Description de points clés par apprentissage dans des images médicales 3D

Nicolas Loiseau–Witon, Razmig Kechichian, Sébastien Valette, Adrien Bartoli

Description de points clés par apprentissage dans des images médicales 3D

Nicolas Loiseau–Witon [1,2]
Sébastien Valette [1]
Razmig Kechichian [1]
Adrien Bartoli [2]

[1] Univ Lyon, INSA-Lyon, Universite Claude Bernard Lyon 1, UJM-Saint Etienne, CNRS, Inserm, CREATIS UMR 5220, U1206, F-69100, LYON, France
[2] Institut Pascal, UMR 6602 CNRS/UCA/CHU, Clermont-Ferrand, France

[1] nom@creatis.insa-lyon.com
[2] Adrien.Bartoli@gmail.com

Résumé

Nous présentons une nouvelle approche pour apprendre des descripteurs de points clés en 3D, que nous appliquons aux images médicales scanners médicaux corps entier. Il a été démontré que les descripteurs de points clés basés sur les Réseaux de Neurones Convolutifs (RNC) donnent de meilleurs résultats que les descripteurs faits à la main, pour les images 2D. L'adaptation aux images 3D telles que les images tomodensitométriques n'est cependant pas simple, essentiellement en raison du manque de données d'entraînement labellisées.
Nous proposons de générer des données d'entraînement semi-synthétiques. L'idée principale est d'estimer d'abord la densité des transformations locales entre les patients à partir d'un petit nombre d'images tomodensitométriques dont les correspondances entre des points de repère anatomique, définis par des experts, sont connues. Nous échantillonnons ensuite un grand nombre de transformations à partir de cette densité et transformons des volumes labellisés, pour lesquels nous pouvons ensuite former des correspondances de points clés exactes en utilisant une correspondance guidée par la transformation. Notre fonction de description est un RNC à deux étages, que nous formons en utilisant la perte par triplets inspirée par l'apprentissage de descripteurs 2D, avec une extraction de triplets en ligne.
Nos résultats expérimentaux montrent que notre descripteur appris surpasse le descripteur 3D-SURF créé à la main dans une évaluation sur les données semi-synthétiques ainsi qu'en recalage d'images 3D réelles, avec un temps d'exécution similaire.

Mots Clef

Imagerie médicale - Points clés - Perte par triplet - Réseau de neurones convolutionnel

Abstract

We present a new approach to learn 3D keypoint descriptors which we apply to full-body medical CT scans. It has been demonstrated that keypoint descriptors based on Convolutional Neural Networks (CNNs) achieve better results than hand-made descriptors for 2D images. The adaptation to 3D images such as CT scans is however not straightforward, essentially because of the lack of labelled training data.
We propose to generate semi-synthetic training data. The key idea is to first estimate the density of local inter-patient transformations from a small number of CT scans with known keypoint correspondences. We then sample a large number of transformations from this density and warp unlabelled CT scans, for which we can subsequently retrieve groundtruth keypoint correspondences using transformation-guided matching. Our description function is a two-stage CNN, which we train by using the triplet loss inspired by 2D descriptor learning, with online triplet mining.
Our experimental results show that our learned descriptor outperforms the hand-crafted 3D-SURF descriptor on a synthetic benchmark and on a real registration test, with similar runtime. We have proposed a new learning approach for 3D descriptors. Preliminary experiments illustrate the advantages of learned 3D descriptors over hand-crafted descriptors.

Keywords

Medical imaging - Keypoints - Triplet loss - Convolution NeuralNetwork

1 Introduction

L'anatomie computationnelle se concentre sur l'analyse de la variabilité anatomique humaine. Les applications typiques sont : la découverte des différences entre les sujets sains et malades et la classification des anomalies. Un outil fondamental de l'anatomie computationnelle, qui constitue l'objet central de cet article, est le calcul des correspondances de points dans des volumes (images 3D) tels que les volumes de tomographie (CT), par ordinateur, pour plusieurs sujets. Plus précisément, nous considérons des

points clés détectés automatiquement et leurs descripteurs locaux, calculés à partir de l'image ou de la parcelle de volume entourant chaque point clé. Ces descripteurs sont essentiels et doivent être discriminants et répétables [5, 16]. La détection et la description de points clés sont deux problèmes à part entière, et nous ne traiterons ici que celui de la description. Nous utilisons une implémentation de SURF en 3D pour nos étapes de détection de points clés dans les images.

