, Weinzaepfel et al., 2019 ; Streiff et al., 2021], «*la répétabilité des points clés est un problème qui ne peut être résolu par un entraînement supervisé standard. En fait, l'utilisation de la supervision se résume essentiellement dans ce cas à imiter un détecteur existant plutôt que de découvrir des points clés potentiellement meilleurs*». C'est pourquoi R2D2 considère la répétabilité comme une tâche autosupervisée et entraîne le réseau de

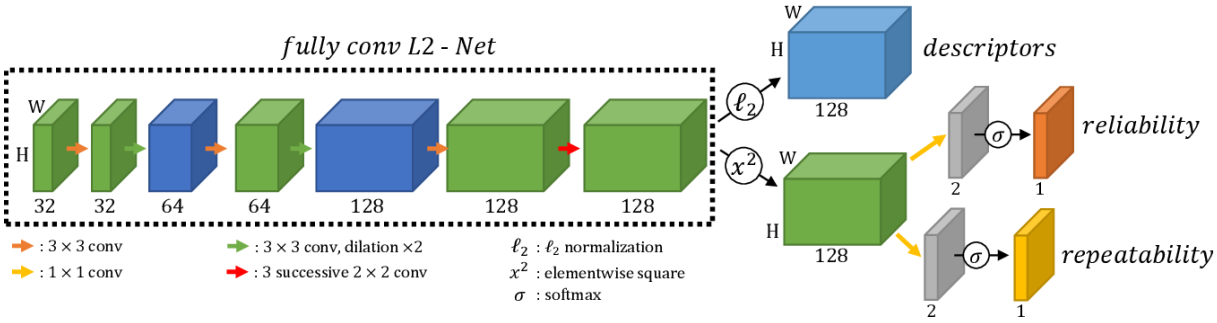


FIGURE 1.28 – Architecture de type «*détecter-et-décrire*» du réseau r2d2.
Source: Revaud, Weinzaepfel et al. (2019).

manière que les positions des maxima locaux dans S soient covariantes aux transformations d'images naturelles telles que les changements de point de vue ou d'éclairage. La Figure 1.28 représente l'architecture du réseau R2D2.

Le réseau R2D2 se différencie des architectures précédentes en ce qu'il n'essaie pas de définir un point clé de manière arbitraire comme un maximum local à travers les canaux ou la carte de réponse du détecteur. Au lieu de cela, il estime la répétabilité et la fiabilité des points clés en deux étapes distinctes.

Pour décrire l'approche R2D2, introduisons quelques concepts. Le réseau utilisé, nommé L2-Net [Tian et al., 2017], maintient la même taille d'entrée pour chaque couche, produisant ainsi un volume de sortie similaire à celui de l'entrée. Il est entraîné à produire trois cartes de réponse pour une image I de dimensions $H \times W$" :

- Carte 3D de réponse au descripteur : $X \in \mathbb{R}^{H \times W \times D}$ qui correspond à un ensemble dense de descripteurs, un par pixel. D représente la dimension d'un descripteur.
- Carte de réponse au détecteur : $S \in [0,1]^{H \times W}$ qui permet de générer des points clés épars, mais répétables.
- Carte de réponse de fiabilité : $R \in [0,1]^{H \times W}$ qui indique la fiabilité du descripteur $X_{i,j}$, avec (i,j) les pixels de l'image.

Nous allons maintenant décrire les différentes fonctions de perte utilisées lors de l'apprentissage du réseau.

Répétabilité - Il y a deux fonctions de perte utilisées dans l'étape de détection : $\mathcal{L}_{cosim}$ et $\mathcal{L}peaky$. La première fonction (voir équation 1.27) est maximisée lorsque les cartes de réponses S et S' des images I et I' sont identiques et que leur maxima corresponds exactement. La seconde fonction, $\mathcal{L}peaky$ (voir équation 1.28), permet de maximiser les pics locaux de la carte de répétabilité afin d'éviter que la première fonction $\mathcal{L}cosim$ ne minimise de manière triviale les cartes S et S' en les rendant constantes.

La première fonction de perte calcule la similarité cosinus moyenne sur de nombreuses parcelles d'images, ceci permet de calculer la similarité de deux vecteurs à n dimensions en

déterminant le cosinus de leur angle. L'ensemble des parcelles de l'image est noté $\mathcal{P} = p$, et donc $S[p] \in \mathbb{R}^{N^2}$ dénote les parcelles de taille $N \times N$ extraites de S. La fonction s'écrit :

$$\mathcal{L}_{cosim}(I, I', U) = 1 - \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} cosim(S[p], S'_U[p]) \tag{1.27}$$

Où U correspond à la vérité terrain permettant de mettre en commun chaque pixel entre les deux images. La seconde fonction évite au réseau une minimisation triviale de S et S'_U par des constantes et s'écrit :

$$\mathcal{L}_{peaky}(I) = 1 - \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} \left(\max_{(i,j) \in p} S_{ij} - \operatorname*{mean}_{(i,j)} S_{ij} \right) \tag{1.28}$$

Finalement, la fonction de perte de répétabilité s'écrit :

$$\mathcal{L}_{repetabilite}(I, I', U) = \mathcal{L}_{cosim}(I, I', U) + \frac{1}{2}(\mathcal{L}_{peaky}(I) + \mathcal{L}_{peaky}(I')) \tag{1.29}$$

Carte de fiabilité et descripteurs - Le réseau estime la fiabilité de la parcelle p_{ij} en utilisant la précision moyenne (PM), ou «*Average Précision*» en anglais, optimisée pour l'apprentissage dans [He et al., 2018] et noté $\widetilde{AP}$. Généralement, pour calculer cette PM, après avoir calculé les descripteurs pour chaque parcelle d'images au sein d'un mini-lot, il faut calculer la matrice des distances euclidiennes entre tous les descripteurs. Chaque ligne de la matrice peut être interprétée comme les distances entre un descripteur requête d'une première parcelle de l'image et toutes les autres d'une seconde image. L'entraînement consiste donc à maximiser la précision moyenne calculée pour chaque requête q dans le lot B sur l'ensemble du lot. Cette PM est maximisée lorsque les parcelles sont les plus fiables possibles. L'entraînement revient donc à maximiser l'PM sur l'ensemble des parcelles (noté B) dans le mini-lot d'images. La fonction s'écrit :

$$\mathcal{L}_{AP} = \frac{1}{B} \sum_{ij} 1 - \widetilde{AP}(pij) \tag{1.30}$$

Dans l'article R2D2, le calcul des descripteurs est automatiquement exécuté à l'aide d'un réseau complètement convolutionnel et où une parcelle de taille M est définie autour de chaque pixel qui peut être comparée à toutes les autres parcelles de la seconde image.

Selon [Revaud, Weinzaepfel et al., 2019], «*un descripteur local est extrait pour chaque localisation, mais toutes les localisations ne sont pas toutes intéressantes [...] Il devient donc clair que l'optimisation du descripteur de points même dans de telles régions d'image peut nuire aux performances*». Toujours selon les auteurs du réseau R2D2, même les régions bien texturées sont également connues pour leur caractère peu fiable en raison de leur nature sémantique, comme les feuillages d'arbres ou les vagues de l'océan. Il est possible de

trouver de telles régions dans nos images médicales 3D, avec par exemple : les côtes, les vertèbres, etc. Il devient donc évident que l'optimisation forcée du descripteur de parcelles, même dans les régions sans signification de l'image, pourrait entraver l'entraînement et les performances en temps d'exécution.

C'est pourquoi R2D2 propose d'améliorer la PM pour éviter au réseau de gaspiller des ressources sur des régions non distinctives à l'aide d'un terme de fiabilité (noté R), et selon l'équation :

$$\mathcal{L}_{AP,R} = \frac{1}{B}\sum_{ij}\left(1 - \widetilde{AP}(pij)R_{ij} + k(1 - R_{ij})\right) \tag{1.31}$$

Où $k \in [0, 1]$ est un paramètre représentant le seuil de l'PM au-dessus duquel une parcelle est considérée comme fiable et R_{ij} le score de fiabilité à la position i, j et compris entre $[0, 1]$.

1.6 Extracteurs de points clés 3D

Cette Section se concentrera sur l'étude de plusieurs détecteurs et/ou descripteurs de points clés qui sont utilisés en 3D. Certains des extracteurs de points clés couramment utilisés en 2D, ont été étendus pour fonctionner en 3D, [Agier et al., 2016 ; Flitton et al., 2010 ; Rister et al., 2017 ; Scovanner et al., 2007 ; Sipiran et Bustos, 2011]. Cependant, en ce qui concerne le domaine médical, il existe encore peu d'explorations dans le développement des extracteurs de points clés en 3D.

Les images médicales en 3D offrent des avantages pour les réseaux 3D par rapport aux réseaux 2D. Elles contiennent des données volumétriques qui captent les informations spatiales sur trois dimensions. En exploitant le contexte 3D complet, les réseaux 3D permettent une analyse plus précise et complète par rapport aux approches 2D qui ne considèrent que des tranches individuelles. De plus, les réseaux 3D capturent les relations entre les tranches voisines et bénéficient de champs récepteurs plus larges, améliorant ainsi la segmentation, la classification et la détection dans le domaine de l'imagerie médicale.

1.6.1 SIFT 3D

La première étape est similaire à [Lowe, 2004], avec le calcul les DoG dans les images en ajoutant d'une troisième dimension z dans l'équation 1.11.

Finalement, les extréma de la fonction DoG sont extraits de manière similaire à la version en deux dimensions. Cependant, la différence principale réside dans la méthode d'extraction utilisée, qui implique la comparaison d'un voisinage local de $3 \times 3\times$ voxels, soit les 26 voxels adjacents, ainsi qu'avec les 27 voxels voisins de l'échelle correspondante $(k-1, k+1)$.

Diverses méthodes existent pour le calcul d'orientation [Flitton et al., 2010 ; Scovanner

et al., 2007]. Cette dernière méthode calcule les histogrammes vectoriels comme une combinaison de l'orientation des gradients des sous-volumes, dans un espace volumique (ou une région spatio-temporelle 3D, c'est-à-dire trois images 2D consécutives).

Finalement, pour calculer le descripteur, le volume est divisé en 8 sous-régions entourant le point clé (des régions de $4 \times 4 \times 4$ voxels sont utilisées) en vue de calculer les histogrammes.

1.6.2 SURF 3D

Une extension de SURF3D a été proposée par [Agier, 2017 ; Agier et al., 2016], où plusieurs concepts ont été étendus à la 3D pour permettre la détection et la description de points clés dans des images médicales volumiques.

Selon la thèse de [Agier, 2017], la première généralisation en 3D consiste à générer l'image intégrale et à calculer la somme des intensités d'une partie d'un volume. Cette généralisation triviale permet de maintenir la rapidité de calcul de sous-blocs d'intensité de volume en passant d'une complexité de calcul en $O(4)$ pour la 2D à une complexité en $O(8)$ pour la 3D. Contrairement à un calcul naïf qui nécessite trois boucles pour itérer sur chaque élément dans les trois dimensions, avec une complexité en $O(n^3)$, où n est la taille de la région à calculer. Le somme d'une région de l'image intégrale en 3D s'écrit :

$$\begin{aligned} \text{Boite3D}(x_1, x_2, y_1, y_2, z_1, z_2) = &+ S(x_2, y_2, z_2) - S(x_2, y_2, z_1) \\ &+ S(x_2, y_1, z_2) - S(x_1, y_2, z_2) \\ &- S(x_1, y_1, z_2) - S(x_1, y_2, z_1) \\ &+ S(x_2, y_1, z_1) + S(x_1, y_1, z_1) \end{aligned} \tag{1.32}$$

Où S correspond à l'image intégrale et (x_1, y_1, z_1) et (x_2, y_2, z_2) sont deux points décrivant les bords d'un bloc dans lequel on aimerait calculer la somme de l'intensité de chaque voxel.

La deuxième extension consiste à étendre la matrice hessienne et le calcul de son déterminant à la 3D. Le déterminant de la matrice hessienne s'écrit :

$$det(H_{approx}) = D_{xx}D_{yy}D_{zz} + 2\omega^3 D_{xy}D_{yz}D_{xz} - \omega^2 D_{xx}D_{yz}D_{yz} - \omega^2 D_{yy}D_{xz}D_{xz} - \omega^2 D_{zz}D_{xy}D_{xy} \tag{1.33}$$

D'après [Agier et al., 2016], pour maintenir une taille de vecteur de description raisonnable, il est proposé de limiter la taille des blocs à $2 \times 2 \times 2$ plutôt que les blocs de $4 \times 4 \times 4$ initialement prévus pour le descripteur. Cette modification permet de réduire la taille du vecteur de description de 384 à 48.

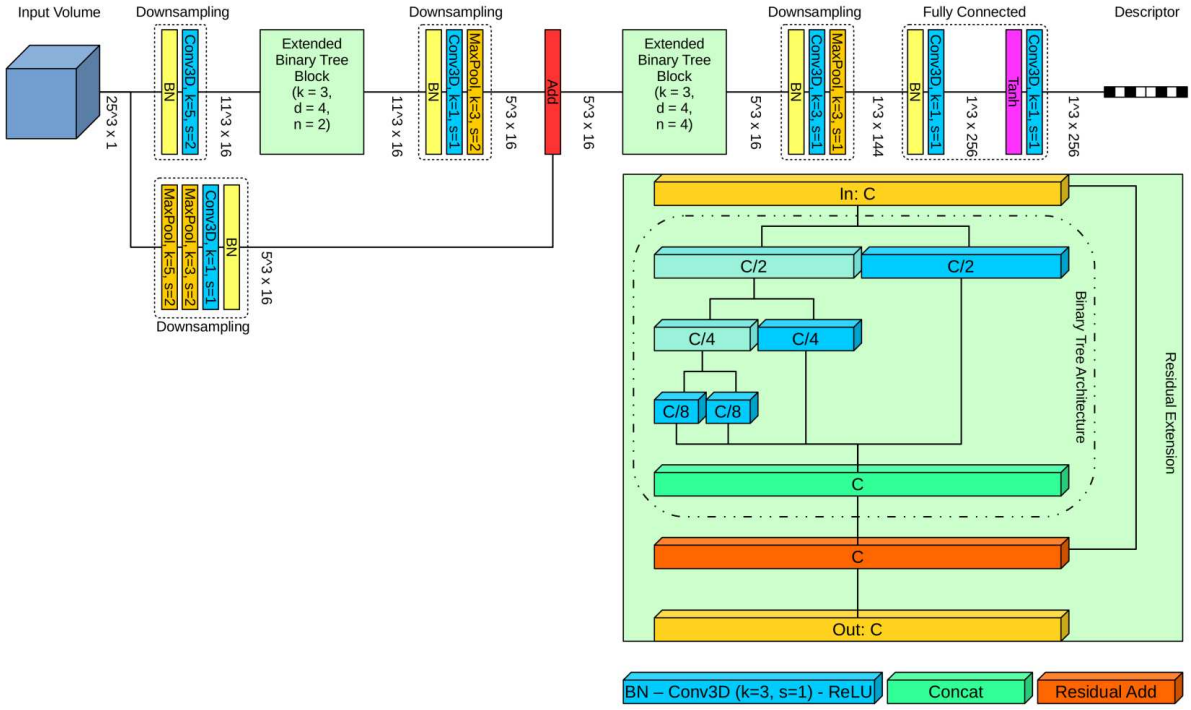


FIGURE 1.29 – Extension 3D de l'architecture des blocs d'arbres binaires.
Source: Blendowski et Heinrich (2018)

1.6.3 Descripteur binaire par apprentissage 3D

L'article Blendowski et Heinrich (2018) a pour objectif de créer un descripteur binaire pour caractériser les points clés issus d'images médicales volumétriques. Ce descripteur est basé sur une extension en 3D des architectures d'arbres binaires [Zhang et al., 2017]. D'après les auteurs de l'article, cette structure a montré d'excellents résultats dans la tâche de classification 2D en réduisant considérablement le nombre de paramètres d'apprentissage. Ainsi, elle offre une base solide pour les réseaux de neurones convolutifs 2D.

Le réseau (décrit dans la Figure 1.29) utilise la fonction d'activation hyperbolique (tanh) plutôt que ReLU afin de limiter les valeurs de sortie entre $[-1, 1]$. Bien que cette plage de valeurs puisse être obtenue facilement en utilisant la fonction $sign$ (décrite dans l'équation 1.34), une quantification directe peut entraîner une baisse importante des performances selon [Blendowski et Heinrich, 2018]. Pour remédier à ce problème, une fonction d'activation $\mathcal{L}quant$ est ajoutée à une triple fonction de perte charnière. Cette fonction $\mathcal{L}quant$ pénalise les éléments des descripteurs dont la distance absolue à 1 est supérieure à 0.

$$sign(x) = \begin{cases} -1 & \text{si } x < 0 \\ 0 & \text{si } x = 0 \\ 1 & \text{si } x > 0 \end{cases} \tag{1.34}$$

Méthode	Détecteur			Descripteurs		
	Multi échelle	Invariance en rotation	Invariance à l'illumination	Invariance en échelle	Invariance en rotation	Invariance à l'illumination
Harris	—	✓	✓	—	—	—
FAST	—	—	✓	—	—	—
BRISK	✓	—	✓	✓	✓	✓
ORB	—	✓	✓	✗	✓	✓
FREAK	—	—	—	✗	✗	✓
BRIEF	—	—	—	✗	✗	✓
KAZE	✓	✓	✓	✓	✓	✓
SIFT	✓	✓	✓	✓	✓	✓
SURF	✓	✓	✓	✓	✓	✓
Triplet-Net	—	—	—	✓	✓	✓
TILDE	✓	✓	✓	—	—	—
LIFT	✓	✓	✓	✓	✓	✓
SuperPoint	✓	✓	✓	✓	✓	✓

TABLEAU 1.1 – Tableau résumant les différents détecteurs et descripteurs de points clés.
Source: Cuiyin et al. (2021)

1.6.4 Critiques et Conclusions

Au cours des deux sections précédentes, nous avons examiné divers extracteurs de points, qu'ils soient de nature mathématique ou basés sur l'apprentissage. Le Tableau 1.1 offre un récapitulatif des différents détecteurs et/ou descripteurs de points clés, avec les principales caractéristiques qu'ils offrent. Nous avons décidé de présenter ces trois caractéristiques en les mettant en relation avec le contexte de l'imagerie médicale. Tout d'abord, ils doivent être capables de détecter de manière fiable et précise, même lorsque les tailles des patients ou les positions de capture d'images varient, d'où l'invariance en rotation et la détection multiéchelle. De plus, l'invariance à l'illumination est essentielle pour résoudre le défi de la multimodalité des images médicales.

Les méthodes classiques d'extraction de points clés ont rapidement été surpassées par les méthodes d'apprentissage. Des méthodes classiques telles que SURF et SIFT ont déjà été étendues à la troisième dimension et ont donné d'excellents résultats. C'est pourquoi l'extension et l'utilisation de méthodes neuronales en 3D semblent prometteuses.

Dans un premier temps, pour confirmer l'hypothèse selon laquelle il est possible d'utiliser des réseaux de neurones pour ce type d'applications en 3D, nous souhaitons mettre en place un descripteur de points clés. Pour cette première étape, plusieurs choix s'offrent à nous, mais les réseaux à triplets ou à listes ont montré les meilleures performances, c'est pourquoi nous nous tournons vers eux.

Ensuite, nous souhaitons étendre un extracteur de points 2D à la 3D pour démontrer une fois de plus que la détection est possible en 3D. Pour cela, nous avons identifié plusieurs réseaux en 2D parmi lesquels nous avons choisi de nous concentrer sur le réseau Superpoint, qui utilise

un apprentissage entièrement supervisé, et le réseau R2D2, qui utilise un apprentissage supervisé de manière distante. Ces deux réseaux mettent en œuvre la détection multiéchelle ainsi que l'invariance en rotation et à l'illumination.

2

Données et Augmentation de données

Sommaire

L'apprentissage neuronal est devenu une méthode couramment utilisée pour la résolution de problèmes dans de nombreux domaines, y compris le recalage de données médicales. L'un des principaux facteurs de succès de l'apprentissage neuronal est la disponibilité de données d'entraînement de qualité pour les algorithmes d'apprentissage. Dans le domaine médical, l'acquisition des images de haute qualité est particulièrement importante, car les erreurs de classification ou de prédiction peuvent avoir des conséquences graves pour les patients.

Les images médicales sont des informations essentielles pour les professionnels de la santé afin de diagnostiquer et traiter les maladies, ainsi que pour améliorer la qualité des soins prodigués aux patients. Dans le domaine de la recherche médicale, l'exploitation des données médicales est cruciale pour étudier et comprendre les mécanismes biologiques des maladies, développer de nouveaux traitements et améliorer l'état de santé de population. Cependant, l'obtention de ces données est souvent difficile en raison de problèmes de confidentialité et de la rareté des échantillons.

Afin de pallier ce problème, l'augmentation d'échantillons médicaux a été proposée comme une solution prometteuse. Elle consiste à générer de nouvelles images à partir d'un ensemble de données existant en appliquant des techniques de traitement du signal, d'apprentissage automatique et d'intelligence artificielle. Cette technique permet d'augmenter la taille de l'ensemble de volumes et de créer des données supplémentaires pour des cas rares ou peu fréquents.

Ce chapitre de thèse se concentre sur les méthodes de génération de données médicales pour l'apprentissage neuronal, en mettant l'accent sur les défis spécifiques liés à la collecte de données médicales et les techniques utilisées pour améliorer la qualité des images générées. Nous examinerons également les résultats de diverses expériences menées pour évaluer l'efficacité de ces méthodes, ainsi que les implications pour la recherche future dans ce domaine.

2.1 Images 2D et Données médicales

Nous allons analyser dans un premier temps les données utilisées dans les réseaux extracteurs de points en 2D. Nous analyserons dans un second temps les données qui sont à notre disposition en vue d'effectuer nos apprentissages et nos tests. Nous emploierons indifféremment les termes : base de données, ensemble de données, ou encore jeu de données.

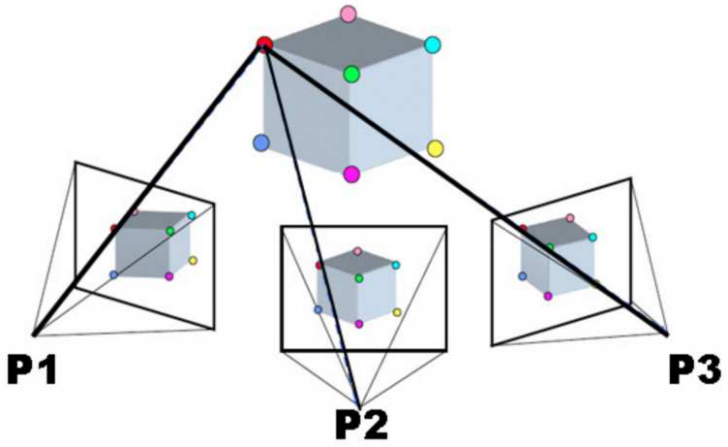


FIGURE 2.1 – Schéma simplifié du procédé de la Structure From Motion (SfM).
Source: Wikipédia.

2.1.1 Structure From Motion (SfM) et Données d'apprentissage 2D

Nous allons maintenant évoquer les différents ensembles de données d'apprentissage employés dans les réseaux de neurones qui seront utilisés dans la suite de cet état de l'art. Nous allons également étoffer la définition de la SfM [Hartley et Zisserman, 2004b] (donnée dans la Section 1.5.2) qui est une méthode permettant d'extraire des points clés répétés à travers plusieurs images.

Structure From Motion (SfM) - La SfM vise à trouver des correspondances à travers plusieurs images représentants la même scène depuis différents points de vue. Ces correspondances sont extraites à l'aide de détecteurs comme SIFT et mises en commun à travers les différents points de vue. L'objectif étant de déterminer un ensemble de points de repère, dont les emplacements sont connus.

Les trajectoires des caractéristiques extraites sont ensuite utilisées pour reconstruire leurs positions 3D et le mouvement de la caméra. Ceci dans un but de reconstruction 3D d'objet (voir Figure 2.1).

Cette technique est employée dans le domaine de la détection par apprentissage pour déterminer de façon précise les correspondances existantes entre différentes images représentant le même objet ou la même scène.

Données d'apprentissages - Dans le domaine de l'apprentissage de détecteurs et descripteurs, quatre jeux de données sont généralement utilisés (Voir Figure 2.2). Nous allons ici donner quelques exemples d'images tirées de ces jeux de données.

Le premier ensemble d'images, Aachen Day-Night datasets [Sattler et al., 2012], contient plusieurs centaines de scènes et chacune d'entre elles contient environ 8 prises de vues différentes. La SfM sera généralement utilisée ici pour générer les correspondances exactes entre chacune des images.