Les descripteurs appris basés sur les réseaux de neurones convolutionnels (RNC) ont récemment montré un grand succès pour les images 2D [10, 21, 4]. Cependant, alors que les descripteurs classiques d'images 2D ont été étendus aux volumes [1, 18], les approches récentes basées sur l'apprentissage ont été limitées à la détection et à la description 2D. À notre connaissance, l'extension aux descripteurs 3D n'a été proposée que dans [6], dans le contexte de la recherche d'images. Cette approche n'est pas directement comparable à la nôtre en raison d'objectifs de formation différents et nous aurions besoin d'accéder au code source pour une comparaison plus approfondie.

Nous proposons une méthodologie pour apprendre des descripteurs de points clés 3D à partir de données volumétriques adaptées au recalage d'images. La principale difficulté consiste à définir une approche d'apprentissage solide, combinant un ensemble de données d'apprentissage et une fonction de perte. En résumé, nous proposons de générer des données semi-synthétiques en transformant des volumes réels et d'utiliser une fonction de perte par triplet inspirée de l'apprentissage de descripteurs 2D. Nos résultats expérimentaux montrent que notre descripteur appris surpasse le descripteur 3D-SURF [1], une extension 3D de SURF, avec un temps d'exécution similaire.

2 Travaux antérieurs

Description des méthodes classiques

La conception de descripteurs locaux a connu une remarquable évolution au cours des dernières décennies, et a fourni un grand nombre de descripteurs de caractéristiques, avec des descripteurs tels que SIFT [16], SURF [5], ORB [8] et BRISK [15]. Ces descripteurs ont été développés pour répondre à de multiples objectifs, tels que l'optimisation de la précision des correspondances ou la vitesse d'extraction, et ils ont tous montré leur efficacité dans des problèmes variés de vision par ordinateur. Depuis quelques années, avec l'expansion des réseaux de neurones dans un grand nombre d'applications, la littérature en description de points clés à aussi vu apparaître des détecteurs et descripteurs de points clés par apprentissage [4, 9]. Nous étendons ici l'apprentissage de descripteurs de points clés 2D à des images médicales 3D.

Description de caractéristiques par apprentissage profond

Récemment, la littérature en vision par ordinateur a vu apparaître un certain nombre de résultats de pointe dans le domaine de la description de points clés. Dans [19, 11], des architectures profondes ont été proposées pour l'apprentissage de ces descripteurs, et l'approche siamoise [13] est devenue une base pour de tels réseaux. Comme son nom l'indique, cette architecture est composée de deux branches similaires (cf Figure 1), et chaque branche aura une copie du réseau en parallèle. Ces sous-réseaux partagent les mêmes poids, lors de l'entraînement. Ainsi le réseau peut apprendre à décrire des parcelles d'images extraites autour de points clés détectés, avec comme but final de rapprocher les descripteurs de points clés similaires à être proche dans l'espace de description, et à écarter les points clés ne correspondants pas entre eux. Par la suite, la fonction de coût par triplet est apparue, sur le même principe que les réseaux siamois, avec une branche supplémentaire afin de bien distinguer les points clés détectés similaires de ceux qui ne correspondent pas.
En dépit de la puissance de l'apprentissage profond avec ce type d'architecture, le manque de base complète d'apprentissage ne permet pas à ces RNC d'acquérir un apprentissage correct. Dans cet article, nous visons à corriger ce problème de manque de base de données annotées, en créant une pseudo vérité terrain semi-synthétique. Nous nous baserons sur ce jeu de données afin d'entraîner notre RNC basé sur une fonction de perte par triplets. Nous montrerons que notre approche obtient de bons résultats sur les bases de données finales.

3 Méthode

Notre objectif principal est donc d'entraîner un modèle capable de décrire une parcelle d'image autour d'un point clé détecté. Pour ce faire, nous allons utiliser la fonction de coût par triplet [20], qui nécessite la connaissance des correspondances exactes entre les différents points clés détectés. Notre premier objectif est donc de créer un jeu de données de référence définissant les correspondances exactes des points clés entre plusieurs volumes. En 2D, ces correspondances peuvent être établies en utilisant l'approche *Structure-from-Motion* [3]. Dans les images médicales 3D, les points clés peuvent être définis comme des repères anatomiques placés par des experts médicaux. Cependant, aucun ensemble de données annotées de cette ampleur n'est disponible publiquement. Nous proposons donc de créer un jeu de données de référence semi-synthétique en transformant des volumes réels.