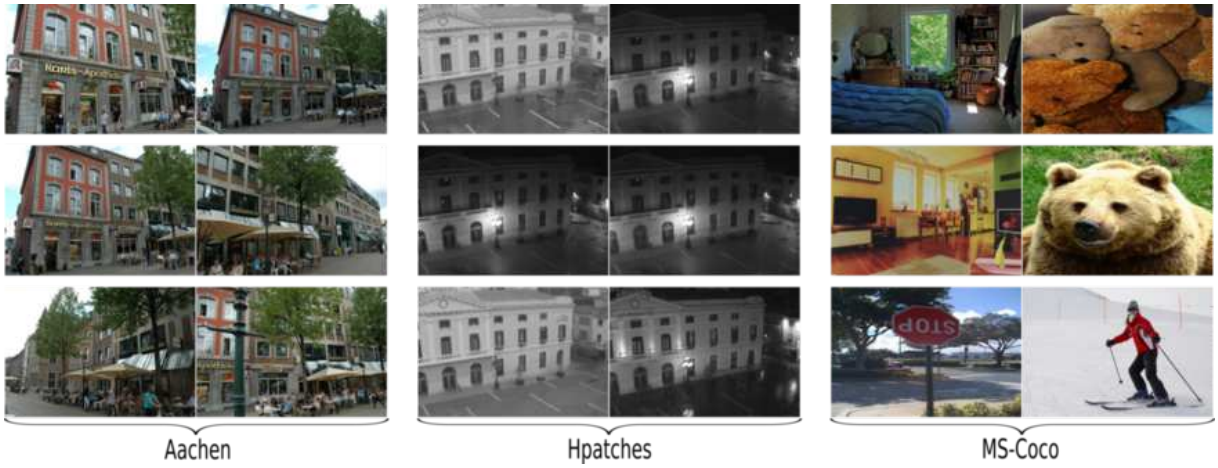


FIGURE 2.2 – Exemple de trois jeux de données utilisés dans la suite de cet état de l'art.
Source: Images tirées des jeux de données MS-Coco, HPatches et Aachen.

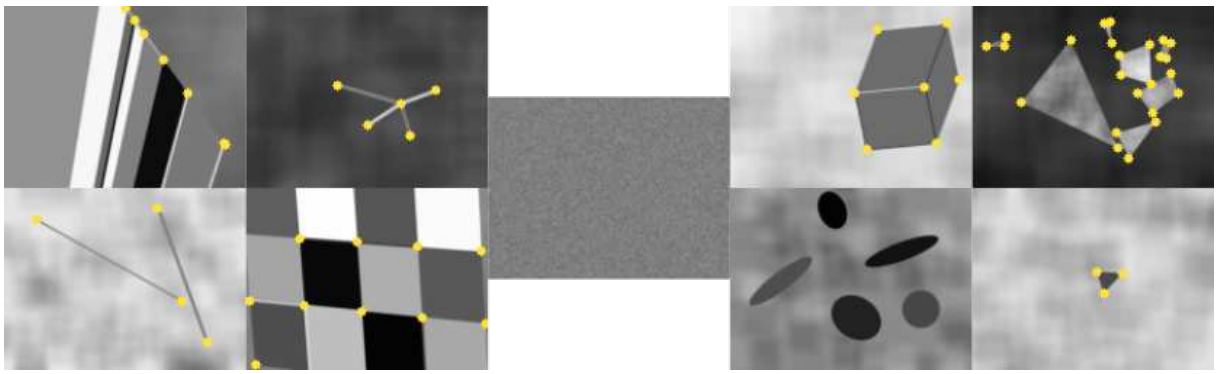


FIGURE 2.3 – Exemple d'images générées par les créateurs du réseau superpoint, présentant diverses formes simples telles que : des droites, des cubes, des polygones et des sphères, ainsi que des images présentant un bruit gaussien sans forme particulière. Les points jaunes sur ces images représentent les points correspondant à des zones spécifiques des formes générées et servent de vérité terrain lors des entraînements.

Le second jeu de données, HPaches [Lin et al., 2014], contient également des prises de vues (différentes ou similaire) de plusieurs scènes. Contrairement au premier jeu de données, celui-ci est fourni avec les transformations exactes entre chaque image.

Le dernier ensemble, MS-COCO [Balntas et al., 2017], contient plusieurs milliers d'images de scènes différentes.

Nous allons aussi présenter un exemple d'images issues d'un jeu de données d'apprentissage récent créé par les développeurs de Superpoint (voir la partie Section 1.5.4). Ces images sont constituées de formes géométriques simples en deux dimensions, telles que des quadrilatères, des triangles, des lignes et des ellipses. Chaque image est accompagnée d'un fichier qui décrit précisément les emplacements des points clés, tels que les coins et les bords des formes (voir la Figure 2.3).

2.1.2 Données synthétiques et médicales volumétriques

Nous disposons de deux ensembles de données : le premier nommé VISCERAL (voir [Langs et al., 2013]), et le second provenant des hôpitaux de la ville de Saint-Étienne. Dans les deux prochaines sous-parties, nous allons décrire ces deux jeux de données. Nous ajouterons une dernière partie détaillant un jeu de données synthétique créé d'une façon similaire à superpoint [DeTone et al., 2017].

Jeu de données VISCERAL - La première base de données nommée VISCERAL est un ensemble de données pour l'évaluation de la segmentation et du recalage d'images médicales. Elle est créée par un ensemble d'établissements et de laboratoires travaillant en coopération, et elle contient des données d'IRM et de scanner de différents organes tels que le foie, les poumons et les reins. Les données proviennent du CHU (Centre hospitalier universitaire) de Heidelberg [1]. Les images sont annotées pour permettre l'évaluation quantitative des méthodes de segmentation et de recalage. VISCERAL a été utilisé pour de nombreuses compétitions et évaluations de recherche dans le domaine de la vision par ordinateur en médecine.

L'ensemble de données VISCERAL contient des images (ou volumes), de deux modalités différentes (IRM et TDM), ainsi que des points de repère anatomiques apposés dans certains volumes. Les images médicales TDM sont découpées en deux ensembles (ou groupes). Le premier groupe, nommé *Gold*, contient 20 volumes TDM chacun d'eux accompagnés d'environ 40 points de repère anatomiques (voir Figure 2.4). Le second ensemble est nommé *Silver* et contient 60 images TDM 3D, mais sans points de repère anatomiques. Les volumes de ces deux groupes imagent des zones similaires du corps humain, l'abdomen et le thorax.

Les volumes sont stockés au format Nifti (angl. «*Neuroimaging Informatics Technology Initiative*»: initiative technologique en informatique de la neuro-imagerie) plutôt que DICOM (angl. «*Digital imaging and communications in medicine*»: imagerie numérique et communications en médecine). Nifti est un format de fichier simple qui stocke toutes les informations de l'image dans un seul fichier. En revanche, DICOM utilise un système de fichiers qui stocke les informations de l'image dans plusieurs fichiers.

Le format DICOM est conçu pour stocker des données de manière à ce qu'elles puissent être facilement partagées et analysées entre différents systèmes d'imagerie médicale. Cependant, Nifti est plus adapté aux analyses de recherche en neuro-imagerie et offre des fonctionnalités avancées pour le traitement et l'analyse de données.

Jeu de données provenant des hôpitaux de Saint-Étienne - Concernant la deuxième base de données, elle contient plusieurs milliers de volumes (environ 5000) acquis depuis différentes parties du corps humain. Les parties imagées peuvent être le crâne, le corps entier ou le tronc, l'abdomen (voir Figure 2.5 pour un exemple de ces volumes).

De nombreux volumes provenant de cette base de données ne peuvent pas être utilisés pour l'apprentissage. Cela est dû au fait que beaucoup d'entre eux ne contiennent qu'une ou quelques coupes d'un patient, ou bien que les volumes sont de tailles trop petites. Pour nous

1. voir visceral.eu

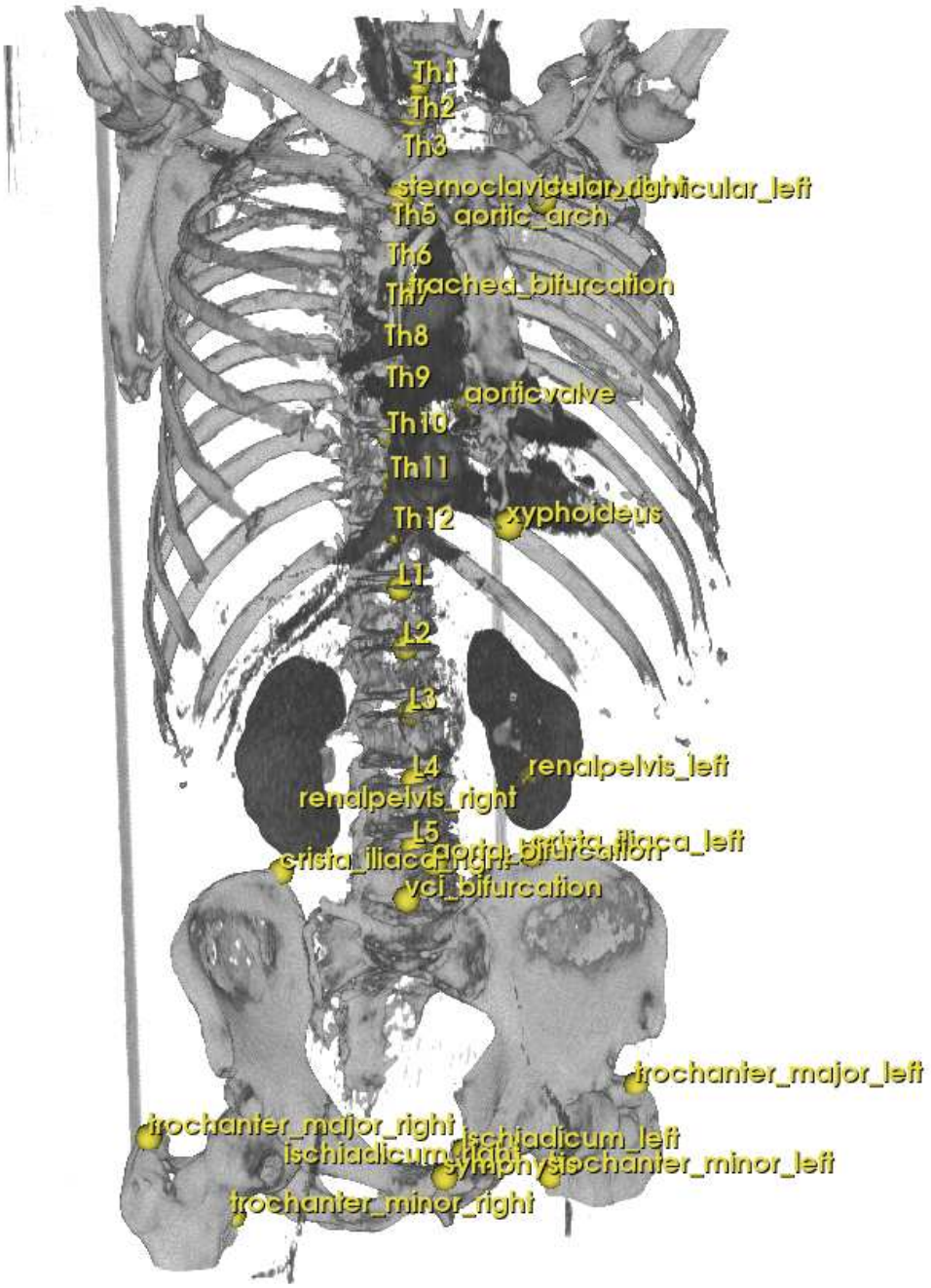


FIGURE 2.4 – Visualisation d'un volume provenant de la base *Gold* de VISCERAL, avec 36 points de repère anatomiques en jaune. Pour des raisons de lisibilité, seuls les os et certains organes ont été extraits. Nous effectuons cette extraction des os et organes à l'aide du rendu de volume («*volume rendering*» en anglais).

assurer que seuls les volumes de taille adéquate sont utilisés pour entraîner un réseau de neurones 3D, nous ne retenons que les volumes dont les dimensions dans les axes X, Y, Z sont supérieurs à $96 \times 96 \times 96$ voxels. Les volumes qui ont une dimension inférieure ne sont pas suffisamment grands pour permettre un apprentissage optimal du réseau de neurones.

FIGURE 2.5 – Visualisation d'un échantillon de volumes provenant de la base de données de Saint-Étienne. Pour des raisons de lisibilité, seuls les os et certains organes ont été extraits.

Données synthétiques - D'une manière similaire à superpoint, nous avons également créé un ensemble de données composé de formes simples. Ce nouveau jeu de données est formé de plusieurs formes simples 3D, telles que : des cubes, des droites, des sphères, des ellipsoïdes et des images comprenant du bruit gaussien. La Figure 2.6 montre un exemple de ces images synthétiques générées en 3D.

Pour sélectionner les formes 3D, nous avons analysé les images TDM à notre disposition. Ainsi, nous avons décidé de représenter les vaisseaux sanguins à l'aide de cylindres creux ou non (lignes), les organes tels que les poumons ou les reins par des ellipsoïdes, et les éléments similaires tels que les côtes ou les vertèbres par l'assemblage de formes similaires. Afin de générer un nombre suffisant de données, nous créons aléatoirement une forme et lui appliquons une transformation affine aléatoire.

2.2 Augmentation de données

Une première contribution de cette thèse provient de la façon dont nous obtenons des points clés répétables en vue des différents apprentissages. Nous allons détailler ici la méthode que nous mettons en place.

Pour générer de nouveaux volumes, nous utilisons une méthode qui consiste à estimer la densité de probabilité des transformations entre chacun des volumes, et à en tirer une transformation affine aléatoire. Cette transformation est ensuite utilisée pour déformer les volumes TDM et établir des correspondances entre les points clés.

Nous commençons par calculer les transformations affines locales entre les volumes en utilisant la méthode des moindres carrés. Pour chaque point de repère anatomique dans chaque volume de l'ensemble *Gold*, nous estimons une transformation locale en utilisant ce point de repère, ainsi que les trois points de repère les plus proches, et les quatre points de repère correspondants dans un autre volume.

Si le repère et ses trois voisins sont presque alignés sur une même ligne (comme dans le cas des repères vertébraux), alors le problème des moindres carrés devient mal conditionné, ce qui signifie que les résultats obtenus peuvent être peu fiables. Dans ces situations, nous décidons de rejeter la transformation estimée.

Nous pouvons obtenir au plus $Lk(k-1)/2$ matrices de transformation affine de taille 4×4, où L est le nombre de repères anatomiques (40 dans notre cas) et k est le nombre de

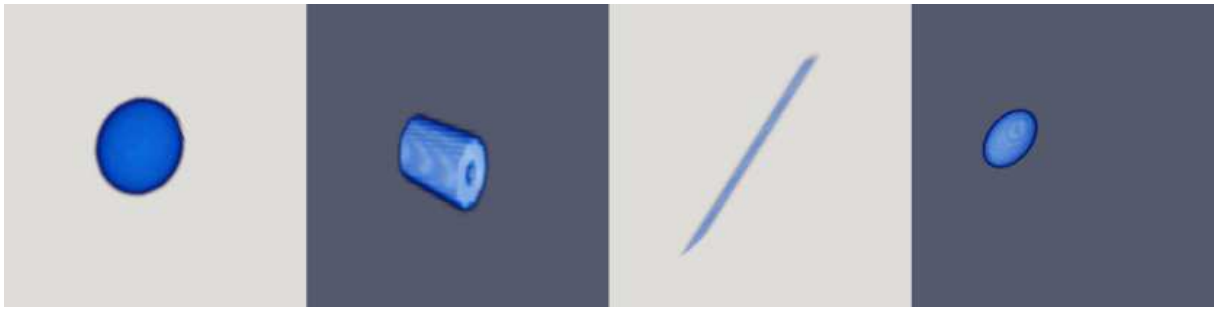


FIGURE 2.6 – Exemple d'images synthétiques générées automatiques, présentant diverses formes simples telles que : des droites, des cubes, des cylindres et des sphères, ainsi que des images présentant un bruit gaussien sans forme particulière.

volumes contenant ces repères (20 dans notre cas). Nous utilisons un test de Pearson (voir Figure 2.7) pour déterminer si les éléments de ces matrices sont indépendants, ce qui nous permet d'échantillonner chaque élément de manière indépendante. Nous appliquons ensuite la méthode d'estimation par noyau à ces matrices pour évaluer la densité des transformations intervolumes. Nous utilisons le test t de Student pour déterminer que le noyau gaussien est un bon ajustement pour cette estimation. Enfin, nous utilisons les règles de Scott (voir [Scott, 2015]) pour mesurer la largeur de bande du noyau.

Afin de produire notre ensemble de données semi-synthétiques, nous utilisons la densité de probabilité pour échantillonner les transformations que nous appliquons aux volumes du sous-ensemble *Silver*. Nous utilisons le terme *semi-synthétique* pour désigner la génération d'une nouvelle image synthétique à partir d'une image naturelle, c'est-à-dire une image acquise par des méthodes telles que les scanners ou IRM. En d'autres termes, nous obtenons le volume transformé V_ti en appliquant une transformation échantillonnée t à un volume *Silver* V_i. Nous détectons ensuite les points clés dans les volumes d'origines et transformés V_i et V_ti en utilisant le détecteur et descripteur SURF3D, ce qui nous permet d'obtenir deux ensembles de points clés P_i et P_ti. Pour déterminer si un point clé se trouve à l'intérieur ou à l'extérieur du corps du patient, nous créons un masque pour chaque volume en utilisant une méthode de seuillage sur l'image CT. Les points clés situés à l'extérieur du corps du patient sont alors supprimés.

Ensuite, nous appliquons la transformée inverse t^{-1} aux points clés de V_t^i qui ont été filtrés par le masque, de manière à les ramener dans leur emplacement d'origine. Puis, nous utilisons un arbre k-d pour trouver les correspondances de points clés entre les volumes d'origine et transformés, en sélectionnant les paires de points $(p \in P_i, q \in t^{-1}(P_i^t))$ dont la distance entre p et q est inférieure à 4 mm. Nous avons choisi ce seuil de distance pour garantir un grand nombre de correspondances correctes. Grâce à cette méthode combinée avec la détection de points clés à l'aide de SURF3D, nous parvenons à identifier environ 20 000 points clés répétés dans chaque volume transformé.

Nous utiliserons la même méthode de transformation des volumes durant les apprentissages ne nécessitant pas de points clés et donc non supervisés.

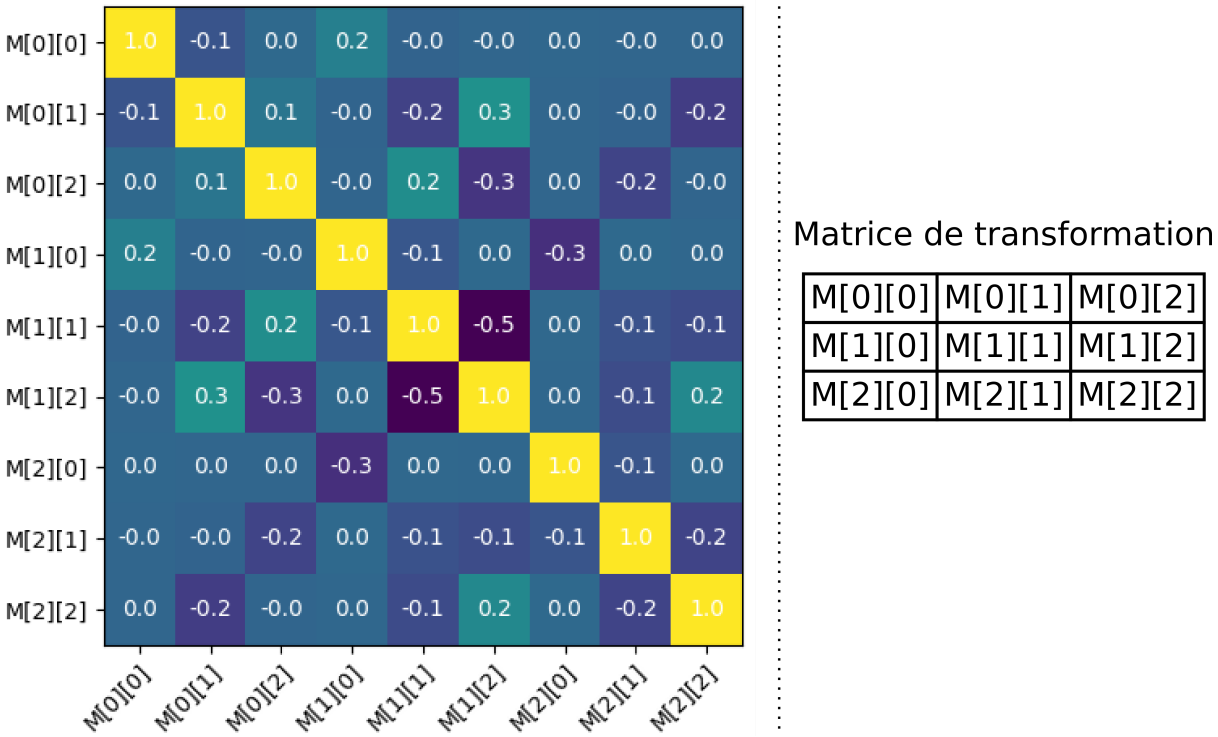


FIGURE 2.7 – Matrice de corrélation du test de Pearson, sur les éléments des matrices de transformations estimées grâce aux points de repère anatomiques. Chaque ligne et colonne de cette matrice représente les éléments des matrices de transformations affines qui sont mis en correspondance grâce au test de Pearson. Par exemple, la ligne notée $M[0][0]$ représente le premier élément des matrices affines comparé à tous les autres éléments à l'aide du test de Pearson, un exemple de notation de ces matrices affines est visible sur la figure de droite.

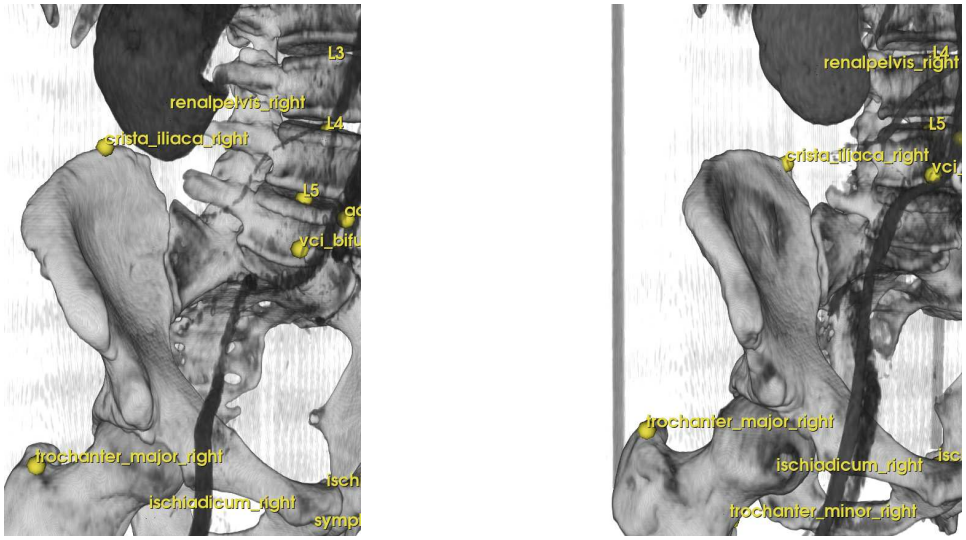
2.3 Analyse des points de repères anatomiques et replacement

Nous allons maintenant détailler, analyser et discuter des points de repère anatomiques (ou marqueurs) apposés sur les volumes du groupe *Gold*, voir [Krenn et al., 2017].

2.3.1 Analyse des points de repères anatomiques

Une méthode couramment utilisée pour l'analyse des extracteurs de points clés consiste à recaler les volumes des patients en utilisant les points clés détectés. Nous effectuons ce processus de recalage dans un espace commun à l'aide de l'outil FROG, comme expliqué dans la Section 1.2.4. Une fois que le recalage est effectué, il est possible de transformer les points de repère anatomiques dans le même espace pour analyser la distance moyenne entre chaque point. Une petite distance moyenne entre ces marqueurs anatomiques indique une bonne qualité du recalage.

Le positionnement des points de repère est une étape minutieuse qui demande du temps aux praticiens hospitaliers, et présente des défis complexes. Cette complexité réside dans le fait que le positionnement des points de repère pour identifier une partie spécifique du corps



(a) Volume 100 **(b)** Volume 110

FIGURE 2.8 – Les Figures 2.8(a) et 2.8(b) présentent des points de repère anatomiques placés par des experts, chez deux patients différents. Dans un souci de clarté visuelle, seuls les os et quelques organes ont été sélectionnés et extraits dans cette figure.

humain peut varier considérablement d'un praticien à l'autre. Par exemple, pour identifier le nez, un point de repère pourrait être placé sur la pointe nasale ou au centre du nez, il représenterait toujours la même partie du corps. Cette variabilité de placement entre les points peut être observée dans la Figure 2.8, où deux points censés être identiques sont placés différemment (par exemple les points notés «*crista_iliaca_right*», ou crête iliaque droite en français). Toutefois, pour minimiser la distance entre les points de repère anatomiques après recalage, il est essentiel d'avoir des points placés de manière précise et similaire dans chaque image.

Un autre point intéressant à considérer lors de la création de ces repères est leur emplacement dans le corps humain. Pour analyser l'alignement de manière globale, en prenant en compte à la fois les tissus denses tels que les os, et les tissus mous comme les organes, il est important de répartir les points de repère dans toutes les zones du corps de façon homogène.

La base de données VISCERAL contient un peu plus de 40 points de repère anatomiques dans les volumes du groupe *Gold*. Parmi eux, seuls 36 points sont placés de manière similaire dans tous les volumes. Parmi ces 36 marqueurs, 75% sont situés dans les tissus denses tandis que les 25% restants sont situés dans les tissus mous. Cette répartition permet une bonne analyse du recalage dans les os, mais est peu efficace dans les organes.

C'est pour ces différentes raisons que nous avons mis en place une méthodologie afin de replacer les points de repère anatomiques. Nous allons maintenant analyser ce replacement des marqueurs.

2.3.2 Placement alternatif des points de repères anatomiques

Nous avons défini un placement alternatif rapide et précis pour replacer chaque point et en ajouter de nouveaux qui permettraient d'analyser plus en profondeur les tissus mous, voir

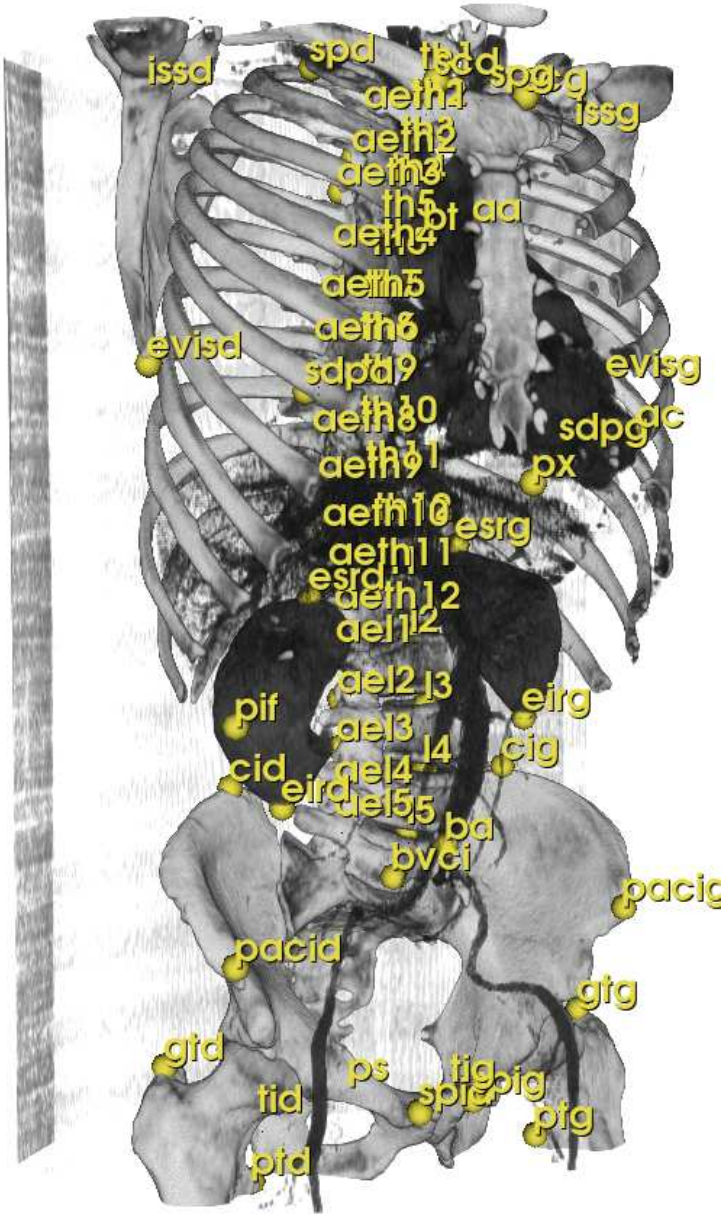


FIGURE 2.9 – Visualisation d'un volume provenant de la base *Gold* de VISCERAL, avec 68 points de repère anatomiques provenant de notre placement alternatif, en jaune. Pour des raisons de lisibilité, seuls les os et certains organes ont été extraits.

la Figure 2.9. Nous avons également redéfini la nomenclature, voir le Tableau 2.1.

Nous avons mis au point une méthode rapide et précise de placement alternatif pour remplacer chaque point existant et ajouter de nouveaux points, qui permettent d'analyser plus en détail les tissus mous, comme indiqué dans le Tableau 2.1. En outre, nous avons revu la nomenclature associée. Le placement des points a été réalisé à l'aide du logiciel *3DSlicer*.

Le Tableau 2.1 présente les distances entre les points de repère anatomiques ini-

Zone anatomique	Points de repères anatomiques (notation)		Distances moyennes	
	VISCERAL	Placement alternatif	VISCERAL	Placement alternatif
Vertèbres thoraciques (1 à 12)	Th[1-12]	th[1-12]	9.75	**8.47**
Vertèbres lombaires (1 à 5)	L[1-5]	l[1-5]	12.74	**9.94**
Apophyses épineuses lombaires (1 à 5)	✗	ael[1-5]	✗	**9.17**
Apophyses épineuses thoraciques (1 à 12)	✗	aeth[1-12]	✗	**10.67**
Petits trochanters (g et d)	trochanter_minor[g-d]	pt[g-d]	**4.23**	5.32
Tubérosités ischiatiques (g et d)	ischiadicum[g-d]	ti[g-d]	5.02	**4.14**
Pointe du sacrum	✗	ps	✗	**6.04**
Symphyses pubiennes iliaques (g et d)	symphysis	spi[g-d]	10.56	**3.58**
Grands trochanters (g et d)	trochanter_major[g-d]	gt[g-d]	14.85	**5.05**
Pointes antérieurs des crêtes iliaques (g et d)	✗	paci[g-d]	✗	**6.43**
Bifucation vci	vci_bifurcation	bvci	8.60	**7.73**
Bifurcation aortique	aorta_bifurcation	ba	**7.29**	9.22
Crêtes iliaques (g et d)	crista_iliaca_[g-d]	ci[g-d]	9.73	**7.19**
Pointe inférieure du foie	✗	pif	✗	**27.68**
Extrémités inférieurs des reins (g et d)	✗	eir[g-d]	✗	**13.20**
Extrémités supérieurs des reins (g et d)	✗	esr[g-d]	✗	**13.67**
Sommets pulmonaires des diaphragmes (g et d)	✗	sdp[g-d]	✗	**9.32**
Apex cardiaque	✗	ac	✗	**13.35**
Extrémités inférieurs des scapulas (g et d)	✗	evis[g-d]	✗	**10.55**
Processus xyphoïde	xyphoideus	px	**12.43**	13.12
Bifucation tracheal	trachea_bifurcation	bt	3.99	**3.67**
Arc aortique	aortic_arch	aa	8.66	**6.37**
Sternoclavicules (g et d)	sternoclavicular_[g-d]	sc[g-d]	**4.84**	5.00
Sommets des poumons (g et d)	✗	sp[g-d]	✗	**5.66**
Incisures supras scapulaires (g et d)	✗	iss[g-d]	✗	**2.34**
Pelvis rénal	renalpelvis_[g-d]	✗	**8.60**	✗
Valve Aortique	aorticvalve	✗	**8.54**	✗

TABLEAU 2.1 – Tableau résumant les points de repères anatomiques provenant de la base de données VISCERAL et également ceux replacés. Dans le tableau, (g et d) signifie gauche et droit.

tialement fournis par la base VISCERAL et les nouveaux points que nous avons ajoutés et replacés, après avoir recalé les volumes dans un espace commun. Les résultats indiquent que la plupart des distances moyennes entre ces nouveaux points sont meilleures que celles avec les points de VISCERAL.

La nouvelle base de points de repère anatomique que nous avons établie comprend désormais 68 marqueurs, dont 14 se trouvent dans les tissus mous, par rapport aux 5 présents dans la base VISCERAL. L'incorporation de ces nouveaux repères permet une analyse plus précise du recalage dans les tissus mous. Par exemple, nous sommes en mesure de quantifier la distance entre les points de repère anatomiques situés à la pointe inférieure du foie, après

recalage. La distance mesurée est de $27,68$ mm pour SURF3D. Nous utiliserons la nouvelle base de 68 points de repère anatomique pour effectuer nos évaluations par la suite.

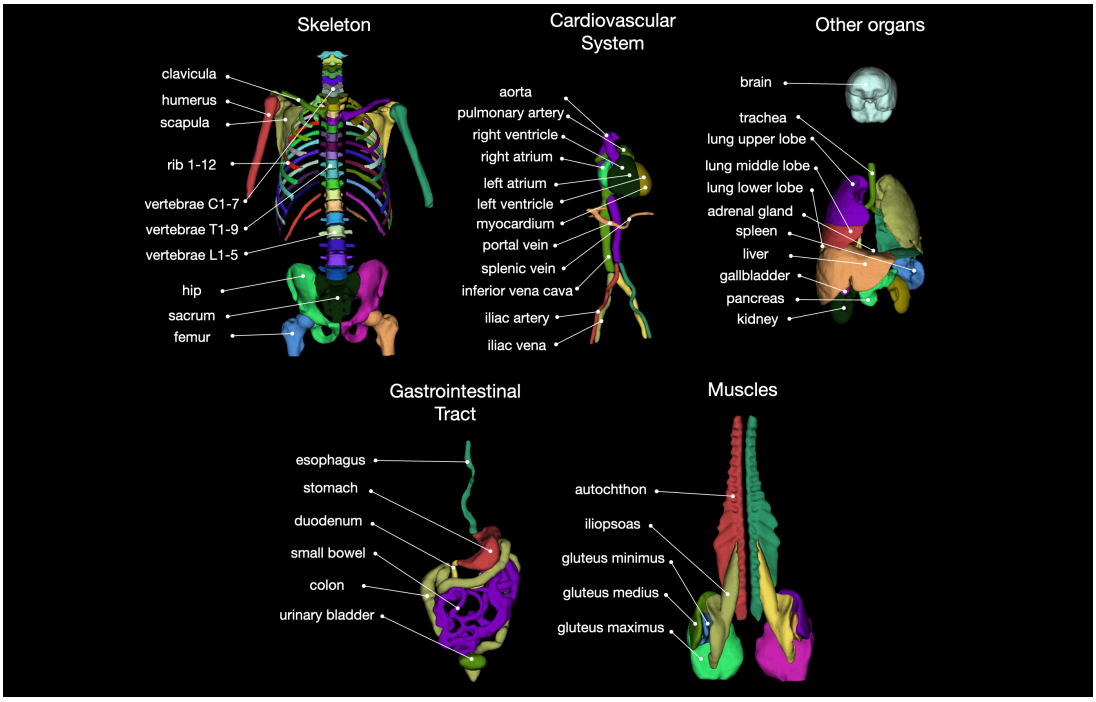


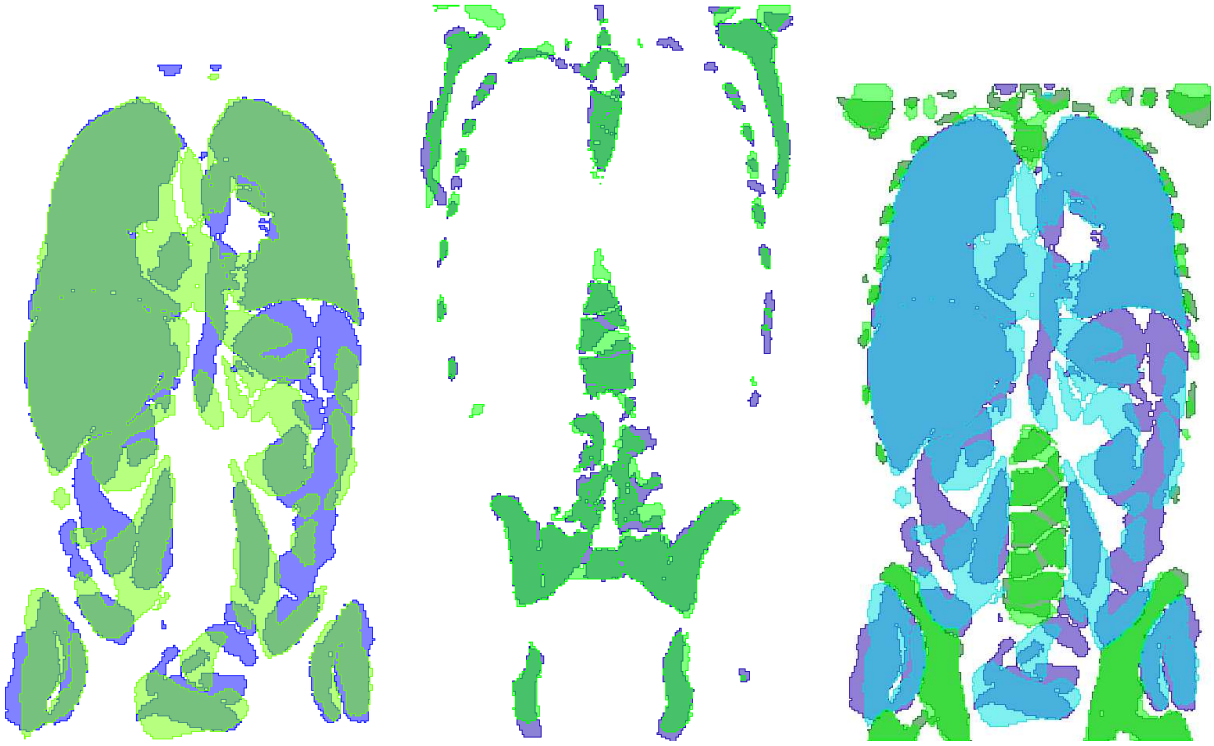
FIGURE 2.10 – Aperçu des 104 structures anatomiques pouvant être segmentées par TotalSegmentator. **Source:** Wasserthal et al. (2022)

La métrique principale que nous avons utilisée, qui permet de calculer les erreurs de recalage et qui s'obtient en calculant la moyenne de toutes les distances entre les marqueurs anatomiques, révèle une distance moyenne de 8.65 mm pour les marqueurs de la base VISCERAL, contre 8.67 mm pour les nouveaux points que nous avons ajoutés. Cette légère augmentation de distance s'explique facilement par l'ajout des nouveaux points dans les tissus mous, qui sont clairement des zones où le recalage est plus difficile à effectuer. Toutefois, en excluant les nouveaux points de repère dans les tissus mous, la distance moyenne totale tombe à 6.83 mm.

Nous avons constaté que les régions contenant des organes présentent un recalage moins précis que les structures solides telles que les os. Pour analyser ce recalage imparfait, nous pouvons examiner le pourcentage de points clés détectés en relation avec les zones molles et denses. D'après la référence [Gallagher et al., 2013], le corps humain est constitué à 20% d'os, et approximativement le double en termes de masse musculaire et d'organes. En nous basant sur ces données, nous devrions donc identifier deux fois plus de points clés dans les tissus mous par rapport aux tissus denses.

Nous pouvons segmenter nos volumes de patients en utilisant la méthode « *TotalSegmentator* » [2] décrite dans la référence [Wasserthal et al., 2022]. Cette technique de segmentation permet de distinguer 104 structures anatomiques, dont 27 organes, 59 os, 10 muscles et 8

2. https://github.com/wasserth/TotalSegmentator



(a) Segmentations représentant les tissus mous.

(b) Segmentations représentant les tissus denses.

(c) Segmentations représentant les tissus denses et mous.

FIGURE 2.11 – Superposition de deux segmentations après calcul des transformées vers un espace commun, des volumes de deux patients. Le calcul des transformées a été effectué par FROG.

vaisseaux différents (voir Figure 2.10). La majorité de ces structures appartiennent au thorax ou à l'abdomen, ce qui correspond à nos données.

En combinant les muscles et les os entre eux dans les segmentations, nous pouvons estimer que 17% des zones segmentées sont constituées de muscles, d'organes et 3% de tissus denses. En examinant le nombre de points clés détectés dans chacune de ces zones, nous trouvons approximativement 20% de points détectés dans les zones denses, 26% dans les muscles et les organes, et le reste dans des zones non segmentées, telles que les zones graisseuses, la peau ou même à l'extérieur du patient.

Ces segmentations peuvent également être utilisées pour évaluer le recalage dans les différents tissus, comme le montre la Figure 2.11. Visuellement, il est apparent que le recalage dans les tissus denses semble de meilleure qualité en comparaison avec le recalage dans les tissus mous.

3

Entraînement du descripteur

Sommaire

Comme nous l'avons vu précédemment, les points clés (ou points d'intérêt) et leurs descripteurs sont des éléments importants pour l'analyse et la compréhension des images, dans les domaines de la vision par ordinateur et de la reconnaissance de formes. Ces points caractéristiques peuvent être utilisés pour diverses tâches telles que le recalage d'images, la reconnaissance d'objets, la reconstruction 3D, la segmentation d'images ou encore la mise en correspondance d'images.

Lorsqu'il s'agit de mettre en correspondance les images, les points clés doivent être décrits afin d'être associés. Dans cette étape, la description revêt une importance particulière. Elle permet de représenter chaque point clé sous forme d'un vecteur de caractéristiques qui capture les informations clés de l'environnement environnant du point. Cette représentation doit être invariante par rapport aux transformations géométriques et photométriques, tout en étant capable de distinguer les points clés qui ne sont pas similaires.

L'apprentissage de descripteurs de points clés est un domaine amplement étudié en 2D, mais assez peu en 3D. Nous explorerons deux approches d'apprentissage, la première par réseau siamois utilisant des triplets ([Schroff et al., 2015 ; Vijay Kumar B et al., 2015]) et la seconde utilisant des listes (angl. «*listwise*», [Cakir et al., 2019 ; He et al., 2018]) pour apprendre des représentations de points clés discriminantes. Nous comparerons ensuite ces descripteurs appris aux descripteurs de 3D-SURF.

3.1 Apprentissage de descripteur de point clé

Comme nous l'avons vu dans l'état de l'art (voir Chapitre 1), il existe plusieurs méthodes d'apprentissage des descripteurs de points clés. Nous nous intéresserons tout particulièrement à deux d'entre elles : les méthodes siamoises par triplets ou duets et les méthodes par listes, voir [Masone et Caputo, 2021]. Dans la suite de ce chapitre, nous nommerons indifféremment le voisinage direct d'un point clé, parcelle, fragment, sous-bloc ou encore sous-volume (en 3D).

3.1.1 Approche par triplets

La première approche consiste à former des triplets de parcelles d'images, puis pour chacune d'entre elles, un vecteur de description sera calculé. Ces derniers sont utilisés pour le calcul de la fonction de perte permettant de mesurer la distance entre les trois vecteurs. L'apprentissage consiste à augmenter la distance pour deux éléments qui ne correspondent pas

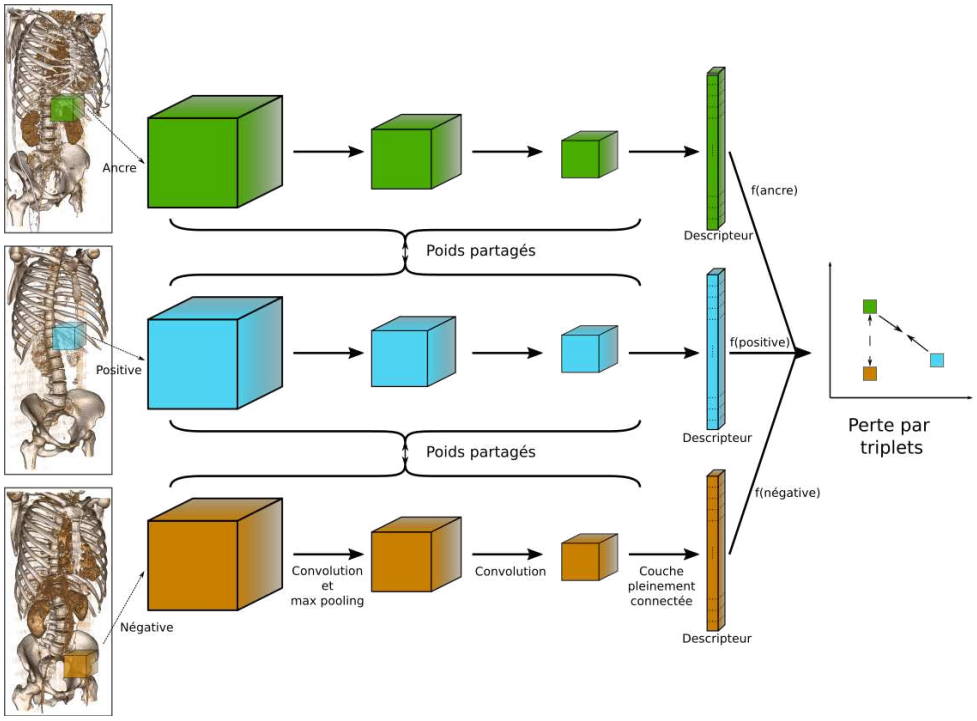


FIGURE 3.1 – Schéma d'apprentissage avec une architecture par triplets. Un triplet de sous-volumes a, p, n est envoyé au réseau et les descripteurs résultants servent au calcul de la fonction de perte par triplets. Les sous-blocs de volumes en entrées peuvent varier en taille, de 10^3 à 25^3 voxels.

(la parcelle nommée ancre (a) et celle nommée négative (n)) et diminuer la distance entre les éléments similaires (la parcelle nommée ancre (a) et celle nommée positive (p)), voir Figure 3.1.

Minage des triplets - La méthode par triplets nécessite un minage (ou recherche) de ces triplets, en vue d'assurer un bon apprentissage en ne présentant que des éléments cohérents. Différents procédés existent pour extraire les meilleurs triplets, tels que : le «*minage dur*» qui consiste à former des triplets au sein d'un mini-lot de sous-images ou l'échange d'ancre. Le Chapitre 1.5.1 décrit en détail le procédé le plus efficace qui consiste à éliminer les triplets faciles et à ne conserver que les triplets difficiles et semi-difficiles le cas échéant. Le second procédé, nommé «*échange d'ancre*», permet d'économiser un temps important de calcul. Selon [Balntas et al., 2016], un réseau utilisant la méthode de «*minage dur*» des triplets utilise 67% de son temps d'entraînement à former les meilleurs triplets.

Le procédé d'échange d'ancre (voir Figure 3.2) consiste à interchanger la parcelle ancre et la positive, si $\delta_* = \delta'$, avec $\delta_* = \min(\delta, \delta')$, $\delta' = ||f(p) - f(n)||_2$ et $\delta = ||f(a) - f(n)||_2$. En d'autres termes, si la distance entre les descripteurs n et p est plus petite que la distance entre les descripteurs a et n, alors p devient l'ancre et a devient la positive, permettant ainsi d'obtenir systématiquement un triplet difficile pour l'étape de rétropropagation du gradient.

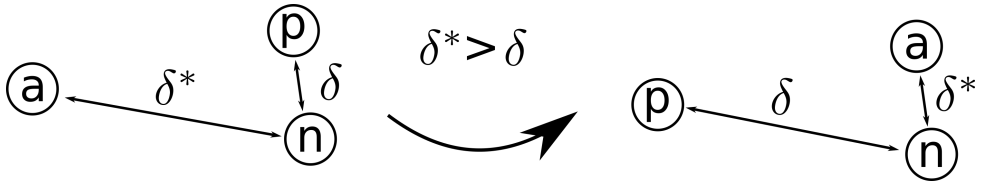


FIGURE 3.2 – Schéma montrant le procédé d'échange d'ancre, en vue d'obtenir toujours un triplet difficile. Si la distance δ^* est supérieure à la distance δ alors on va opérer un changement entre la parcelle a (ancre) et la parcelle p (positive), pour obtenir un triplet difficile.

3.1.2 Approche par liste

Les apprentissages siamois par duets et triplets sont les méthodes les plus populaires et les plus utilisées pour l'apprentissage de représentations d'images ou de calcul de descripteurs, mais elles souffrent de deux limitations. La première est due à la méthode de minage des triplets, qui entraîne un surcoût mémoriel et calculatoire significatif à l'apprentissage et peut entraîner de mauvais résultats si cette étape n'est pas réalisée correctement. Ce surcoût de calcul est d'ailleurs une notion très importante en 3D, car les sous-blocs de volumes que nous allons donner au réseau seront forcément beaucoup plus importants en dimensions (i.e nombre de voxels). En 2D, une image de taille 32^2 (soit 1024 pixels) sera extraite autour d'un point que l'on aimerait décrire ; si l'on étend cette parcelle à la 3D, sa taille sera de 32^3 (soit 32768 voxels), ce qui augmente considérablement le temps nécessaire à l'apprentissage du même réseau étendu à la 3D. La seconde limitation est théorique, il a été démontré dans [Liu, 2009] que ces approches ne permettent pas de réellement optimiser la moyenne des précisions sur chaque élément (angl. «*mean average precision*», mAP). C'est pourquoi la seconde approche vise à optimiser directement cette mAP.

Pour évaluer les performances de la correspondance du plus proche voisin (dans notre cas, du descripteur d'image le plus proche), il est possible d'adopter la précision moyenne («*Average Precision*» en anglais), une mesure couramment utilisée. L'AP évalue les performances d'un système de recherche dans le cadre de la pertinence binaire : les résultats de la recherche sont soit «*pertinents*» soit «*non pertinents*» selon une requête donnée. Cela correspond naturellement à la configuration de correspondance de descripteurs locaux, où, étant donné un descripteur de référence (requête), les descripteurs dans une image cible sont soit une correspondance correcte, soit une correspondance incorrecte. La mAP correspond simplement à la moyenne des AP et se calcule de la façon suivante :

$$AP(q) = \frac{\sum_{k=1}^{m} P@k(q) \cdot l_k}{\#\{\text{descripteurs pertinents}\}} \tag{3.1}$$

Avec m le nombre total de descripteurs associés à une requête q, l_k la vérité terrain binaire sur la pertinence du descripteur à la k-ième position et :

$$P@k(q) = \frac{\#\{\text{descripteurs pertinents dans les k premières positions}\}}{k} \tag{3.2}$$

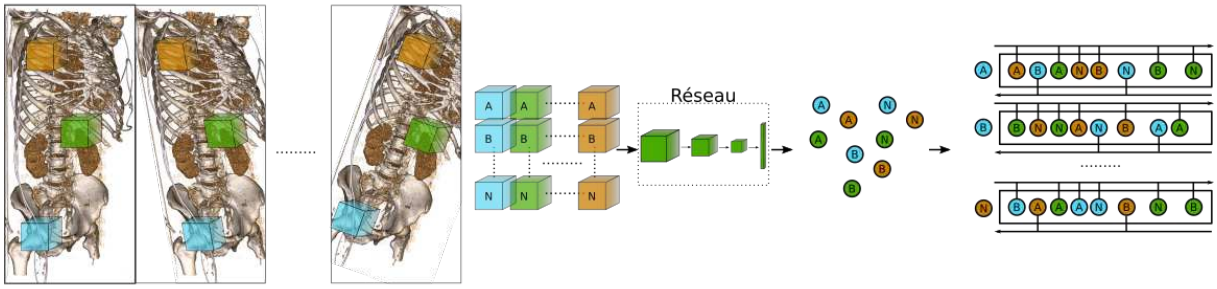


FIGURE 3.3 – Schéma d'apprentissage d'un réseau utilisant des listes et optimisant la mAP. Cette fonction de perte permet d'optimiser directement la précision moyenne, voir Équations 3.1 et 3.2. Dans cette figure, on peut voir à gauche les volumes desquels sont extraits les sous-volumes nécessaires à la formation des listes de sous-volumes. Les descripteurs correspondants à chacune des parcelles sont ensuite calculés par le réseau, puis utilisés pour calculer la mAP. Les parcelles de volumes en entrée peuvent varier en taille, de 10^3 à 25^3 voxels. Sur le côté droit, nous avons des représentations de listes de descripteurs pour lesquelles nous souhaitons améliorer l'appariement. Plusieurs listes de descripteurs sont présentes, où chaque descripteur est symbolisé par une lettre dans un cercle coloré et où chaque descripteur correspond à une même localisation dans deux patients différents sont de même couleur. Dans la deuxième liste de descripteurs (la seconde ligne), on observe que la requête du descripteur B en bleu n'est suivie d'aucun autre élément de la même couleur bleue en tête de liste. Cela signifie qu'aucun descripteur correspondant à la même localisation n'est suffisamment proche dans l'espace des descriptions. Par conséquent, notre objectif est de rapprocher en tête de liste les éléments A et N, qui sont en bleu.

Ici $P@k(q)$ permet de définir la précision à la position k pour une seule requête. Ces équations permettent d'obtenir le taux de descripteurs similaire (pertinents) au sein des k premiers descripteurs. Un descripteur *pertinent* est un descripteur devant être associé au descripteur requête, c'est-à-dire que les deux descripteurs décrivent la même image, ou même parcelle des images. Dans ces deux équations, $\#\{.\}$ renvoie la cardinalité d'un ensemble.

Cette seconde approche utilise une formulation par liste (voir Figure 3.3) permettant un calcul différentiable afin d'utiliser la rétropropagation, voir [Cakir et al., 2019 ; He et al., 2018 ; Revaud, Almazan et al., 2019]. Cette formulation de la mAP nécessite des mini-lots (angl. «*mini batches*») de grandes tailles, ce qui pose problème dans l'apprentissage 3D qui demande une quantité de mémoire plus importante.

Pour notre étude des descripteurs et en vue d'obtenir un descripteur efficace et cohérent pour notre application finale, c'est-à-dire une description précise, rapide et permettant une mise en correspondance juste des points clés, nous avons exploré les deux approches : par triplets et par listes.

Nous avons tout d'abord mis en œuvre les deux méthodes en 2D, afin d'analyser quels hyperparamètres sont les plus importants à faire varier comme : le taux d'apprentissage (angl. «*learning rate*»), la taille des mini-lots et du lot global, la taille des descripteurs, la taille des parcelles à extraire et à décrire autour d'un point clé, etc. Cette première exploration des hyperparamètres en 2D permet une recherche beaucoup plus rapide, en moyenne 4 à 5 heures pour un entraînement en 300 époques et 500k parcelles d'images.

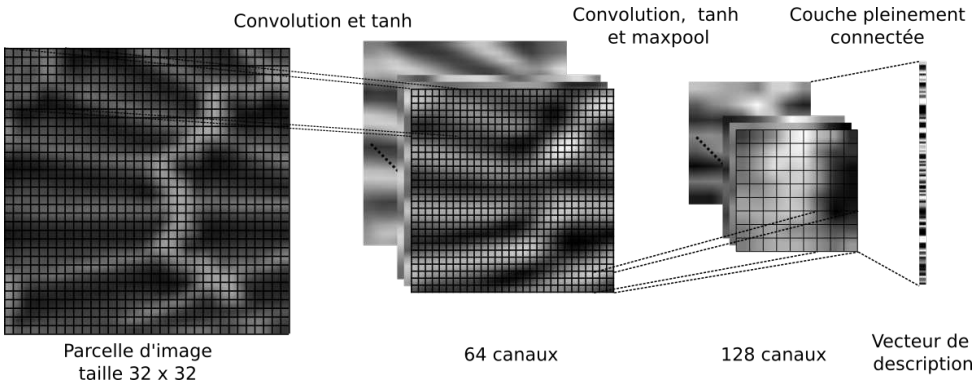


FIGURE 3.4 – Architecture du réseau utilisé lors de l'apprentissage du descripteur par triplets et par listes. Ce réseau est composé de plusieurs phases de convolution et maxpooling, et finalement d'une couche pleinement connectée permettant d'obtenir notre vecteur de description.

Après avoir effectué cette première étape en 2D, nous avons étendu la méthode à la 3D, sur notre jeu de données médicales.

3.2 Réseau, base de données et méthodes

Dans cette partie, nous allons analyser le réseau de neurones ainsi que les données que nous avons utilisées lors de l'apprentissage, de l'évaluation et de la phase de test.

3.2.1 Réseau de neurones

Il convient en premier lieu de choisir une architecture de réseau de neurones capable de supporter une détection précise et discriminante des points clés. Deux contraintes principales se posent alors : la taille des images à traiter en entrée, ainsi que la taille de l'architecture du réseau, afin de garantir la faisabilité de l'entraînement. Plusieurs choix de réseau s'offrent à nous à ce moment, un réseau profond tel qu'un VGG (voir [Simonyan et Zisserman, 2014]) ou bien un réseau peu profond avec quelques couches de convolutions (en général moins de 3 couches), voir [Zagoruyko et Komodakis, 2017].

Dans notre approche, nous nous sommes tournés vers un réseau peu profond, mais plus large afin de mémoriser les différentes descriptions des parcelles d'images fournies en entrée. Ce réseau est composé d'une normalisation des parcelles d'entrée suivie de deux couches de convolutions chacune accompagnée de la fonction d'activation tangente hyperbolique (Tanh) et finalement une couche pleinement connectée permettant de calculer le vecteur de description final, voir Figure 3.4.

Ce réseau contient environ 1.5M paramètres ajustables lors de l'entraînement en 3D, ces paramètres correspondent aux poids des connexions et sont les éléments appris lors de l'entraînement. Un tel réseau permet l'utilisation de processeur (angl. «*central processing unit*», CPU) dont nous disposons en grande quantité, plutôt que celle d'unité de traitement graphique (angl. «*Graphics Processing Unit*», GPU) qui sont des ressources très coûteuses. Le passage

d'un mini-lot de 1000 parcelles de taille 24^3 dans le réseau de neurones se fait en un temps quasiment similaire sur GPU ou CPU dans notre cas.

3.2.2 Base de données

Pour nos différents entraînements et tests, nous avons utilisé trois ensembles de données, deux pour la 2D et un pour la 3D.

Ensemble de données 2D - Pour estimer la taille de l'ensemble de données nécessaire à un entraînement en 3D, nous avons tout d'abord procédé à quelques entraînements en 2D, sur deux ensembles de données. Le premier ensemble, nommé « *Yosemite* » (voir [Winder et al., 2009]), est formé de plusieurs milliers d'images et dont les correspondances entre chaque image sont connues. Le second ensemble d'images 2D que nous avons utilisé provient de la base de données VISCERAL. Nous avons extrait des coupes d'images au sein des volumes dans l'axe coronal des patients, puis utilisé le détecteur SURF afin d'extraire des points clés.

Ensemble de données 3D - L'ensemble des patients que nous avons utilisés provient de la base de données VISCERAL et a permis de créer deux ensembles de données grâce à deux méthodes différentes.

Dans un premier temps, nous avons voulu créer un ensemble de données en utilisant la méthode FROG comme référence dans la mise en correspondance des points clés. Pour créer ce jeu de données, nous apparions chaque point clé détecté grâce à la méthode 3D-SURF. Après calcul de ces appariements et du recalage des différents volumes dans un espace commun, il nous est également possible de transformer chaque point détecté de chaque volume dans cette espace commun à l'aide de la méthode FROG. Puis si pour un point donné, il existe au moins 10 points distants d'un seuil inférieur à 8 mm, on considère que la moyenne de ces localisations correspond à un point clé répété à travers tous les volumes. Les points clés résultants de ce seuillage seront retransformés dans leur espace d'origine puis utilisés pour extraire les sous-volumes. Nous nommons cette première base d'apprentissage *Base_FROG*

Pour créer notre second jeu de données, nous avons extrait des sous-volumes autour de chaque point clé détecté selon la méthode expliquée dans la Section 2.2. Cet ensemble de données, durant l'entraînement, ne nous permet pas d'extraire les éléments au sein de différents patients. Cette contrainte vient du fait que la correspondance entre les points n'est connue qu'au sein d'un même patient, nous n'avons pas cette connaissance des correspondances dans le cas de l'interpatient. Nous avons créé ce second ensemble de données, car le premier ne permet pas une précision assez importante de la localisation des points clés, ce qui implique une extraction des parcelles autour des points clés peu fiable. Cette base de données est nommée *Base_synthétique*

Durant l'entraînement, la méthode par triplets permet d'utiliser tous les patients en même temps, tout en extrayant les triplets au sein d'un même patient. Cependant, il n'est pas possible d'utiliser des patients différents pour former nos triplets, car nous n'avons pas la connaissance des similarités (i.e des appariements) des points à travers plusieurs patients. C'est

pourquoi la méthode par liste nous oblige également à former nos mini-lots de sous-volumes obligatoirement au sein d'un seul patient.

Plusieurs options s'offrent à nous pour extraire les sous-volumes, afin de nourrir le réseau de neurones tout au long de l'entraînement :

- La première option consistant à sauvegarder en mémoire chaque volume, puis soit à les charger un par un durant l'entraînement, soit à tous les charger en mémoire et extraire des parcelles de volumes autour des points clés précédemment détectés.
- La seconde option revenant à extraire et à sauvegarder les sous-volumes extraits autour des points clés dans des fichiers distincts. Dans cette méthode, nous créons autant de fichiers contenant les régions entourant les points que de points clés détectés.

L'avantage de la première possibilité étant de pouvoir utiliser n'importe quelle taille de sous-volume durant l'entraînement et donc de pouvoir faire varier nos tests plus simplement. Malheureusement, cette option est extrêmement coûteuse en termes de chargement des volumes des patients, environ 10 secondes par volume et 10 minutes par époque pour la totalité des patients, étape qui doit être réexécutée à chaque époque. Ces tests de durées d'exécution ont été menés sur une plateforme Linux 64-bit disposant d'Intel Xeon 2.6 GHz. La seconde option permet de charger n'importe quel sous-volume en temps constant (environ quelques microsecondes), mais nous oblige à recalculer chaque fichier contenant les sous-volumes si nous souhaitons faire varier les tailles de volumes en entrée. Nous avons opté pour la seconde méthode qui permet de réduire la durée d'une époque d'entraînement de 1h30 à 15 minutes environ.

Durant l'entraînement en 3D, nous avons donc utilisé la base de données de VISCERAL, qui contient 60 volumes sans points de repère anatomiques (le groupe *Silver*) et 20 volumes contenant chacun 68 marqueurs anatomiques (le groupe *Gold*). Nous avons fractionné notre ensemble de données provenant du groupe *Silver* en deux sous-ensembles, le premier contenant 80% des volumes de patients (soit 50 volumes) pour l'entraînement du descripteur et 20% (soit 10 volumes) pour l'évaluation du réseau tout au long de l'apprentissage. Nous avons conservé le second groupe afin d'effectuer nos tests après l'entraînement.

3.3 Entraînement et résultats

Dans cette partie, nous allons discuter du choix des hyperparamètres et de l'optimisation du réseau au travers de ces différentes variables, puis nous détaillerons les résultats obtenus en 2D puis en 3D.

Voici une liste des hyperparamètres qui nous semblent les plus importants à analyser durant l'étude en 2D avant le passage à la 3D :

- Optimiseur : l'algorithme descente de gradient stochastique (angl. «*stochastic gradient descent*», SGD) ([Ruder, 2016]) ou ADAM, [Kingma et Ba, 2017].
- Taux d'apprentissage (angl. «*learning rate*»).

Hyperparamètres	Méthode de recherche	Intervalle à explorer
Optimiseur	Grille	[SGD, ADAM]
Largeur du réseau	Grille	$[(32, 64), (64, 128), (128, 256)]$
Normalisation	Grille	[Présence, Absence]
Taille du descripteur	Grille	$[24, 48, 64, 96, 128]$
Marge	Grille	$[0.1, 0.4, 0.8, 2.0, 4.0]$
Taille du lot	Grille	$[100k, 400k, 1000k]$
Taille du mini-lot	Grille	$[100, 300, 600]$
Taille des parcelles	Grille	$[24, 32, 64]$
Taux d'apprentissage	Aléatoire	$[10^{-5}, 0.9]$

TABLEAU 3.1 – Ce tableau met en lumière les différents hyper-paramètres et intervalles de recherche que nous allons étudier, ainsi que la méthode de recherche mise en place pour chacun d'entre eux. Nous noterons par la suite : TV pour la Taille du Vecteur de description, LR pour le taux d'apprentissage («*Learning Rate*» en anglais) et TP pour la Taille des Parcelles.

(a) Parcelles d'images autour des points clés

(b) Parcelles d'images décrites par SIFT

(c) Parcelles d'images décrites par le réseau de neurones

FIGURE 3.5 – Dans ces trois schémas, des parcelles d'images sont extraites autour de points clés. Dans 3.5(a), une réduction des dimensions est effectuée directement sur les images. Dans les deux autres schémas 3.5(b), 3.5(c), les images sont d'abord décrites par SIFT ou le réseau appris et ensuite une réduction de dimensionnalité est effectuée.

- Taille du lot (angl. «*batch size*»), c'est-à-dire le nombre total d'images utilisées lors d'une époque d'apprentissage.
- Taille du descripteur de sortie.
- Taille des images en entrée du réseau de neurones.
- Taille du mini-lot d'images, c'est-à-dire le nombre d'images utilisé chaque session d'apprentissage.
- Largeur du réseau de neurones.

Notre recherche d'hyperparamètres a été faite en 2D, en combinant deux méthodes de recherche, la première par grille et la seconde par recherche aléatoire, voir [Bergstra et Bengio, 2012] etFigure 1.21 (voir Section 1.4.1).

Nous allons maintenant détailler les entraînements et résultats obtenus selon les

différents hyperparamètres choisis. Durant l'entraînement nous utiliserons plusieurs métriques, afin d'estimer le bon apprentissage du réseau, qui seront : la FPR95 (voir Chapitre 1.2.2), le taux d'appariement des descripteurs (TAD) (voir Chapitre 1.2.3) et finalement les fonctions de pertes d'entraînement et d'évaluation. Nous utiliserons également une métrique visuelle en utilisant t-SNE, voir Chapitre 1.2.2. Concernant l'évaluation du réseau en 3D, nous utilisons la métrique de distance moyenne entre les points de repère anatomiques, après recalage dans un espace commun par l'algorithme FROG. Nous noterons DMMA, la métrique de distance moyenne entre les marqueurs (ou points de repère) anatomiques.

3.3.1 Entraînements et résultats en 2D

Suite à une série d'apprentissages, nous avons fixé empiriquement quelques hyperparamètres comme : l'optimiseur, la largeur du réseau de neurones et l'utilisation de couches de normalisations. Durant ces tests, il est apparu que l'utilisation de l'algorithme SGD donnait de meilleurs résultats (dans notre cas) en comparaison à l'algorithme d'optimisation ADAM. L'utilisation de normalisation après chaque couche de convolution tend à réduire l'efficacité du réseau de description. Finalement, nous fixons la largeur du réseau à 64 canaux pour la première convolution et 128 canaux en sortie de la seconde convolution.

La visualisation des fonctions de pertes de plusieurs entraînements dans lesquelles nous avons fait varier l'optimiseur, la largeur du réseau de neurones, l'ajout ou non de couches de normalisations et la perte par triplets ou liste semble indiquer que restreindre l'apprentissage du réseau à 100 époques est largement suffisant. Les apprentissages semblent trouver un état stable aux alentours de la 75e époque d'entraînement.

Le Tableau 3.2 résume les différents entraînements et résultats obtenus en 2D sur l'ensemble de données *Yosemite*. On peut observer que les hyperparamètres les plus impactant sont : le taux d'apprentissage, la taille du descripteur et la taille des mini-lots pour la perte par liste. Dans ce tableau, on peut remarquer que la FPR95 la plus faible est obtenue par l'algorithme par triplets et que la TAD la plus faible quand elle est obtenue par l'algorithme à base de liste. Ceci peut probablement s'expliquer par le fait que le RNC basé liste optimise directement la précision moyenne, qui est une fonction permettant d'optimiser directement le nombre de descripteurs similaires retrouvés dans les k premiers éléments comparés.

Pour obtenir un résultat plus visuel, nous avons extrait une parcelle d'image et toutes ses correspondances, ainsi que 100 autres parcelles d'images représentant d'autres éléments. Ensuite, en utilisant le descripteur précédemment entraîné, nous calculons le vecteur de description de chacune de ces parcelles et mesurons la distance entre la parcelle «*requête*» (le premier élément de chaque ligne dans la Figure 3.6). Nous affichons alors les neuf parcelles d'images dont la distance entre leurs descripteurs et celui de l'image requête est la plus faible. La figure Figure 3.6 illustre ce résultat visuellement.

Nous avons effectué un entraînement similaire sur notre ensemble de coupes 2D extraites à partir des images volumiques de la base VISCERAL. Cet entraînement a été effectué d'une manière similaire à celle que nous allons mettre en place en 3D. C'est-à-dire qu'après avoir extrait des parcelles d'images autour des points clés détectés par SURF dans les différentes

Type d'apprentissage	Lot	mini-lot	TV	LR	Marge	TP	FPR95	TAD
triplets	100k	300	128	0.1	2	32	0.062	0.765
	400k	-	-	-	-	-	0.063	0.759
	800k	-	-	-	-	-	0.069	0.747
	400k	-	-	-	-	-	0.100	0.703
	-	500	-	-	-	-	0.068	0.755
	-	100	128	-	-	-	**0.055**	0.770
	-	300	48	-	-	-	0.103	0.702
	-	-	64	-	-	-	0.083	0.722
	-	-	96	-	-	-	0.072	0.747
	-	-	256	-	-	-	0.056	**0.773**
	-	-	128	0.01	-	-	0.104	0.699
	-	-	-	0.001	-	-	0.214	0.616
	-	-	-	0.0001	-	-	0.292	0.568
	-	-	-	0.1	1	-	0.058	0.770
	-	-	-	-	4	-	0.096	0.719
	-	-	-	-	2	24	0.073	0.736
	-	-	-	-	-	64	0.066	0.757
listes	-	1024	-	-	-	32	0.100	0.750
	-	2048	-	-	-	-	0.080	0.818
	-	4096	-	-	-	-	0.123	0.785
	-	2048	48	0.1	-	-	0.101	0.773
	-	-	64	0.1	-	-	0.109	0.781
	-	-	96	0.1	-	-	0.096	0.796
	-	-	256	0.1	-	-	0.098	0.802
	-	-	128	0.01	-	-	0.233	0.732
	-	-	-	0.1	-	24	0.128	0.781
	-	-	-	-	-	64	**0.066**	**0.819**

TABLEAU 3.2 – Tableau résumant différents entraînements en 2D. Chaque colonne représente un hyperparamètre différent avec de gauche à droite, le «*Lot*» représente le nombre total d'images utilisées durant l'entraînement, le «*mini-lot*» étant le nombre d'images utilisées lors de l'entraînement. La taille du vecteur final est notée «*TV*», le taux d'apprentissage «*LR*» et la taille des parcelles d'images «*TP*». Les métriques dans les deux dernières colonnes sont la FPR95 et le taux d'appariement des descripteurs notée «*TAD*».

images, nous formons nos triplets en prenant deux points clés qui ne correspondent pas dans une même image (les parcelles ancre (a) et négative (n)), et par une transformation affine aléatoire de l'ancre, nous obtenons l'élément positif (p) de notre triplet. Nous obtenons des résultats similaires à l'entraînement sur les images provenant de la base de données Yosemite, ce qui nous a incités à étendre ce réseau et ces deux types d'apprentissages en 3D.

3.3.2 Entraînement et résultats en 3D

Suite à l'analyse des hyperparamètres en 2D, nous avons fixé certains hyperparamètres en 3D pour effectuer un premier apprentissage et obtenir une base de comparaison.

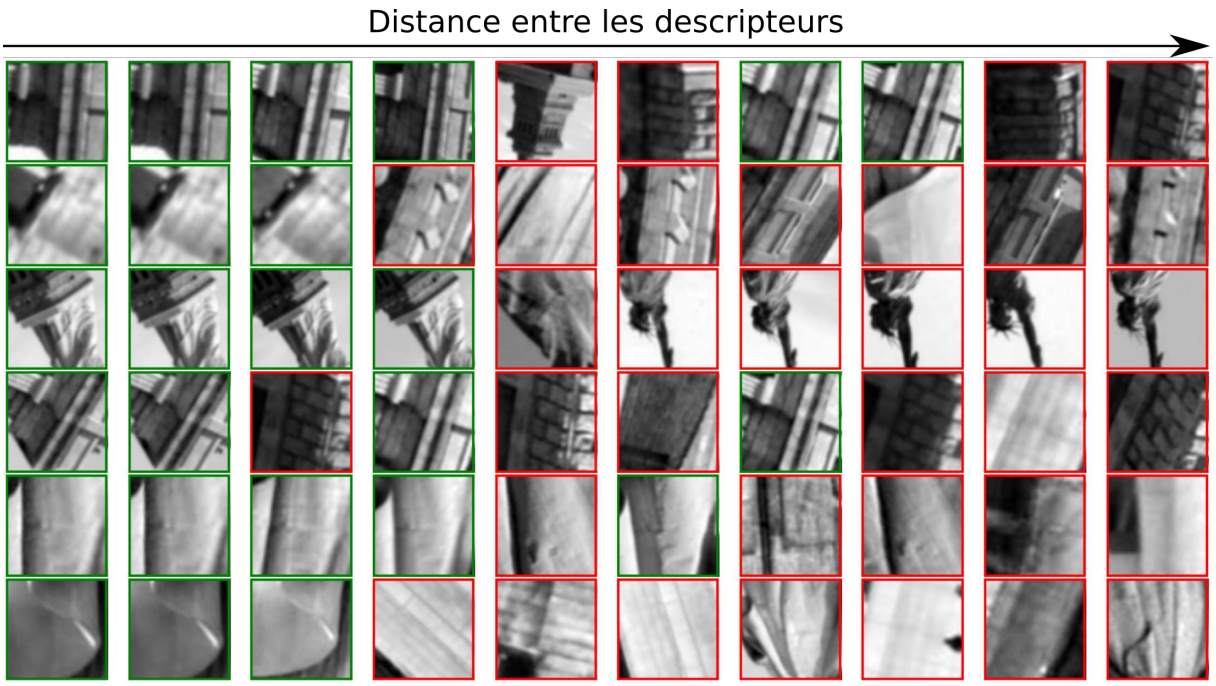


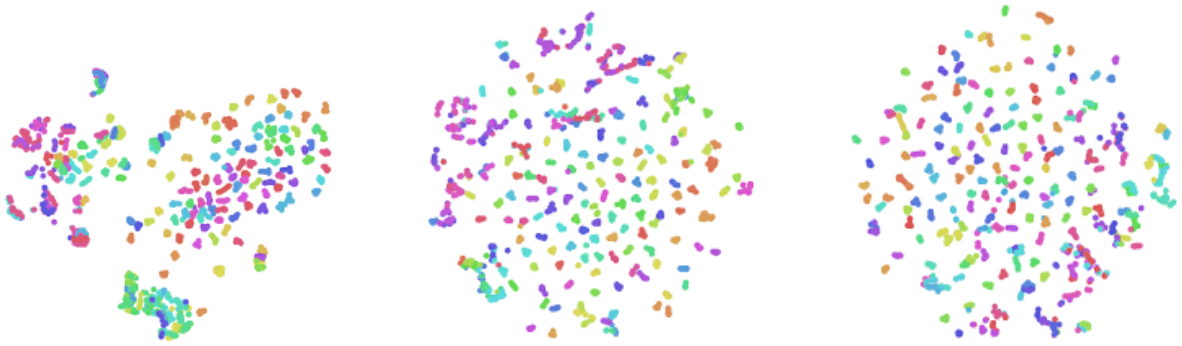
FIGURE 3.6 – Chaque ligne de cette figure représente les 9 premiers éléments dont la distance entre leurs descripteurs et celui de l'image cible (la première de chaque ligne) est la plus faible. Ce test a été effectué sur l'ensemble de données nommé *Liberty*, avec un réseau entraîné sur l'ensemble de données *Yosemite*.

Type de descripteur	ensemble de données	FPR95	Distance moyenne entre les points de repère anatomiques
3D-SURF	-	0.077	8.74 mm
Triplets	*Base_synthétique*	**0.007**	**8.54 mm**
	Base_FROG	0.05	9.59 mm
Listes	*Base_synthétique*	0.027	9.84 mm
	Base_FROG	0.009	8.65 mm

TABLEAU 3.4 – Ce tableau compare les meilleurs entraînements obtenus sur les ensembles de données (d'entraînement) «*Base_FROG*» et «*Base_synthétique*». Ces résultats ont été obtenus calculés sur l'ensemble d'images composant le groupe Gold de la base VISCERAL.

L'optimisation du réseau est effectuée par l'algorithme SGD et le taux d'apprentissage est fixé à 0.1. Nous définissons également le nombre de sous-volumes à 400k, la taille des mini-lots à 300 et la marge à 1.

Notre modèle est suffisamment léger pour être entraîné sur CPU et ainsi permettre d'approfondir l'exploration des hyperparamètres. Effectivement, quelques essais chronométrés ont révélé qu'un entraînement sur GPU ne réduisait pas significativement la durée de l'apprentissage, car le temps requis pour le processus d'apprentissage provenait principalement des Entrée/sorties (ES). Notre implémentation est faite grâce à la bibliothèque PyTorch, voir [Paszke et al., 2019]. L'apprentissage d'une époque avec 500k triplets s'effectue en environ 20 minutes et utilise 10 Go de mémoire sur une plateforme Linux 64-bit disposant d'Intel Xeon 2.6 GHz. Notre



(a) Parcelles d'images autour des points clés

(b) Parcelles d'images décrites par 3D-SURF

(c) Parcelles d'images décrites par le réseau de neurones

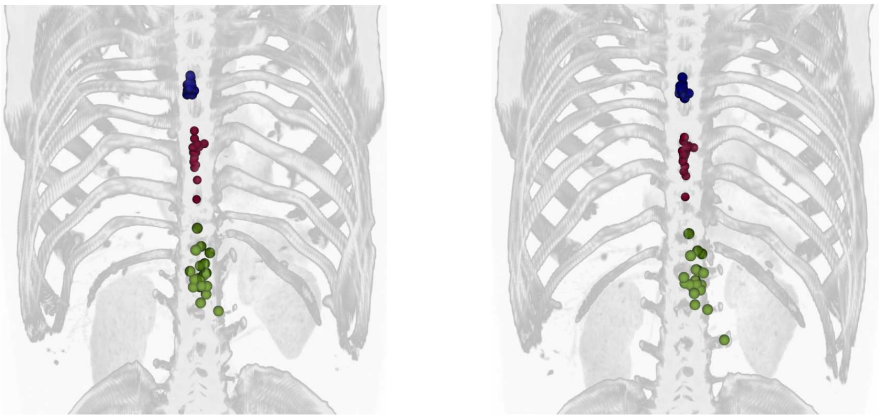
FIGURE 3.7 – Dans ces trois schémas, des parcelles d'images sont extraites autour de points clés. Dans 3.7(a), une réduction des dimensions est effectuée directement sur les images. Dans les deux autres schémas 3.7(b), 3.7(c), les images sont d'abord décrites par 3D-SURF (3.7(b)) ou le réseau appris (3.7(c)) puis une réduction de dimensionnalité est effectuée et enfin les points sont affichés.

recherche d'hyperparamètres a été effectuée avec Ray Tune et nous avons suivi l'évolution de l'entraînement grâce à «*Weights & biases*».

Nous avons tout d'abord effectué nos apprentissages en nous basant sur le recalage FROG pour calculer notre vérité terrain. Cette méthode d'apprentissage ne donnant pas de résultats très probants, nous avons entraîné de nouveau notre descripteur. Ce second apprentissage a été effectué en se basant sur la méthode de recherche de la correspondance entre les images décrite dans la Section 2.2. Le Tableau 3.3 compare plusieurs métriques, de meilleurs résultats obtenus de différents entraînements sur ces deux ensembles de données. Dans ce tableau, en comparaison aux entraînements en 2D, nous pouvons voir que l'apprentissage par triplets donne de meilleurs résultats en comparaison de l'approche par listes.

Pour illustrer l'effet de l'apprentissage des descripteurs, la Figure 3.7 montre la projection 2D des descripteurs de points clés calculés via la méthode t-SNE. Dans cette figure, 200 points clés ont été sélectionnés au hasard dans l'ensemble de validation. Chaque point clé étant présent dans 10 volumes transformés, nous obtenons un total de 2000 points clés. Les points clés correspondants sont affichés avec la même couleur. La Figure 3.7(a) montre la projection 2D des parcelles d'origine, tandis que la Figure 3.7(c) montre la projection des descripteurs calculés par notre réseau de neurones basé sur l'apprentissage siamois en triplets. Nos descripteurs produisent des groupes plus serrés, avec une meilleure distribution spatiale globale, similaire à celle des descripteurs 3D-SURF, présentée dans la Figure Figure 3.7(b).

Pour illustrer la répartition des points de repère après le recalage des images, la Figure 3.8 montre 3 types différents de points de repère dans l'espace commun : la bifurcation trachéale (en bleu), la vertèbre thoracique 9 (en rouge) et le processus xiphoïde (en vert). Le recalage a été effectué avec l'algorithme FROG en utilisant 3D-SURF (Figure 3.8(a)) et nos descripteurs appris (Figure 3.8(b)). Comme chaque point de repère est présent chez 20 sujets du groupe Gold, il y a un total de 60 points de repère. Le meilleur cas de recalage des points de repère, pour les deux descripteurs, correspond à la bifurcation trachéale (en bleu), avec une



(a) Descripteurs 3D-SURF. **(b)** Descripteurs appris.

FIGURE 3.8 – Recalage de différents types de points de repères anatomiques dans un espace commun pour 20 patients : bifurcation trachéale (meilleur cas, bleu, en haut), vertèbre thoracique n° 9 (cas intermédiaire, rouge, au milieu) et processus xiphoïde (pire cas, vert, en bas), recalés avec les descripteurs 3D-SURF (à gauche) et nos descripteurs (à droite).

distance moyenne de 4,8 mm pour notre réseau de description et 4,6 mm pour 3D-SURF. Le pire cas, pour les deux descripteurs, correspond au processus xiphoïde (en rouge) avec une distance moyenne de 14,4 mm pour le réseau et 12,7 mm pour 3D-SURF. La vertèbre thoracique 9 (en bleu) correspond à un cas intermédiaire, avec une distance moyenne de 8,4 mm pour le réseau de neurones et 10,5 mm pour 3D-SURF. L'analyse montre que notre méthode donne de meilleurs résultats que 3D-SURF pour tous les points de repère vertébraux.

Nous avons effectué quelques tests supplémentaires sur le type de transformations appliquées aux volumes. En utilisant le module «*RandomElasticDeformation*» fourni par la librairie Torchio, nous pouvons appliquer une déformation élastique aléatoire dense. Ces transformations n'ont pas permis d'améliorer les résultats.

3.4 Conclusion

Nos résultats montrent qu'un descripteur 3D entraîné sur des données semi-synthétiques, peut égaler et même surpasser un descripteur traditionnel. Cependant, les résultats obtenus sur la distance moyenne entre les points de repère anatomiques ne permettent pas d'évaluer spécifiquement le descripteur de points. En effet, le recalage obtenu par FROG permet d'évaluer à la fois le descripteur et le détecteur, ainsi que la méthode d'appariement de ces descripteurs ; nous restons dépendant de la méthode de détection des points clés, 3D-SURF.

C'est pourquoi le prochain chapitre portera sur l'apprentissage d'un détecteur de points clés jumelé à un descripteur.

Nous aimerions également étendre cette méthode d'apprentissage de descripteurs à des images multimodales. Cependant, ce nouvel entraînement est compliqué à mettre en œuvre en raison du manque de données IRM médicales dont nous disposons, ainsi que de leur qualité et de nos contraintes. Pour effectuer nos entraînements, nous devons idéalement disposer

d'images acquises de différents patients, prises dans la même position pour les deux modalités. Une autre option à explorer serait la génération d'images d'IRM à partir d'images de scanner en utilisant des réseaux antagonistes génératifs (angl. «*Generative adversarial networks*», GAN).

4

Entraînement d'extracteurs de bout en bout

Sommaire

Il a été démontré dans de nombreux articles que, l'apprentissage couplé du descripteur et du détecteur offrait de bien meilleurs résultats que des apprentissages réalisés séparément et utilisés à la chaîne, voir [Cuiyin et al., 2021]. C'est pourquoi dans ce nouveau chapitre, nous nous sommes concentrés sur des apprentissages de bout en bout, c'est-à-dire optimisant les étapes de détection et de description de points clés.

Nous nous concentrerons sur trois réseaux extracteurs de points clés. Les deux premiers sont des extensions 3D de deux extracteurs de points clés populaires en 2D, SuperPoint et R2D2. Le troisième extracteur utilise une fonction de perte optimisant directement la métrique de répétabilité entre les points clés. Le but de cette partie est d'explorer et d'évaluer les performances de ces détecteurs et descripteurs utilisant des approches entièrement supervisées ou à supervision distante, c'est-à-dire une supervision non pas directement sur la sortie de notre réseau, mais uniquement sur la connaissance des transformées entre les images.

Tout d'abord, nous examinerons en détail les principes et les caractéristiques de SuperPoint et R2D2, ainsi que leur extension en 3D. Nous discuterons des défis associés à cette extension, ainsi que des améliorations potentielles en termes de détection et de description de points clés. Des expériences et des analyses rigoureuses seront menées pour comparer les performances de ces détecteurs étendus en 3D avec 3D-SURF.

Ensuite, nous aborderons la mise en œuvre d'un nouvel extracteur de points clés, en présentant les motivations et les objectifs de ce développement. La nouvelle fonction de perte, conçue pour améliorer la qualité et la robustesse des points clés extraits, sera décrite en détail. Nous comparerons les performances de cet extracteur avec celles des méthodes SuperPoint et R2D2 étendues en 3D, ainsi qu'avec d'autres techniques de l'état de l'art. Des expériences et des évaluations approfondies seront menées pour valider l'efficacité et l'exactitude de cet extracteur, ainsi que son impact sur le recalage à grande échelle.

Les résultats obtenus dans cette section permettront d'éclairer la recherche future sur la détection et la description de points clés, et de contribuer à l'amélioration des techniques de recalage à grande échelle.

4.1 Entraînement de réseaux en 3D

Dans cette partie, nous allons décrire quelques éléments communs à chaque entraînement des réseaux que nous utiliserons par la suite.

Ensemble de données et découpage - Durant chaque entraînement des réseaux, nous utilisons tous les volumes provenant de la base de données VISCERAL, auxquels nous avons

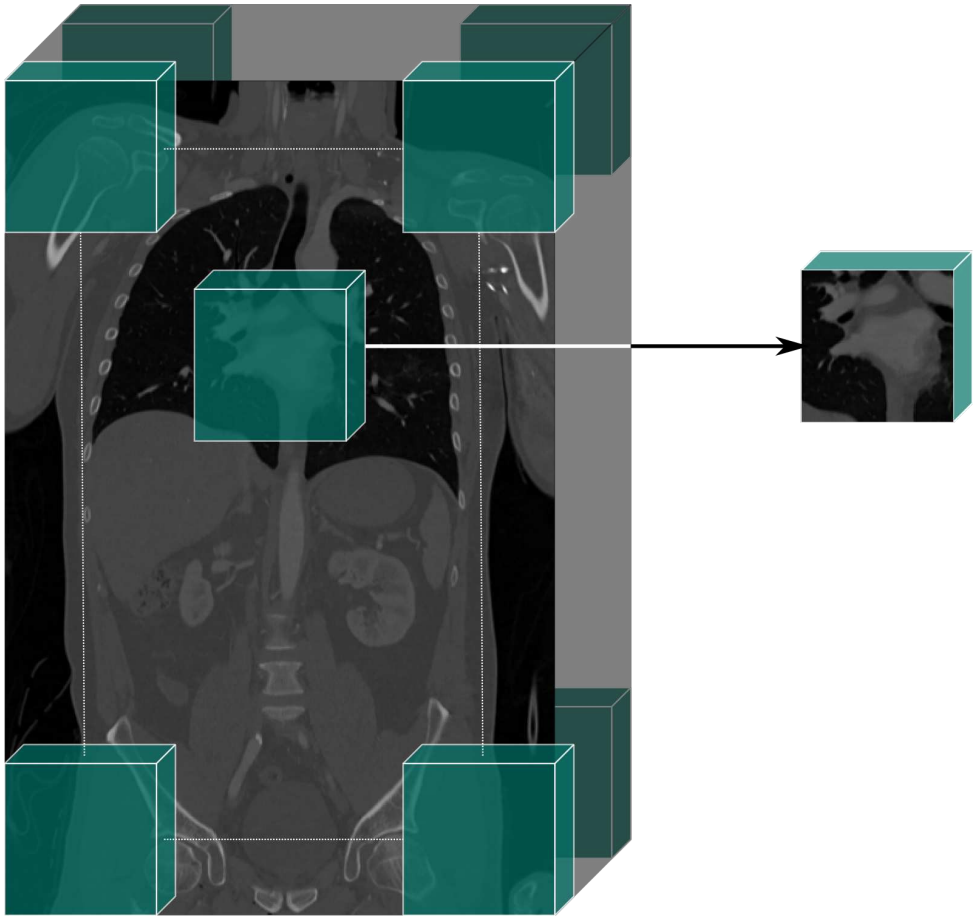


FIGURE 4.1 – Schéma représentant le découpage d'un volume utilisé lors de l'entraînement et de l'inférence des réseaux. Chaque cube vert représente un sous-volume extrait du volume du patient original. Ces cubes sont extraits tout le long du volume du patient.

ajouté 40 volumes provenant de la base de données de Saint-Étienne. Nous avons découpé notre nouvel ensemble de données de la façon suivante : la base d'entraînement est composée de 100 sujets, 20 sujets pour la base de validation, et 20 sujets provenant du groupe Gold pour les tests. Pour des soucis de compréhension, nous nommerons cette nouvelle base de données «*VISCERALST*». Le groupe des 100 sujets d'entraînement, 20 sujets de validation et les 20 derniers sujets du groupe Gold pour les tests seront respectivement nommés : «*VISCERALST_E*», «*VISCERALST_V*» et «*VISCERALST_T*».

Chaque volume sera utilisé complètement, c'est-à-dire que nous utiliserons le volume dans son entièreté lors de l'entraînement des réseaux. Cependant, l'entraînement des différents réseaux en 3D demande beaucoup plus de mémoire GPU qu'en 2D. C'est pourquoi nous avons dû adapter l'entraînement pour le rendre réalisable sur les volumes entiers. Pour ce faire, nous appliquons le réseau sur des parcelles de volumes, que nous appelons «*sous-blocs*» ou «*sous-volumes*», comme l'illustre la Figure 4.1. Un mini-lot d'entraînement sera donc composé d'un ou plusieurs sous-volumes.

Dynamique Minimale	Dynamique Maximale	Distance moyenne entre les points de repère anatomiques	Écart type
-1500	2096	8.98 mm	7.65
0	200	8.45 mm	6.97
-50	200	8.46 mm	7.08
-50	250	**8.37 mm**	**7.00**

TABLEAU 4.1 – Tableau montrant l'influence de la dynamique (plage de valeurs) lors du recalage des 20 volumes provenant du groupe Gold dans la base VISCERAL.

Entraînement - Nous avons effectué nos recherches d'hyperparamètres, au cours de chaque étape d'entraînement des réseaux, en employant la même méthode que celle décrite dans le Chapitre 3. Nous optimisons les prochains réseaux à l'aide d'ADAM avec un taux d'apprentissage de 0.001. Nos entraînements ont été réalisés sur une plateforme Linux 64-bit et sur des GPU V100 NVIDIA.

Évaluation - Nous avons évalué les différents réseaux extracteurs en utilisant trois métriques. La première calcule la répétabilité entre les points clés. Après extraction des points clés, nous obtenons deux ensembles P et P_t à partir de deux volumes V et V^t (avec V^t le volume V transformé par t). En utilisant t^{-1}, nous transformons les points P^t vers P. Avec un simple arbre k-d, nous calculons les distances entre P et $P^{t^{-1}}$. Pour une distance inférieure à un seuil donné entre deux points, nous considérons que ces deux points sont répétés.

La seconde métrique est la correspondance entre les descripteurs, qui mesure la capacité du réseau à discriminer les points clés. Pour deux caractéristiques extraites d et d_2 correspondant à la même localisation dans V et V^t, et un ensemble de descripteurs D_2 extraits de V^t, nous considérons qu'il y a correspondance entre d et d_2 lorsque la similarité est supérieure à celle existante entre d et tous les descripteurs de D_2.

Enfin, la troisième métrique permet d'obtenir la distance moyenne entre les points de repère anatomiques ; cette distance est calculée sur les marqueurs anatomiques dans les volumes du groupe Gold après les avoir recalés dans un espace commun à l'aide de la méthode FROG. Une faible distance entre les points de repère anatomiques indique de bons résultats, contrairement à la répétabilité et au score d'appariement.

Influence de la dynamique des images - Le Tableau 4.1 montre l'influence de la modification de la dynamique des volumes lors du recalage. On peut voir dans ce tableau qu'une restriction de la dynamique entre les plages de valeurs $[-50; 250]$ permet d'obtenir un meilleur recalage et donc une distance moyenne entre les points de repère anatomiques plus faible (ici 8.37 mm). C'est pourquoi, lors de nos futurs entraînements, nous limiterons la dynamique des volumes entre les valeurs $[-50; 250]$.

L'amélioration du recalage par la simple réduction de la dynamique est probablement due au fait que les structures du corps humain, telles que les os et les organes, restent visibles et apparaissent plus distinctement. Il est également important de souligner que les volumes

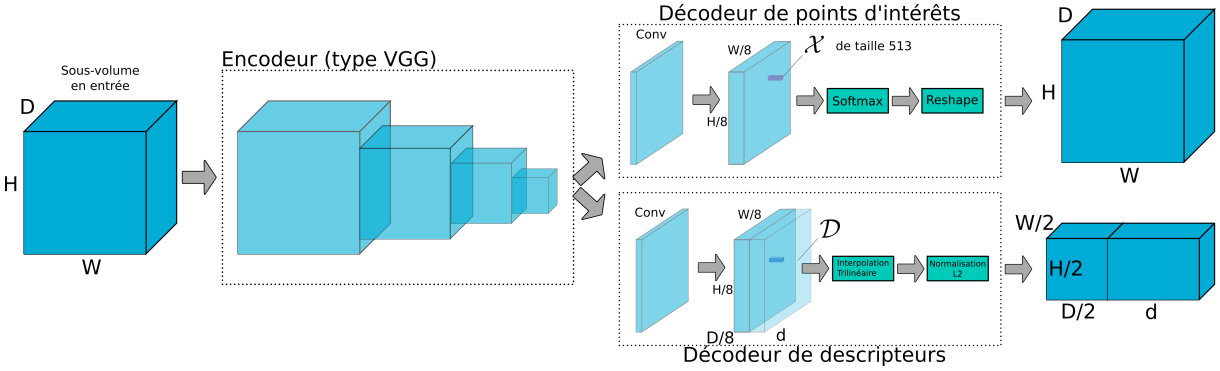


FIGURE 4.2 – Schéma représentant l'architecture du réseau SP3D. Dans ce schéma, $[H, W, D]$ représentent les dimensions du sous-volume en entrée du réseau et d la taille des descripteurs. De gauche à droite, nous avons un premier cube représentant le volume en entrée du réseau, qui est passé dans l'encodeur de type VGG. La sortie de l'encodeur est utilisée doublement, c'est-à-dire à la fois pour le détecteur et pour le descripteur. Dans le schéma, $\mathcal{X}$ correspond à l'ensemble des canaux d'un même voxel, i.e $\mathcal{X} \in \mathbb{R}^{H_c \times W_c \times D_c \times 513}$

de scanner d'abdomens et de thorax provenant de la base de données VISCERAL ont été améliorés en termes de contraste, dans le but d'obtenir une résolution plus précise en vue d'une segmentation fine des différents organes et os.

4.2 Extension et entraînement du réseau SuperPoint : SP3D

Nous allons tout d'abord aborder, dans cette section, les divers concepts essentiels à l'apprentissage du réseau SuperPoint, qui doivent être adaptés à la 3D afin d'obtenir une version tridimensionnelle du réseau (voir Figure 4.2). Par la suite, nous examinerons les deux jeux de données que nous avons employés et discuterons de la stratégie d'apprentissage que nous avons mise en œuvre pour obtenir un réseau performant. Nous désignerons par SuperPoint3D (SP3D) l'extension de SuperPoint en 3D.

4.2.1 Extension et entraînement

La première adaptation nécessaire pour réaliser un entraînement en 3D concerne le réseau de neurones lui-même, dont une illustration est présentée dans la Figure 4.2. Outre l'ajustement en 3D de différents éléments, tels que l'encodeur de type VGG ou l'interpolation des descripteurs, un élément spécifique appelé «*Reshape*» sur la Figure 4.2 doit être étendu à la 3D. De plus, il est essentiel d'étendre les fonctions de perte à la 3D.

Remaniement des pixels («*Reshaping*») - La méthode nommée réseau neuronal convolutionnel efficace sous-pixel (angl. «*Efficient Sub-pixel Convolutional Neural Network*», ESPCN) proposée par [Shi et al., 2016] et utilisée par SuperPoint permet de remettre à l'échelle la sortie

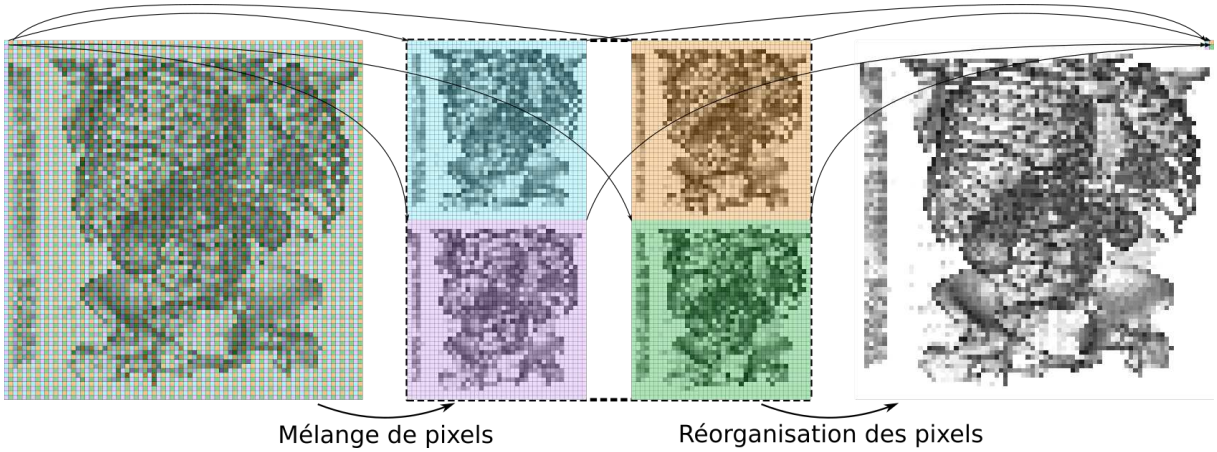


FIGURE 4.3 – Schéma représentant les fonctions de mélange et réorganisation des pixels par les fonctions «*pixel shuffle*» et «*pixel unshuffle*». En partant d'une image complète et en se déplaçant de gauche à droite dans la Figure, l'opération de «*Pixel Shuffle*» découpe d'abord l'image en blocs de pixels contigus, où chaque bloc représente un groupe de canaux. Ensuite, les pixels sont réarrangés en fonction des différents canaux qui leur sont attribués (étape au centre de la Figure). Enfin, les blocs sont concaténés pour former un tenseur de sortie avec une résolution spatiale plus basse mais avec un plus grand nombre de canaux. L'opération «*Pixel Unshuffle*» est l'opération exactement opposée qui permet de restaurer la résolution et le nombre de canaux d'origine. Les pixels retrouvent ainsi leur position spatiale initiale.

du décodeur en réorganisant tous les pixels de chaque canal en une seule image. Cette réorganisation des pixels est appelée «*pixel shuffle*» dans la librairie PyTorch [1] dans la bibliothèque Pytorch. Cette approche permet au réseau d'apprendre une meilleure correspondance entre la basse et la haute résolution par rapport à un filtre d'interpolation standard. Grâce à la diminution de la taille de l'image d'entrée, une taille de filtre plus petite peut être utilisée pour extraire des caractéristiques. La complexité computationnelle et le coût en mémoire sont réduits, augmentant ainsi considérablement l'efficacité. C'est pourquoi l'ESPCN est devenu un choix idéal pour la super-résolution des vidéos HD en temps réel et est mis en œuvre ici.

L'algorithme du pixel shuffle (voir Figure 4.3) prend en entrée un tenseur (généralement une image ou un ensemble de canaux dans notre cas) et un facteur d'échelle r. Un tenseur est un tableau à plusieurs dimensions, il permet de généraliser la notion de matrice et de vecteur. Il va ensuite réorganiser les éléments du tenseur d'entrée de manière à augmenter sa résolution spatiale tout en réduisant le nombre de canaux. En d'autres termes, le pixel shuffle répartit les informations contenues dans les canaux sur les dimensions spatiales (hauteur et largeur) de l'image.

Selon [DeTone et al., 2017 ; Odena et al., 2016], les couches de suréchantillonnage ont tendance à ajouter une grande quantité de calculs et peuvent introduire des artefacts en forme de damiers, non désirables. C'est pourquoi SuperPoint utilise cette méthode de super-résolution explicite afin de réduire les calculs du modèle.

1. PixelShuffle

Encodeur et fonctions de pertes - Comme nous l'avons vu dans la Section 1.5.4, Superpoint utilise une fonction de perte double qui optimise le détecteur dans une branche et le descripteur dans une autre. Ces fonctions de pertes utilisent la même sortie de l'encodeur qui réduit la dimension du volume d'entrée, de taille $H \times W \times D$, à une dimension $H_c \times W_c \times D_c$, avec $\{H\}_c = \{H\}/8$, $\{W\}_c = \{W\}/8$ et $\{D\}_c = \{D\}/8$. Dans la suite de cette section, nous ferons référence à un voxel de sortie de l'encodeur, par le terme *vox_cell* représentant un sous-volume de taille 8^3 dans le volume en entrée.

Le passage d'un volume dans le réseau et ces deux décodeurs permet de générer deux nouvelles cartes de réponses. La première correspond à une carte de probabilité de détection, chaque voxel en sortie peut être un point clé à extraire. Le volume en sortie de la convolution de l'encodeur prend la forme $\mathcal{X} \in \mathbb{R}^{H_c \times W_c \times D_c \times 513}$, puis une étape de Softmax permet d'obtenir la probabilité de voir apparaître un point clé à travers les canaux, la dernière dimension (le dernier canal) est supprimée et l'étape de réorganisation des pixels permet d'obtenir un volume de taille égale à celle du volume d'entrée en sortie de détecteur, c'est-à-dire $\mathbb{R}^{H \times W \times D}$. Le nombre de 513 canaux s'explique par le fait que nous ajoutons un canal de biais aux 512 canaux obtenus par la sortie de l'encodeur du réseau.

Pour la partie détection du réseau, chaque valeur en sortie correspond à une probabilité d'être un point d'intérêt dans le volume en entrée. La fonction de perte mise en place pour cette étape de détection est une fonction d'entropie croisée entre la sortie du décodeur et la vérité terrain. Dans cet entraînement, la vérité terrain correspond aux points clés que nous avons détectés avec 3D-SURF et le réseau préentraîné. Ce sont ces points clés que nous souhaitons voir appris par SP3D. Les points clés sont extraits dans les volumes de la base de données composée des images provenant des hôpitaux de Saint-Étienne et de la base VISCERAL, ces mêmes volumes qui nous servent de base d'apprentissage. La fonction de perte s'écrit ainsi :

$$\mathcal{L}_p(\mathcal{X}, \mathcal{Y}) = \frac{1}{H_c W_c D_c} \sum_{h=1,w=1,d=1}^{H_c,W_c,D_c} (l_p(\mathsf{x}_{hw}; y_{hw})) \tag{4.1}$$

où $\mathsf{x}_{hw} \in \mathcal{X}$ correspond à chaque vecteur à travers les canaux en sortie de l'encodeur (voir Figure 4.2), $\mathcal{Y}$ la vérité terrain fournie au réseau (y_{hw} étant une entrée individuelle de $\mathcal{Y}$) et enfin :

$$l_p(\mathsf{x}_{hwd}; y_{hwd}) = -\log\left(\frac{\exp(x_{hwdy})}{\sum_{k=1}^{513} \exp(\mathsf{x}_{hwdk})}\right) \tag{4.2}$$

Concernant l'étape de description, dans la version originale de Superpoint, la partie description permettait d'obtenir un descripteur par pixel dans l'image en entrée. Dans notre extension SP3D, nous avons fait le choix de ne conserver qu'un seul vecteur de description pour une zone de 4^3 voxels. Ce choix a été motivé par deux éléments, le premier provient de l'architecture même de Superpoint. Suite à l'encodage de l'image par le réseau VGG en 2D, un calcul softmax est appliqué à travers les canaux obtenus (dans la partie détection) et ne permet donc la détection que d'un point clé par zone de 8×8 pixels ($8 \times 8 \times 8$ voxels en 3D) ; il n'y aura donc qu'un seul descripteur utilisé dans cette zone de 8×8 pixels ($8 \times 8 \times 8$ voxels

en 3D). C'est pourquoi en 3D nous ne calculons qu'un seul descripteur pour cette zone. Le second élément provient de l'empreinte mémoire nécessaire à l'entraînement du réseau, cette réduction du nombre de descripteurs permet également de réduire le nombre de calculs et la mémoire nécessaire à l'entraînement du réseau.

L'entraînement du descripteur Superpoint se fait à l'aide d'une double fonction de perte charnière (angl. «*hinge loss*»). La première perte charnière tend à rassembler les descripteurs correspondant à la même localisation dans l'espace de description ; la seconde permet de discriminer les descripteurs provenant de localisations différentes.

Cette fonction de perte est appliquée à chaque paire de descripteurs existante entre les cartes de descriptions $\{\mathcal{D}, \mathcal{D}'\}$ de taille H_c, W_c, D_c correspondants respectivement aux volumes $\{\mathcal{I}, \mathcal{I}'\}$ en entrée du réseau. Les correspondances entre les deux volumes résultants de l'encodeur VGG sont obtenues à l'aide de la matrice de transformation affine nommée $\mathcal{H}$. Selon la notation de Superpoint, la correspondance induite par la matrice de transformation nous permet de faire correspondre le voxel (h, w, d) d'une image, avec le voxel noté (h', w', d') d'une seconde image transformée par $\mathcal{H}$. Cette correspondance entre deux paires peut s'écrire de la façon suivante :

$$s_{hwd,h'w'd'} = \begin{cases} 1, & si||\widehat{\mathcal{H}p_{hwd}} - p_{h'w'd'}|| \leq 8 \\ 0, & sinon \end{cases} \tag{4.3}$$

où p_{hwd} correspond à la localisation du centre du voxel au sein d'un *vox_cell* et $||\widehat{\mathcal{H}p_{hwd}}||$ désigne la transformation de la coordonnée du point par la matrice affine. Dans cette équation, 8 correspond à la distance entre les centres de deux *vox_cell* et est exprimée en voxels. Comme pour Superpoint, nous avons conservé le terme λ_d qui permet de mitiger le fait que le réseau aura plus souvent à faire à de mauvaises correspondances (négatives) qu'à de bonnes correspondances (positives).

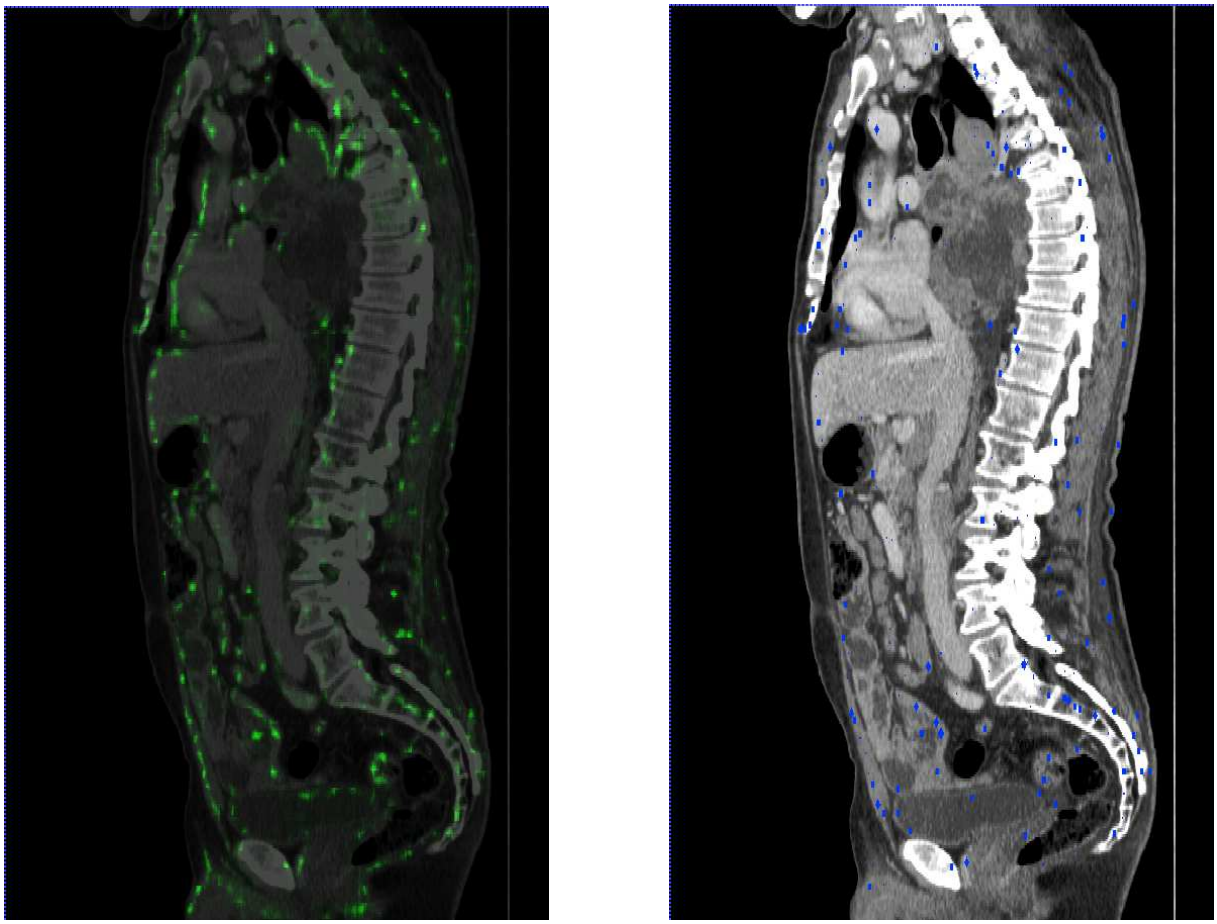
Pour le calcul final de la fonction de perte appliquée aux descripteurs, une double fonction de perte charnière avec une marge positive m_p et une marge négative m_n est employée. La fonction de perte en 3D s'écrit de la façon suivante :

$$\mathcal{L}_d(\mathcal{D}, \mathcal{D}', S) = \frac{1}{(H_c W_c D_c)^2} \sum_{\substack{h=1 \\ w=1 \\ d=1}}^{H_c, W_c, D_c} \sum_{\substack{h=1 \\ w=1 \\ d=1}}^{H_c, W_c, D_c} \sum_{\substack{h=1 \\ w=1 \\ d=1}}^{H_c, W_c, D_c} l_d(d_{hwd}, d'_{h'w'd'}; s_{hwd,h'w'd'}) \tag{4.4}$$

Avec

$$l_d(d, d'; s) = \Big(\lambda_d \ * \ s \ * \ \max(0, m_p - d^T d')\Big) + \Big((1 - s) \ * \ \max(0, d^T d' - m_n)\Big) \tag{4.5}$$

Le symbole S désigne l'ensemble des correspondances entre les deux volumes, d^T correspond au descripteur à la position hwd et d' celui à la position $h'w'd'$. La première partie de



(a) Réponse au détecteur SP3D.

(b) Points clés extraits par SP3D.

FIGURE 4.4 – La Figure 4.4(a) correspond à une coupe de patient sur laquelle la sortie du détecteur a été affichée, avec en vert les zones à forte réponse. Dans la Figure 4.4(b), la même coupe du patient est affiché, avec les points clés extraits de la carte de réponse.

l'équation ci-dessus permet d'éloigner les descripteurs qui ne correspondent pas, et la seconde partie de l'équation permet de rapprocher les descripteurs similaires.

4.2.2 Apprentissage et base de données

Concernant l'apprentissage de SP3D, nous avons suivi la méthode de Superpoint qui consiste à exécuter un double apprentissage. Dans un premier temps, nous avons généré une base de données constituée de formes 3D simples (voir Section 2.1.2). Cette base de données permet un préentraînement des poids du réseau qui n'est effectué que sur la partie détection du réseau. Ce réseau préentraîné à détecter les points clés dans des formes simples est appelé MagicPoint3D (MP3D). Cette première base de données est composée de 7 formes simples différentes, chacune regroupant 3000 volumes pour l'entraînement, et 300 volumes pour la validation.

Extracteur	Répétablité (2mm)	Correspondance des descripteurs	Distance moyenne entre les points de repère anatomiques
3D-SIFT	0.37	0.16	12.2 mm
3D-SURF	0.46	0.34	8.37 mm
SP3D	**0.51**	**0.48**	**7.98 mm**

TABLEAU 4.2 – Tableau montrant les performances obtenues avec SP3D, en comparaison avec 3D-SURF et 3D-SIFT.

Dans un second temps, nous nous basons sur la méthode décrite dans la Section 2.2 pour générer nos volumes desquels nous extrayons des points clés à l'aide de 3D-SURF. Nous appliquons ce processus en mêlant les points clés détectés par 3D-SURF, à ceux détectés par le réseau MP3D nouvellement entraîné sur les données entièrement synthétiques. Ce nouvel ensemble de points clés va permettre d'effectuer un nouvel entraînement afin de spécialiser le réseau sur nos volumes médicaux. Lors de ce nouvel entraînement, nous incluons tous les décodeurs, c'est-à-dire le détecteur et également le descripteur, et nous démarrons l'apprentissage avec comme base de poids (les paramètres du réseau), ceux du réseau MP3D. Cette méthode est appelée apprentissage par transfert (angl. «*transfer learning*», voir [Tan et al., 2018]).

Lors de ces deux entraînements, nous utilisons les volumes de la base entièrement synthétiques puis le groupe *VISCERALST_T*, ainsi que des listes de points clés associés comme référence pour la vérité terrain. Les positions de ces points clés sont connues dans les données entièrement synthétiques (lors de leur génération) où elles sont extraites à l'aide de 3D-SURF et du réseau MP3D. Ces deux apprentissages sont donc entièrement supervisés, car nous nous appuyons sur une vérité que nous souhaitons transmettre au réseau. De plus, nous utilisons les transformations entre chaque volume pour calculer la position des points clés détectés, ceci entre les différents volumes au sein d'un même patient.

Lors de l'optimisation du réseau, nous utilisons une taille de mini-lot de 1 sous-volume et sa version transformée, ainsi que les points clés associés à ces deux sous-volumes. Pour compenser ce faible nombre d'images, nous appliquons la méthode d'accumulation des gradients sur 8 volumes. Une seule époque, prenant en compte l'entraînement et l'évaluation, prend approximativement 1,5 heure et 20Go de mémoire sur une plateforme Linux 64-bit et sur des GPU V100 NVIDIA.

4.2.3 Inférence et résultats

Au cours de l'inférence, le réseau utilise la carte de réponse au détecteur pour extraire les points clés. Suite au calcul de cette carte de réponse, un algorithme de suppression des valeurs non maximales est utilisé pour extraire les coordonnées dont la réponse est supérieure à un seuil fixé à 0.010 (voir Figure 4.4). Nous avons fixé cette valeur empiriquement afin d'extraire au moins 20000 points clés par patient.

Le Tableau 4.2 montre que SP3D donne de meilleurs résultats pour toutes les mesures en comparaison à 3D-SIFT et 3D-SURF, avec une distance moyenne entre les points de repères anatomiques de 7.98mm pour SP3D.

4.3 Extension du réseau R2D2 : R2D3

Dans un second temps, nous avons étendu un réseau utilisant une approche de supervision distante. Nous qualifions cette supervision de «*distante*», car elle n'est pas complètement supervisée. En effet, nous ne fournissons pas directement au réseau les localisations que le réseau doit apprendre comme pour SP3D. Cependant l'entraînement de ce réseau n'est pas complètement non supervisé, une forme de supervision est mise en place par la connaissance exacte des transformations existantes entre les images.

Nous avons choisi d'utiliser un réseau non supervisé afin de nous affranchir d'un détecteur préexistant pour établir notre référence de vérité terrain. Notre objectif était de permettre au réseau d'apprendre de nouvelles localisations sans le contraindre à utiliser des localisations qui pourraient être contre-productives ou non reproductibles.

Fonction de pertes - Les formules mathématiques derrière les fonctions de pertes en 2D et 3D sont les mêmes que celles décrites dans la Section 1.5.6, c'est pourquoi nous ne les détaillerons pas ici. Le changement majeur dans cette partie viendra de l'implémentation des fonctions qui sera faite, en ajoutant une troisième dimension pour l'extraction des descripteurs et des points clés.

Encodeur et décodeur 3D - Durant nos premiers entraînements de ce réseau, nous avons fait face à un problème de mémoire GPU. En effet, le réseau VGG tel qu'il est employé dans le réseau original R2D2, ne réduit pas les dimensions du réseau à chaque étape. Cette non-réduction des images en 2D ne pose pas de souci majeur, mais elle ne permet pas un entraînement réaliste en 3D. C'est pourquoi nous avons choisi une architecture de réseau différente ; le réseau présenté par [Zhao et al., 2022] semblait plus cohérent en termes de taille et nombre de poids après adaptation aux fonctions de pertes de R2D2 en 3D (voir Figure 4.5).

Le réseau encodeur et décodeur est constitué de trois grandes parties :

- Encodeur : cette première partie du réseau prend un volume I de taille $H \times W \times D$ en entrée et lui applique une série de quatre blocs constitués de convolutions, max-pooling et ReLU. Le premier bloc est composé de deux couches de convolutions 3^3 et de la fonction d'activation ReLU. Les trois derniers blocs sont également composés de deux couches de convolutions 3^3 chacune suivie de la fonction ReLU et d'une étape de max-pooling. Les différentes étapes d'encodages voient la dimension du bloc originale réduite respectivement par 2, 4, 8 et 32.

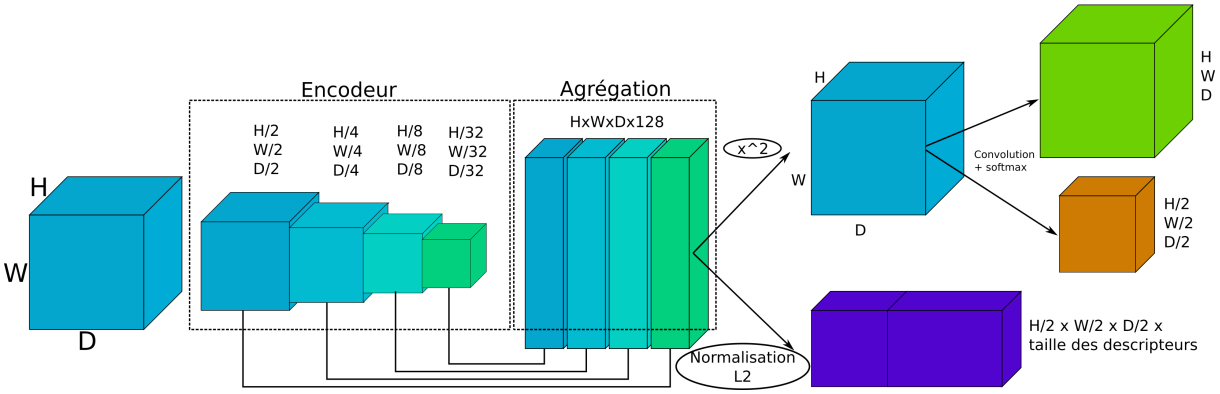


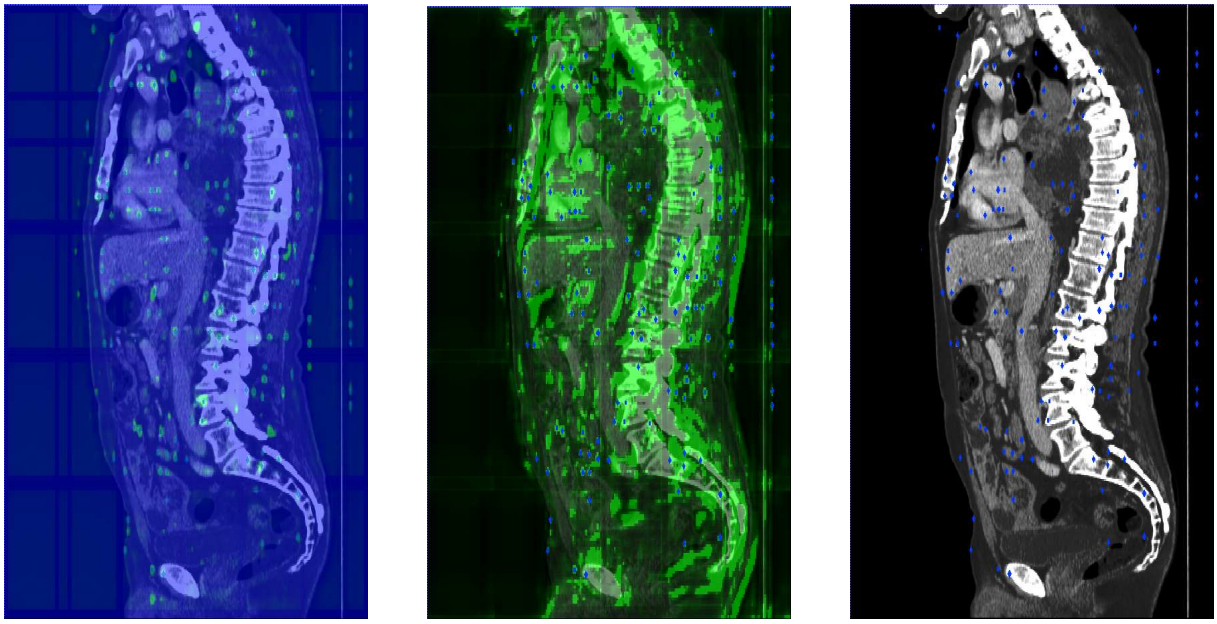
FIGURE 4.5 – Schéma représentant l'architecture du réseau R2D3, avec l'encodeur et l'agrégateur.

- Agrégation des canaux : cette seconde étape agrège les différents blocs précédemment évoqués, après les avoir redimensionnés à la taille originelle du volume d'entrée, à l'aide d'une interpolation trilinéaire et une convolution de taille 1 permettant d'adapter le nombre de canaux.
- Détection et description : la dernière partie, composée du détecteur et du descripteur, utilise de façon conjointe la sortie de la partie précédente. Le détecteur calcule tout d'abord l'opération du carré sur chaque élément, puis ce nouveau tenseur de valeurs sera utilisé pour former deux cartes de données. La première correspond à une carte de score de répétabilité et la seconde à une carte de fiabilité des détecteurs. Le descripteur, quant à lui, utilise une opération de normalisation L_2 sur chaque descripteur.

Pour comparaison, le réseau original utilisé par R2D2 étendu à la 3D, demande 17 Go de mémoire lors de l'apprentissage (lors de la propagation vers l'avant et la rétropropagation des gradients) et optimise 1M de paramètres pour des volumes de taille 100^3. Le nouvel encodeur/agrégateur optimise quant à lui 200k paramètres, pour 6Go de mémoire et des volumes en entrée de dimension 128^3. Ce gain de mémoire va nous permettre d'augmenter la taille des sous-volumes que nous fournissons au réseau durant l'entraînement, et ainsi permettre un apprentissage plus rapide et plus global sur nos données.

4.3.1 Jeu de données et entraînement

Durant l'entraînement de ce réseau, nous avons utilisé la base de données « *VISCERALST* ». Le réseau nécessite deux images en entrée, nous lui fournissons donc un sous-volume extrait d'un volume de patient, et une version transformée aléatoirement de ce sous-volume. Nous fournissons également la matrice de transformation qui lie les deux sous-volumes. Dans ce cas, l'apprentissage n'est pas entièrement supervisé, mais plutôt supervisé de manière distante. Cela signifie que le réseau n'utilise pas de référence de vérité terrain directe pendant l'apprentissage, mais plutôt une supervision distante basée sur la connaissance de la matrice de transformation affine entre les volumes et, par conséquent, entre les points clés.



(a) Carte de réponse au détecteur de R2D3.

(b) Carte de fiabilité de R2D3.

(c) Points clés extraits par l'algorithme de suppression non maximale.

FIGURE 4.6 – Figure montrant les cartes de réponses de R2D3
Les 3 Figures montrent les cartes de réponses de R2D3. La première à gauche (a) présente la réponse au détecteur, celle du centre montre la réponse de fiabilité attribuée à chaque point (b) et finalement la troisième présente les points clés extraits par l'algorithme de suppression non maximale au travers de la carte de réponse au détecteur et après suppression des points considérés comme «*non fiables*».

Lors de l'optimisation du réseau, nous utilisons une taille de mini-lot de 1 sous-volume et sa version transformée, ainsi que la matrice affine de transformation qui relie les deux sous-volumes. Pour compenser ce faible nombre d'images, nous appliquons la méthode d'accumulation des gradients, sur 16 volumes. Une seule époque, prenant en compte l'entraînement et l'évaluation, prend approximativement 2 heures et 25Go de mémoire sur une plateforme Linux 64-bit et sur des GPU V100 NVIDIA.

4.3.2 Inférence et résultats

La Figure 4.6 montre le résultat visuel du réseau R2D3, tant au niveau de la carte de répétabilité que de la carte de fiabilité.

Au cours de l'inférence, le réseau utilise la carte de réponse au détecteur pour extraire les points clés. Suite au calcul de cette carte de réponse, un algorithme de suppression des valeurs non maximales est utilisé pour extraire les coordonnées dont la réponse est supérieure à un seuil fixé à 0.6, voir Figure 4.4. Ensuite, seuls les points clés dont les scores de la carte de fiabilité sont supérieurs à 0 sont conservés. Nous avons fixé ces valeurs empiriquement afin d'extraire au moins 20000 points clés par patients. Finalement, nous pouvons choisir les N meilleurs points selon leurs scores de réponse au détecteur.

Extracteur	Répétablité (2mm)	Correspondance des descripteurs	Distance moyenne entre les points de repère anatomiques
3D-SIFT	0.37	0.16	12.2 mm
3D-SURF	**0.46**	**0.34**	**8.37 mm**
R2D3	0.45	0.21	11.3 mm

TABLEAU 4.3 – Tableau montrant les performances obtenues avec R2D2, en comparaison de 3D-SURF et 3D-SIFT.

Le Tableau 4.3 montre que R2D3 donne de moins bons résultats pour toutes les mesures en comparaison à 3D-SIFT et 3D-SURF. On peut voir dans ce tableau que les résultats que nous obtenons avec ce réseau à supervision distante sont bien en dessous de 3D-SURF. Le problème vient probablement de la taille des volumes en entrée que nous pouvons fournir au réseau. Effectivement, nous avons effectué divers tests en 2D et ensuite en 3D, en supposant que l'utilisation de volumes entiers en 3D ou d'images entières en 2D pendant l'entraînement, plutôt que notre découpage en sous-blocs, aurait un impact sur l'entraînement. Pour réaliser ces tests en 3D, nous avons réduit la dimension des volumes de 512^3 à 96^3, ce qui nous permet de directement entraîner le réseau sur des volumes «*entiers*». Pendant ces entraînements, nous avons évidemment détecté beaucoup moins de points clés, soit environ 200 points avec une répétabilité d'environ 0.60 selon les hyperparamètres. Les réseaux 2D utilisent également le réseau entraîné sur les images entières, puis lors de l'inférence, ce même réseau est utilisé à différentes tailles d'images, pour effectuer une forme de détection multiéchelle. Par exemple, pour R2D2, le réseau est appliqué à l'image originale, puis à la même image sous-échantillonnée de moitié à chaque nouvelle inférence du réseau, jusqu'à obtenir une image de taille inférieure à 128 pixels.

4.4 Réseau de Détection et Description multiéchelle Léger en 3D : D&D3LD

Nous avons pu étendre deux réseaux 2D à la 3D, le premier utilisant un entraînement supervisé et le second un entraînement supervisé de manière distante. Nous avons pu voir les avantages et inconvénients de ces différents réseaux, c'est pourquoi nous avons voulu proposer notre propre réseau permettant un apprentissage de bout en bout que nous nommerons Détection et Description multi-échelle Léger en 3D (D&DL3D). Ce réseau est basé sur différentes approches et notamment sur le réseau accurate and lightweight keypoint detection and descriptor extraction (ALIKE) (voir [2022]) pour la structure de l'encodeur et la fonction optimisant le détecteur. Nous avons choisi de ne pas employer la fonction de perte optimisant le descripteur d'ALIKE, car cette dernière demandait trop d'espace mémoire. Nous avons donc décidé d'utiliser une fonction de perte par triplets pour l'entraînement du descripteur.

4.4.1 Réseau et fonction de perte

Concernant le choix de l'encodeur, nous avons de nouveau utilisé l'encodeur/agrégateur décrit dans la Section 4.3 précédente.

Fonction de perte optimisant la répétabilité - La métrique de répétabilité d'un détecteur de points se mesure directement en analysant la localisation des points clés détectés. Cette métrique est régulièrement utilisée comme évaluation de performance d'un détecteur de points clés, mais pas en tant que fonction de perte utilisable pour l'entraînement d'un réseau détecteur de points caractéristiques. Or, une répétabilité importante est l'une des premières caractéristiques que nous souhaitons atteindre dans notre détecteur, c'est pourquoi nous souhaitons mettre en place une fonction de perte optimisant directement cette métrique.

Prenons deux volumes, V_A et V_B, avec V_B le volume V_A transformé par une matrice affine, notée $\mathcal{H}$. Une détection des points clés dans ces deux volumes nous permet d'obtenir deux ensembles de points clés P_A et P_B. Il est alors possible de transformer ces deux ensembles de points vers l'espace de l'un ou l'autre des volumes selon l'équation suivante :

$$\begin{aligned} P_{A\to B} &= \mathcal{H}(P_A) \\ P_{B\to A} &= \mathcal{H}^{t-1}(P_B) \end{aligned} \tag{4.6}$$

Il est maintenant possible de calculer la répétabilité d'un réseau selon l'algorithme suivant :

Données : Deux listes de coordonnées P et $P_{B\to A}$
Résultat : TauxDeRépétabilité = 0
pour *chaque point* p_{A_i} *dans* P_A **faire**
  $j = 0$, $d = \infty$
  tant que *(d > seuil) et (*$j < |P_B|$*)* **faire**
    d = DistanceEuclidienne(p_{A_i}, p_{B_j})
    si *d <= seuil* **alors**
      TauxDeRépétabilité += 1
    $j+ = 1$
TauxDeRépétabilité /= i

Algorithme 2 : Algorithme du calcul de la répétabilité entre les points clés provenant de deux volumes. Avec P une liste de coordonnées dans l'image A et $P_{B\to A}$ une liste de coordonnées dans l'image B transfomées vers l'espace de l'image A par une matrice affine.

En nous basant sur la distance de Chamfer [Huang et al., 2020], qui permet de calculer la distance entre deux nuages de points et sur l'Algorithme 2 de calcul du taux de répétabilité, nous pouvons écrire notre fonction de perte optimisant la répétabilité. En premier lieu, nous définissons une fonction $||\min(p, P)||_2$, qui calcule la distance euclidienne entre un point p et

son point le plus proche dans l'ensemble P. La perte de répétabilité est calculée en additionnant le carré des distances entre les correspondances des plus proches voisins de deux ensembles de points, selon l'équation suivante :

$$\mathcal{L}_{\text{répétabilité}}(P_A, P_B, P_{A\to B}, P_{B\to A}) = \frac{1}{|P_A| + |P_B|}\Bigg(\sum_{p\in P_A} ||\min(p, P_{B\to A})||_2^2 + \sum_{p\in P_B} ||\min(p, P_{A\to B})||_2^2 \Bigg) \tag{4.7}$$

Pour utiliser la fonction de perte, nous avons besoin de la localisation de points clés détectés. Ce seront ces localisations qui seront optimisées pendant l'entraînement. Cependant, la détection de points clés à l'aide d'une étape de NMS n'est pas une opération différentiable en raison de la nature discrète de ces valeurs. En effet, la NMS classique consiste simplement à choisir un pixel (ou voxel) ayant le score le plus élevé dans une région et supérieur à un seuil comme candidat à être un point clé, de sorte que la position du point clé détecté ne peut pas être optimisée directement. La plupart des réseaux n'utilisent la suppression non maximale que sur la carte de réponse intermédiaire générée par le réseau de neurones, comme on peut le voir dans [DeTone et al., 2017 ; Dusmanu et al., 2019 ; Revaud, Weinzaepfel et al., 2019], ce qui est contraire à notre objectif. C'est pourquoi nous nous sommes tournés vers des méthodes d'extraction de localisations de points clés différentiables (voir [Hosang et al., 2017 ; Zhao et al., 2022]) que nous allons aborder dans le paragraphe suivant.

4.4.2 Détection des points clés

Les créateurs du réseau, nommé ALIKE, ont présenté une version différentiable de la NMS nommée DKD (differentiable keypoint detection), que nous allons utiliser et donc décrire maintenant. Cette fonction est illustrée sur la Figure 4.7.

Pour détecter des points clés dans une carte de réponse au détecteur notée S, la méthode de la NMS est généralement utilisée, voir [DeTone et al., 2017 ; Hosang et al., 2016]. Cette opération, équivalente à la fonction $\arg\max$, va permettre de trouver un score maximal dans une fenêtre locale de taille $N \times N$ et peut s'écrire sous la forme :

$$[\hat{i}, \hat{j}]_{NMS} = \arg\max_{i,j}\{s(i,j) | 0 < i, j < N\} \tag{4.8}$$

où $s(i,j)$ représente la carte de score en réponse au détecteur à la position (i,j). Dans cette formulation, la sortie de la fonction est une position dans la carte de réponse S, ce qui rend cette fonction non différentiable et donc ne permet pas de bénéficier de l'apprentissage par rétropropagation des gradients.

C'est pourquoi ALIKE se propose de coupler les points clés avec la carte de réponse, en extrayant les points clés dans une fenêtre locale à l'aide de la fonction $softargmax$. En premier lieu, une suppression des scores non maximum est mise en œuvre. Cette suppression

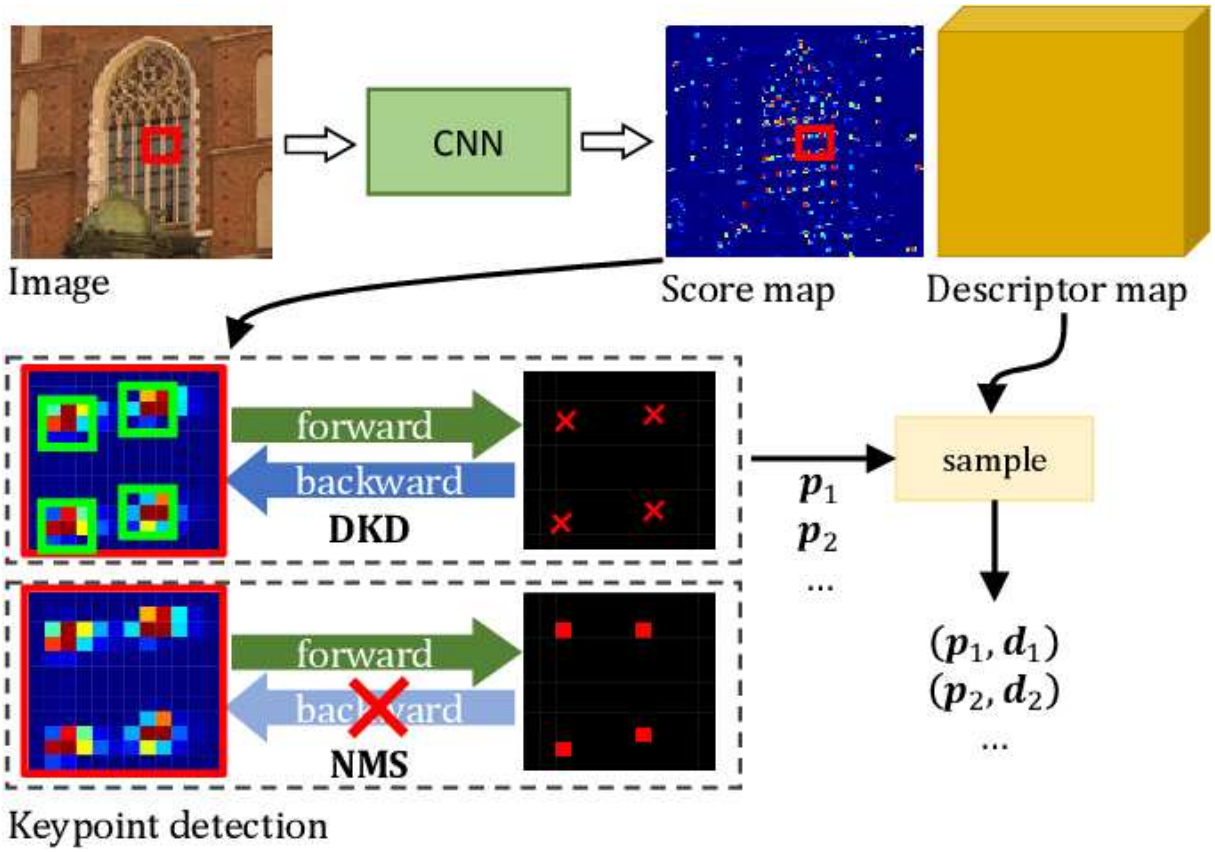


FIGURE 4.7 – Les créateurs du réseau de détection ALIKE ont proposé une détection de points clés différentiable (DKD) pour la détection de points clés et l'extraction de descripteurs basés sur la carte de scores. La détection des points clés sur un patch de carte de scores (la zone de rectangle rouge) est illustrée dans cette figure. Comparé aux méthodes basées sur la suppression non maximale (NMS), le DKD peut exécuter la rétropropagation des gradients et produire des points clés. Ainsi, la position des pixels peut être directement optimisée, afin d'améliorer leur précision.
Source: Zhao et al. (2022)

des scores est réalisée au sein d'une fenêtre de taille $N \times N$, dans la carte de réponse au détecteur et selon la formule suivante :

$$s = \begin{cases} s_{max}, & s = s_{max} \\ 0, & \text{sinon} \end{cases} \tag{4.9}$$

où s_{max} correspond au score maximal dans la fenêtre de taille N^2. Ensuite, un seuil noté th est appliqué sur cette nouvelle carte de score s, afin de filtrer les réponses trop faibles. Pour une NMS classique, c'est à ce moment que sont extraits les points clés, qui sont notés $[u, v]_{NMS}$.

ALIKE utilise ces points clés extraits pour aller plus loin en extrayant les scores locaux notés $[\hat{i}, \hat{j}]_{soft}$ avec la fonction $softargmax$.

Les scores, dans une fenêtre de taille $N \times N$ entourant directement le point clé, sont normalisés par la fonction softmax, selon l'équation :

$$s'(i,j) = softmax(\frac{s(i,j) - s_{max}}{t_{det}}) \tag{4.10}$$

où t_{det} permet de contrôler la finesse de la normalisation et $softmax$:

$$softmax(x) = \frac{e^x}{\sum e^x} \tag{4.11}$$

La carte $s'(i,j)$ indique la probabilité de $[i,j]$ d'être considéré comme un point clé. Ainsi, la position attendue d'un point clé dans une fenêtre locale est calculée par l'équation :

$$[i,j]_{soft} = \sum_{0 \leq i,j \leq N} s'(i,j)[i,j] \tag{4.12}$$

Finalement, la position finale du pixel est obtenue selon :

$$p = [u,v]_{soft} = [u,v]_{NMS} + [i,j]_{soft} \tag{4.13}$$

Cette méthode utilise deux termes pour la détection des points clés d'une image. Le premier terme, $[u,v]_{NMS}$, permet de localiser les points clés et n'est pas différentiable, ce qui signifie que le modèle ne peut pas calculer de gradients à cet endroit. Le deuxième terme, $[i,j]_{soft}$, représente un décalage par rapport à la position des points clés et est couplé aux scores dans une petite zone locale, ce qui le rend différentiable dans cette zone. Ainsi, le modèle est partiellement différentiable et permet au gradient de se propager vers les scores dans cette zone locale grâce au deuxième terme. Optimiser la position des points clés revient donc à optimiser les scores dans cette zone locale. Comme il y a de nombreux points clés, la carte des scores est optimisée de manière parcimonieuse dans chaque zone locale.

4.4.3 Fonction de perte de régularisation locale des scores

L'article [Zhao et al., 2022] décrit une régularisation des scores pour optimiser la détection des points clés par la DKD. Cette régularisation consiste à donner une forme de pic à la localisation du point clé, c'est-à-dire que le score doit être élevé au niveau du point clé et faible autour de ce dernier. Cette régularisation a été mise en œuvre dans d'autres travaux (voir [Dusmanu et al., 2019 ; Revaud, Weinzaepfel et al., 2019]), mais ils ne tenaient pas compte de la distribution spatiale des scores. Pour remédier à cela, l'article propose une nouvelle méthode appelée «*dispersity peak loss*», qui prend en compte la distribution spatiale des scores et résulte en un pic de score à la localisation du point clé et des scores plus faibles autour. Cette fonction utilise la localisation du point clé extraite par la DKD et une parcelle d'image de taille $N \times N$ entourant directement ce point. La distance entre le pixel détecté et chacun des pixels au sein de cette parcelle est calculée et notée de la façon suivante :

$$d(i,j) = \left\{ ||[i,j] - [\hat{i},\hat{j}]_{soft}||_p, 0 \leq i,j < N \right\} \tag{4.14}$$

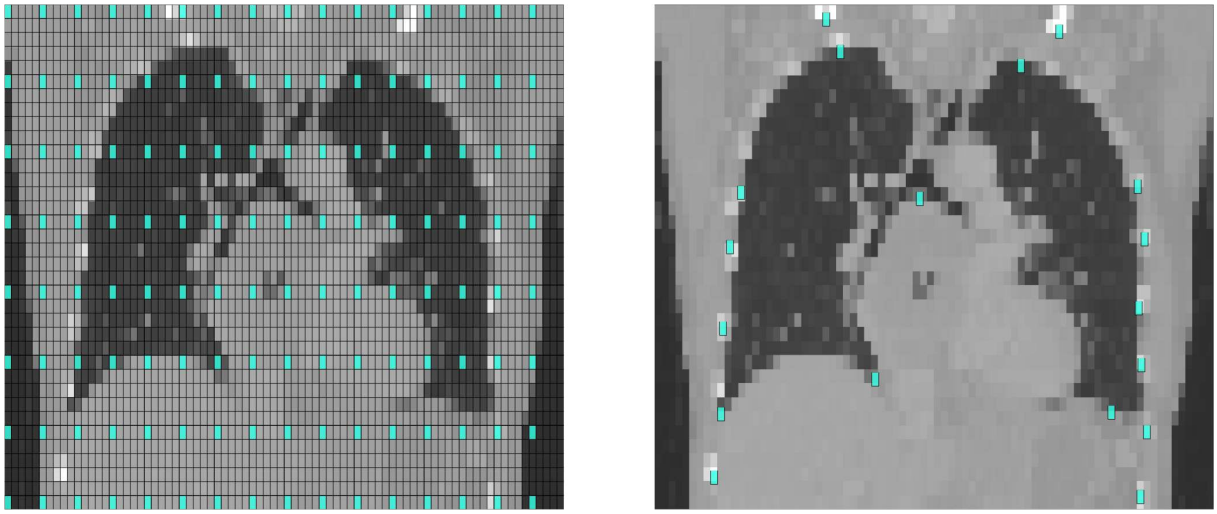


FIGURE 4.8 – Schéma représentant les deux méthodes d'extraction des descriptneurs lors du calcul de la fonction de perte par triplets. Avec à gauche les descripteurs prélevés tous les 5 voxels et à droite les descripteurs extraits aux localisations résultantes par la DKD.

Extracteur	Répétablité (2mm)	Correspondance des descripteurs	Distance moyenne entre les points de repère anatomiques
3D-SIFT	0.37	0.16	12.2 mm
3D-SURF	**0.46**	**0.34**	**8.37 mm**
D&DL3D	0.45	0.21	11.3 mm

TABLEAU 4.4 – Tableau montrant les performances obtenues avec D&DL3D, en comparaison de 3D-SURF et 3D-SIFT.

Pour chaque point détecté, la valeur $softmax$ de descore de détection est multipliée au résultat de la Fonction 4.14, puis ce résultat est moyenné par le nombre de points clés détectés.

4.4.4 Apprentissage du descripteur : perte par triplets

Pour optimiser les descripteurs, nous avons utilisé la fonction de perte par triplets, mais nous avons dû choisir les points à partir desquels extraire les descripteurs pour le calcul de cette fonction. Nous avions deux choix possibles : un échantillonnage régulier (extraire un descripteur à chaque Nième voxel) dans un volume V_1 et extraire les descripteurs correspondants dans le volume V_2, ou ne prendre que les descripteurs correspondant aux points clés détectés par la DKD. La seconde option a été choisie pour empêcher le descripteur d'apprendre des points clés inutiles et pour réduire le temps nécessaire à l'entraînement du réseau. La Figure 4.8 montre les deux méthodes, avec à gauche l'extraction des descripteurs à un certain pas de pixels et à droite les descripteurs extraits uniquement aux localisations données par la DKD.

Extracteur	Répétablité (2mm)	Correspondance des descripteurs	Distance moyenne entre les points de repère anatomiques
3D-SIFT	0.37	0.16	12.2 mm
3D-SURF	0.46	0.34	8.37 mm
R2D3	0.45	0.24	11.3 mm
D&DL3D	0.49	0.23	9.6 mm
SP3D	**0.51**	**0.48**	**7.98 mm**

TABLEAU 4.5 – Tableau montrant les performances obtenues avec SP3D, R2D3 et D&DL3D en comparaison de 3D-SURF et 3D-SIFT.

4.4.5 Inférence et résultats

Concernant l'inférence du réseau, elle se fait de manière similaire à celle de SP3D, c'est-à-dire que pour un volume de patient donné, celui-ci est tout d'abord envoyé au réseau pour en ressortir une carte de réponse au détecteur et au descripteur. Une détection des points clés dans la carte de détection est effectuée par une simple NMS, puis aux localisations des meilleurs points clés (selon la réponse au détecteur et le nombre de points désirés) un descripteur est extrait dans la carte de réponse du descripteur.

Le Tableau 4.4 montre que D&DL3D donne des résultats inférieurs pour toutes les mesures en comparaison à 3D-SURF, mais surpasse 3D-SIFT dans toutes les métriques. On peut voir dans ce tableau que les résultats que nous obtenons avec ce réseau à supervision distante sont bien en dessous de 3D-SURF. Nous avons ici le même problème qu'avec R2D3, concernant la taille des volumes en entrée que nous pouvons fournir au réseau.

4.5 Conclusions

Dans cette dernière section, nous avons mis en place 3 réseaux de neurones permettant l'extraction des points clés de bout en bout, c'est-à-dire à la fois la détection et la description des points au sein d'images médicales volumiques.

Le Tableau 4.5 montre que SP3D donne de meilleurs résultats pour toutes les mesures en comparaison à 3D-SIFT, 3D-SURF, D&DL3D et R2D3.

Les résultats obtenus ne confirment pas notre première hypothèse, selon laquelle un réseau neuronal à supervision distante (c'est-à-dire que l'on ne force pas à apprendre à détecter des points clés spécifiques) serait en mesure d'apprendre à détecter de meilleurs points clés, à l'instar de R2D2 qui n'utilise pas de vérité terrain, excepté pour les transformations qui sont connues.

On peut voir dans les résultats que nous obtenons, que les résultats obtenus par SP3D sont bien meilleurs qu'avec les autres réseaux. Ces résultats s'expliquent probablement par le fait que D&DL3D et R2D3 nécessitent un apprentissage sur des images entières en 2D, ce qui n'est pas encore permis en 3D. Nous avons effectué quelques tests en compressant les

volumes à une taille admissible pour un entraînement sur les volumes entiers. Les résultats semblaient être meilleurs sous cette forme.

Une autre explication pourrait venir du fait que nos entraînements sont réalisés uniquement en intrapatient. Un entraînement en interpatient, avec la connaissance exacte de chaque voxel entre les patients, pourrait améliorer significativement nos entraînements.

Les performances en deçà des attentes pour R2D3 et D&DL3D peuvent probablement s'expliquer par le fait que nous n'avons pas pu entraîner le réseau sur des images entières à haute résolution. Pour remédier à ce problème, une idée serait d'utiliser un réseau nécessitant moins de mémoire, comme un réseau U-Net, ou d'utiliser des noyaux épars.

5

Perspectives et conclusion

Sommaire

5.1 Bilan

Notre étude est arrivée à son terme et il est temps de faire le bilan de nos travaux. Nous avons cherché à combiner les algorithmes existants de détection de points d'intérêt et d'extraction de descripteurs afin de créer un nouvel extracteur capable de réaliser ces deux tâches de manière continue et en 3D. Cette approche nous a permis de détecter efficacement des points d'intérêt dans des volumes 3D et de calculer les descripteurs correspondants, démontrant ainsi la faisabilité d'une extraction d'informations à partir d'images volumiques.

D'un autre côté, bien que le manque initial de données médicales ait été un obstacle majeur à l'utilisation de l'apprentissage par réseau pour extraire des points, nous avons élaboré une nouvelle méthode qui permet de simuler des volumes médicaux pour créer une base de données cohérente à des fins d'apprentissage.

En fin de compte, les résultats ont démontré les avantages de cette méthode, tant en termes de répétabilité des points clés détectés que de recherche de correspondances entre les descripteurs intrapatients. Le calcul de la distance moyenne entre les points de repère anatomiques a également mis en évidence les résultats obtenus par cette méthode en interpatients.

5.2 Perspectives

Selon Kurt Gödel, tout travail de recherche est nécessairement incomplet, car il y aura toujours des questions auxquelles il n'est pas possible de répondre. C'est la raison pour laquelle nous allons aborder dans cette partie quelques pistes qui restent à explorer afin de poursuivre notre travail de recherche.

5.2.1 Amélioration de l'algorithme basé apprentissage distant

La première piste à explorer étant l'amélioration de l'apprentissage du réseau R2D3, qui est l'extension 3D du réseau R2D2, ainsi que le réseau que nous avons mis en place qui optimise directement la répétabilité entre les points clés détectés. Bien que les résultats en 2D aient semblé très prometteurs pour notre application en 3D, l'utilisation d'un entraînement supervisé tel que Superpoint a montré des performances supérieures.

5.2.2 Mise en place d'une évaluation interpatient

Une autre piste intéressante consisterait à mettre en place un test permettant d'analyser la répétabilité des points clés et la discriminabilité de leurs descripteurs entre différents patients. Actuellement, ces deux métriques sont calculées en intrapatient en appliquant une transformation aléatoire, puis en calculant les métriques à partir de ce «*nouveau*» patient et du patient original.

Il serait possible d'envisager d'utiliser un recalage dense, très précis, pour calculer les correspondances entre chaque voxel des images, puis d'utiliser ces transformations afin de calculer les différentes métriques.

5.2.3 Extension des réseaux à la multimodalité

Nous souhaiterions également rendre l'extraction de points clés multimodale, c'est-à-dire pouvoir appliquer un réseau aussi bien sur des images TDM que sur des images IRM. Malheureusement, il n'existe actuellement aucune base de données efficace pour entraîner un réseau de neurones sur différentes modalités. Cependant, il serait envisageable d'utiliser des réseaux générateurs d'images, comme que des réseaux antagonistes génératifs (GAN), pour générer des images IRM à partir d'images tomodensitométriques, puis utiliser ces images pour effectuer un entraînement non supervisé.

5.3 Conclusion

La thèse avait pour objectif d'examiner la faisabilité de mettre au point une méthode pour détecter et décrire les points clés dans des images volumiques médicales. Dans un premier temps, nous avons élaboré des méthodes de description de points clés en utilisant les points détectés par l'algorithme 3D-SURF. Dans un second temps, pour minimiser la dépendance de 3D-SURF et étant donné l'efficacité prouvée des méthodes de détection et de description en 2D, nous avons élargi puis développé des méthodes d'extraction de bout en bout des points clés en 3D.

Nous avons apporté plusieurs contributions dans notre travail visant à élaborer un protocole d'apprentissage pour la détection et la description de points clés. La première contribution, qui est double, consiste en la mise en place d'une méthode de génération artificielle de nouveaux volumes de patients, que nous avons utilisée pour développer la méthode de description de points par l'algorithme en triplets. La seconde contribution permet l'extraction des points clés en un seul passage dans un réseau convolutionnel à partir d'une image médicale volumique. Nous avons également ajouté une dernière contribution, qui consiste à replacer les points de repère anatomiques pour un calcul plus fiable de nos métriques.

Nous avons mesuré les performances de nos différents réseaux à l'aide de métriques de référence. Tout d'abord, les points de repère anatomiques initialement placés par des experts ont été utilisés pour évaluer la précision des points de repère à l'aide de l'algorithme de recalage FROG. Ensuite, la répétabilité des points clés détectés a été évaluée pour déterminer leur

fiabilité et leur cohérence. Enfin, une métrique de correspondance des descripteurs a été utilisée pour évaluer la justesse de l'appariement des points clés.

Cette thèse ouvre la voie à de nouveaux développements pour la création de réseaux de détection et de description de points clés en 3D

Références

Abdi, H. & Williams, L. J. (2010). Principal component analysis. *Wiley Interdisciplinary Reviews : Computational Statistics*, *2*, 433-459.

Abien Fred, A. (2018). Deep learning using rectified linear units (relu). *CoRR*, *abs/1803.08375*. http://arxiv.org/abs/1803.08375

Agier, R. (2017). *Recalage de groupes d'images médicales 3D par extraction de points d'intérêt* (Theses 2017LYSEI093). Université de Lyon. https://theses.hal.science/tel-02004354

Agier, R., Valette, S., Kéchichian, R., Fanton, L. & Prost, R. (2020). Hubless keypoint-based 3d deformable groupwise registration. *Medical image analysis*, *59*, 101564.

Agier, R., Valette, S., Fanton, L., Croisille, P. & Prost, R. (2016). Hubless 3d medical image bundle registration. *"VISAPP 2016 11th Joint Conference"*.

Altwaijry, H., Veit, A., Belongie, S. J. & Tech, C. (2016). Learning to detect and match keypoints with deep architectures. *BMVC*.

Balntas, V., Lenc, K., Vedaldi, A. & Mikolajczyk, K. (2017). Hpatches : A benchmark and evaluation of handcrafted and learned local descriptors. *CoRR*, *abs/1704.05939*.

Balntas, V., Riba, E., Ponsa, D. & Mikolajczyk, K. (2016). Learning local feature descriptors with triplets and shallow convolutional neural networks. *Bmvc*, 3.

Bay, H., Tuytelaars, T. & Van Gool, L. (2006). Surf : speeded up robust features. *European conference on computer vision*, 404-417.

Bentley, J. L. (1975). Multidimensional binary search trees used for associative searching. *Commun. ACM*, *18*(9), 509-517.

Bergstra, J. & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, *13*(10), 281-305. http://jmlr.org/papers/v13/bergstra12a.html

Bhattacharjee, R., Heitz, F., Noblet, V., Sharma, S. & Sharma, N. (2021). Evaluation of a learning-based deformable registration method on abdominal ct images. *IRBM*, *42*(2), 94-105.

Blendowski, M. & Heinrich, M. (2018). 3d-cnns for deep binary descriptor learning in medical volume data.

Borgefors, G. (1986). Distance transformations in digital images. *Computer Vision, Graphics, and Image Processing*, *34*(3), 344-371.

Bralet, A., Kéchichian, R. & Valette, S. (2021). Local surf-based keypoint transfer segmentation. *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, 1390-1393.

Bromley, J., Guyon, I., LeCun, Y., Säckinger, E. & Shah, R. (1993). Signature verification using a "siamese" time delay neural network. In J. Cowan, G. Tesauro & J. Alspector (Éd.), *Advances in neural information processing systems*. Morgan-Kaufmann. https://proceedings.neurips.cc/paper/1993/file/288cc0ff022877bd3df94bc9360b9c5d-Paper.pdf

Brown, L. G. (1992). A survey of image registration techniques. *ACM Comput. Surv.*, *24*(4), 325-376.

Brown, M., Hua, G. & Winder, S. (2010). Discriminative learning of local image descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *33*(1), 43-57.

Cakir, F., He, K., Xia, X., Kulis, B. & Sclaroff, S. (2019). Deep metric learning to rank. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Calonder, M., Lepetit, V., Ozuysal, M., Trzcinski, T., Strecha, C. & Fua, P. (2012). Brief : computing a local binary descriptor very fast. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *34*(7), 1281-1298.

Cootes, T., Twining, C., Petrovic, V., Babalola, K. & Taylor, C. (2010). Computing accurate correspondences across groups of images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *32*, 1994-2005.

Cortes, C. & Vapnik, V. (1995). Support-vector networks. *Machine learning*, *20*(3), 273-297.

Cuiyin, L., Jishang, X. & Feng, W. (2021). A review of keypoints' detection and feature description in image registration.

de Boor, C. (2001). *A practical guide to splines*.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K. & Fei-Fei, L. (2009). Imagenet : a large-scale hierarchical image database. *2009 IEEE conference on computer vision and pattern recognition*, 248-255.

Deng, L. (2012). The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, *29*(6), 141-142.

DeTone, D., Malisiewicz, T. & Rabinovich, A. (2017). Superpoint : self-supervised interest point detection and description. *CoRR*, *abs/1712.07629*.

Ding-Xuan, Z. (2020). Universality of deep convolutional neural networks. *Applied and Computational Harmonic Analysis*, *48*(2), 787-794.

Dusmanu, M., Rocco, I., Pajdla, T., Pollefeys, M., Sivic, J., Torii, A. & Sattler, T. (2019). D2-net : A trainable CNN for joint detection and description of local features. *CoRR*, *abs/1905.03561*.

Elyan, E., Vuttipittayamongkol, P., Johnston, P., Martin, K., McPherson, K., Moreno-Garcia, C., Jayne, C. & Sarker, M. M. K. (2022). Computer vision and machine learning for medical image analysis : recent advances, challenges, and way forward. *Artificial Intelligence Surgery*, *2*.

Fernández Alcantarilla, P., Bartoli, A. & Davison, A. (2012). Kaze features. *ECCV*.

Fischler, M. & Bolles, R. (1981). Random sample consensus : a paradigm for model fitting with applications to image analysis and automated cartography. *Association for Computing Machinery*, *24*(6), 381-395.

Flitton, G., Breckon, T. & Megherbi Bouallagu, N. (2010). Object recognition using 3d sift in complex ct volumes [doi :10.5244/C.24.11]. *Proceedings of the British Machine Vision Conference*, 11.1-11.12.

Fu, Y., Lei, Y., Wang, T., Curran, W. J., Liu, T. & Yang, X. (2020). Deep learning in medical image registration : a review. *Physics in Medicine and Biology*, *65*(20), 20TR01.

Fukushima, K. (1980). Neocognitron : A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, *36*, 193-202.

Gallagher, D., Chung, S. & Akram, M. (2013). Body composition. In B. Caballero (Éd.), *Encyclopedia of human nutrition (third edition)* (Third Edition, p. 191-199). Academic Press.

Gordo, A., Almazan, J., Revaud, J. & Larlus, D. (2017). End-to-end learning of deep visual representations for image retrieval.

Grand-Brochier, M. (2011). *Descripteurs 2d et 2d+t de points d'intérêt pour des appariements robustes* (Theses). Université Blaise Pascal - Clermont-Ferrand II. https://theses.hal.science/tel-00697021

Guy, C. & ffytche, D. (2000). *An introduction to the principles of medical imaging*. PUBLISHED BY IMPERIAL COLLEGE PRESS ; DISTRIBUTED BY WORLD SCIENTIFIC PUBLISHING CO.

Harris, C. G. & Stephens, M. J. (1988). A combined corner and edge detector. *Alvey Vision Conference*.

Hartley, R. & Zisserman, A. (2004a). *Multiple view geometry in computer vision* (2e éd.).

Hartley, R. & Zisserman, A. (2004b). *Multiple view geometry in computer vision* (2e éd.). Cambridge University Press.

Hassaballah, M., Alshazly, H. & Ali, A. (2019). Analysis and evaluation of keypoint descriptors for image matching.

He, K., Lu, Y. & Sclaroff, S. (2018). Local descriptors optimized for average precision.

Horaud, R., Veillon, F. & Skordas, T. (1990). Finding geometric and relational structures in an image, 374-384.

Hosang, J., Benenson, R. & Schiele, B. (2016). A convnet for non-maximum suppression.

Hosang, J., Benenson, R. & Schiele, B. (2017). Learning non-maximum suppression.

Huang, X., Mei, G. & Zhang, J. (2020). Feature-metric registration : a fast semi-supervised approach for robust point cloud registration without correspondences.

Jaiswal, A., Singh, S., Wu, Y., Natarajan, P. & Natarajan, P. (2021). Keypoints-aware object detection. In L. Bertinetto, J. F. Henriques, S. Albanie, M. Paganini & G. Varol (Éd.), *Neurips 2020 workshop on pre-registration in machine learning* (p. 62-72). PMLR.

Kéchichian, R., Valette, S., Sdika, M. & Desvignes, M. (2014). Automatic 3d multiorgan segmentation via clustering and graph cut using spatial relations and hierarchically-registered atlases. *Medical Computer Vision : Algorithms for Big Data*, 201-209.

Kingma, D. P. & Ba, J. (2017). Adam : a method for stochastic optimization.

Krenn, M., Grünberg, K., Jimenez-del-Toro, O., Jakab, A., Salas Fernandez, T., Winterstein, M., Weber, M.-A. & Langs, G. (2017). Datasets created in visceral. *Cloud-based benchmarking of medical image analysis* (p. 69-84). Springer International Publishing.

Krig, S. (2014). Interest point detector and feature descriptor survey. *Computer vision metrics : survey, taxonomy, and analysis* (p. 217-282). Apress.

Krizhevsky, A., Sutskever, I. & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In F. Pereira, C. Burges, L. Bottou & K. Weinberger (Éd.), *Advances in neural information processing systems*. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

Langs, G., Hanbury, A., Menze, B. & Müller, H. (2013). Visceral : towards large data in medical imaging — challenges and directions. *Medical Content-Based Retrieval for Clinical Decision Support*, 92-98".

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation*, *1*(4), 541-551.

Leutenegger, S., Chli, M. & Siegwart, R. Y. (2011). Brisk : binary robust invariant scalable keypoints. *2011 International Conference on Computer Vision*, 2548-2555.

Li, S. Z. & Jain, A. (Éd.). (2009). Hamming distance. In *Encyclopedia of biometrics* (p. 668-668). Springer US.

Lin, T., Maire, M., Belongie, S. J., Bourdev, L. D., Girshick, R. B., Hays, J., Perona, P., Ramanan, D., Doll'a r, P. & Zitnick, C. L. (2014). Microsoft COCO : common objects in context. *CoRR*, *abs/1405.0312*.

Liu, T.-Y. (2009). Learning to rank for information retrieval. *Found. Trends Inf. Retr.*, *3*(3), 225-331.

Loiseau–witon, N., Kéchichian, R., Valette, S. & Bartoli, A. (2021). Description de points clés par apprentissage dans des images médicales 3d. *ORASIS*.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, *60*, 91-110.

Maintz, J. & Viergever, M. A. (1998). A survey of medical image registration. *Medical Image Analysis*, *2*(1), 1-36.

Masone, C. & Caputo, B. (2021). A survey on deep visual place recognition. *IEEE Access*, *9*, 19516-19547.

Mikolajczyk, K. & Schmid, C. (2005). A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*(10), 1615-1630.

Noblet, V. (2006). *Recalage non rigide d'images cérébrales 3d avec contrainte de conservation de la topologie*. https://books.google.fr/books?id=vGhAXwAACAAJ

Odena, A., Dumoulin, V. & Olah, C. (2016). Deconvolution and checkerboard artifacts. *Distill*.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., ... Chintala, S. (2019). Pytorch : an imperative style, high-performance deep learning library. *Advances in neural information processing systems 32* (p. 8024-8035). Curran Associates, Inc. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

Revaud, J., Almazan, J., Sampaio de Rezende, R. & Roberto de Souza, C. (2019). Learning with average precision : training image retrieval with a listwise loss.

Revaud, J., Weinzaepfel, P., Roberto de Souza, C., Pion, N., Csurka, G., Cabon, Y. & Humenberger, M. (2019). R2D2 : repeatable and reliable detector and descriptor. *CoRR*.

Rister, B., Horowitz, M. & Rubin, D. (2017). Volumetric image registration from invariant keypoints. *IEEE Transactions on Image Processing*, *26*, 4900-4910.

Roche, A. (2001). *Recalage d'images médicales par inférence statistique* (thèse de doct.).

Rosenblatt, F. (1958). The perceptron : a probabilistic model for information storage and organization in the brain. *Psychological Review*, *65*, 386-408.

Rublee, E., Rabaud, V., Konolige, K. & Bradski, G. (2011). Orb : an efficient alternative to sift or surf. *2011 International Conference on Computer Vision*, 2564-2571.

Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv preprint arXiv :1609.04747*.

Rumelhart, D. E. & McClelland, J. L. (1987). Learning internal representations by error propagation. *Parallel distributed processing : explorations in the microstructure of cognition : foundations* (p. 318-362).

Sammut, C. & Webb, G. I. (Éd.). (2010). Mean squared error. In *Encyclopedia of machine learning* (p. 653-653). Springer US.

Sattler, T., Weyand, T., Leibe, B. & Kobbelt, L. (2012). Image retrieval for image-based localization revisited. *Proceedings of the British Machine Vision Conference*, 76.1-76.12.

Savinov, N., Seki, A., Ladicky, L., Sattler, T. & Pollefeys, M. (2016). Quad-networks : unsupervised learning to rank for interest point detection.

Scahill, R. I., Frost, C., Jenkins, R., Whitwell, J. L., Rossor, M. N. & Fox, N. C. (2003). A Longitudinal Study of Brain Volume Changes in Normal Aging Using Serial Registered Magnetic Resonance Imaging. *Archives of Neurology*, *60*(7), 989-994.

Schmid, C., Mohr, R. & Bauckhage, C. (1998). Comparing and evaluating interest points. *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, 230-235.

Schmid, C., Mohr, R. & Bauckhage, C. (2000). Evaluation of interest point detectors. *International Journal of Computer Vision)*, *37 (2)*, 151-172.

Schroff, F., Kalenichenko, D. & Philbin, J. (2015). Facenet : a unified embedding for face recognition and clustering. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 815-823.

Scott, D. (2015). *Multivariate density estimation : theory, practice, and visualization*.

Scovanner, P., Ali, S. & Shah, M. (2007). A 3-dimensional sift descriptor and its application to action recognition. *Proceedings of the 15th ACM International Conference on Multimedia*, 357-360. https://doi.org/10.1145/1291233.1291311

Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A. P., Bishop, R., Rueckert, D. & Wang, Z. (2016). Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network.

Shiv Ram, D., Satish Kumar, S. & Bidyut Baran, C. (2021). A comprehensive survey and performance analysis of activation functions in deep learning. *CoRR*, *abs/2109.14545*. https://arxiv.org/abs/2109.14545

Simonyan, K. & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition.

Sipiran, I. & Bustos, B. (2011). Harris 3d : a robust extension of the harris operator for interest point detection on 3d meshes. *The Visual Computer*, *27*, 963-976.

Sotiras, A., Davatzikos, C. & Paragios, N. (2013). Deformable medical image registration : a survey. *IEEE Transactions on Medical Imaging*, *32*(7), 1153-1190.

Streiff, D., Bernreiter, L., Tschopp, F., Fehr, M. & Siegwart, R. (2021). 3d3l : deep learned 3d keypoint detection and description for lidars.

Swain, M. J. & Ballard, D. H. (1991). Color indexing. *International Journal of Computer Vision*, *7*, 11-32.

Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C. & Liu, C. (2018). A survey on deep transfer learning.

Tian, Y., Fan, B. & Wu, F. (2017). L2-net : deep learning of discriminative patch descriptor in euclidean space. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6128-6136.

Tola, E., Lepetit, V. & Fua, P. (2008). A fast local descriptor for dense matching. *Proc. CVPR*.

van der Maaten, L. & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, *9*, 2579-2605. http://www.jmlr.org/papers/v9/vandermaaten08a.html

Verdie, Y., Yi, K. M., Fua, P. & Lepetit, V. (2014). TILDE : A temporally invariant learned detector. *CoRR*, *abs/1411.4568*.

Vijay Kumar B, G., Carneiro, G. & Reid, I. (2015). Learning local image descriptors with deep siamese and triplet convolutional networks by minimising global loss functions.

Wachinger, C., Toews, M., Langs, G., Wells, W. & Golland, P. (2015). Keypoint transfer segmentation. *24*.

Wasserthal, J., Meyer, M., Breit, H.-C., Cyriac, J., Yang, S. & Segeroth, M. (2022). Totalsegmentator : robust segmentation of 104 anatomical structures in ct images.

Winder, S., Hua, G. & Brown, M. (2009). Picking the best daisy. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 178-185.

Yi, K. M., Trulls, E., Lepetit, V. & Fua, P. (2016). Lift : learned invariant feature transform. *CoRR*, *abs/1603.09114*.

Zagorchev, L. & Goshtasby, A. (2006). A comparative study of transformation functions for nonrigid image registration. *IEEE Transactions on Image Processing*, *15*(3), 529-538.

Zagoruyko, S. & Komodakis, N. (2017). Wide residual networks.

Zhang, Y., Ozay, M., Li, S. & Okatani, T. (2017). Truncating wide networks using binary tree architectures.

Zhao, X., Wu, X., Miao, J., Chen, W., Chen, P. C. Y. & Li, Z. (2022). Alike : accurate and lightweight keypoint detection and descriptor extraction.

Liste des Travaux

Conférences internationales

Loiseau–witon, N., Kéchichian, R., Valette, S. & Bartoli, A. (2021b). Learning 3D medical image keypoint descriptors with the triplet loss. *International Journal of Computer Assisted Radiology and Surgery*.

Conférences nationale

- Loiseau–witon, N., Kéchichian, R., Valette, S. & Bartoli, A. (2021a). Description de points clés par apprentissage dans des images médicales 3D. *ORASIS*.

Journaux internationnaux

- Loiseau–witon, N., Kéchichian, R., Valette, S. & Bartoli, A. (2021c). Learning 3d medical image patch descriptors with the triplet loss. *IPCAI*.

FOLIO ADMINISTRATIF

THESE DE L'UNIVERSITE DE LYON OPEREE AU SEIN DE L'INSA LYON

NOM : Loiseau—Witon (Nom de naissance : Loiseau) DATE de SOUTENANCE : 10/11/3024

Prénoms : Nicolas

TITRE : Détection et description de points clés par apprentissage en vue d'un recalage à grande échelle

NATURE : Doctorat Numéro d'ordre : 2023ISAL0101

Ecole doctorale : Électronique, Électrotechnique, Automatique

Spécialité : Traitement du Signal et de l'Image

RESUME :
Les hôpitaux génèrent de plus en plus d'images médicales en 3D. Ces volumes nécessitent un recalage automatique, en vue d'être analyser de manière systématique et à grande échelle. Les points clés sont utilisés pour réduire la durée et la mémoire nécessaire à ce recalage et peuvent être détectés et décrits à l'aide de différentes méthodes classiques, mais également à l'aide de réseaux neuronaux, comme cela a été démontré de nombreuses fois en 2D.

Cette thèse présente les résultats et les discussions sur les méthodes de détection et de description de points clés à l'aide de réseaux neuronaux 3D. Deux types de réseaux ont été étudiés pour détecter et/ou décrire des points caractéristiques dans des images médicales 3D. Les premiers réseaux étudiés permettent de décrire les zones entourant directement les points clés, tandis que les seconds effectuent les deux étapes de détection et de description des points clés en une seule fois.

MOTS-CLÉS : Détection, Description, Point clé, Image médicale 3D, Recalage.

Laboratoire (s) de recherche : CREATIS

Directeur de thèse: Sébastien VALETTE

Co-encadrant de thèse : Razmig Kéchichian

Composition du jury :

Sébastien Valette	(Directeur de thèse),
Razmig Kéchichian	(co-encadrant),
Elsa Angelini	(Examinatrice),
Marc Antonini	(Examinateur),
Franck Hétroy-Wheeler	(Rapporteur),
Olivier Aubreton	(Rapporteur)