3.1 Construction de l'ensemble de données semi-synthétique

Nous utilisons deux sous-ensembles du jeu de données Visceral [14]. Le premier sous-ensemble, nommé *Gold*, contient 20 volumes CT, chacun annoté d'environ 40 points

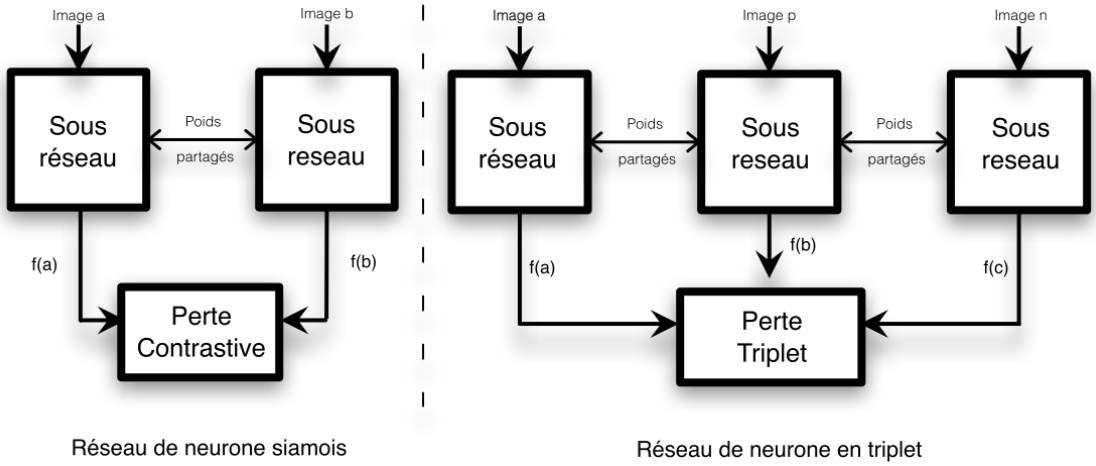

FIGURE 1 – Schéma représentant les deux architectures de réseau. L'architecture siamoise à gauche avec la fonction de coût contrastive et l'architecture en trois branches, avec une fonction de coût par triplet.

de repère anatomiques, placés par des spécialistes. Le second sous-ensemble, appelé *Silver*, contient 60 volumes CT sans points de repère.

Afin de générer de nouveaux volumes, nous estimons la densité de probabilité des transformations inter-volumes possibles, et tirons une transformation affine à partir de cette densité, pour déformer les volumes de CT et définir les correspondances des points clés. Nous calculons d'abord les transformations affines locales inter-volumes au sens des moindres carrés. Pour chaque point de repère dans chaque volume du sous-ensemble *Gold*, nous utilisons ce point de repère, ses trois points de repère les plus proches et les quatre points de repère correspondants dans un autre volume pour estimer une transformation locale. Lorsque le repère et ses trois voisins sont presque colinéaires (par exemple, les repères vertébraux), le problème des moindres carrés est mal conditionné et nous rejetons donc la transformation. On obtient ainsi au maximum $Lk(k-1)/2$ matrices de transformation affine de taille 4×4, où $L = 40$ est le nombre de repères anatomiques et $k = 20$ le nombre de volumes contenant ces points de repères. Le test de Pearson montre que les éléments de ces matrices sont indépendants, ce qui nous permet d'échantillonner chaque élément de cette distribution indépendamment. Nous appliquons la méthode de l'estimation par noyau à ces matrices pour estimer la densité des transformations inter-volumes. Le test t de Student montre qu'un noyau gaussien est un bon ajustement pour cette estimaton. Nous estimons la largeur de bande du noyau via les règles de Scott. Pour générer notre ensemble de données semi-synthétiques, nous tirons les transformations à partir de cette densité et les appliquons aux volumes du sous-ensemble *Silver*. Plus précisément, pour une transformation échantillonnée t et un volume *Silver* V_i, nous obtenons le volume V_i^t. Nous détectons les points clés dans V_i et V_i^t en utilisant 3D-SURF et obtenons deux ensembles de points clés P_i et P_i^t. Notez que 3D-SURF est à la fois un détecteur et un descripteur. Nous allons également détecter le fond de l'image CT à l'aide d'une méthode par seuillage et ainsi créer un masque pour chaque volume, indiquant si un point se situe dans le corps du patient (1) ou en dehors (0). Nous pouvons ainsi supprimer une partie des points clés détectés, à savoir tous ceux ne se trouvant pas à l'intérieur du patient. Nous appliquons ensuite la transformée inverse t^{-1} aux points clés de V_i^t ayant survécu à l'application du masque . Enfin, nous utilisons un arbre k-d pour construire l'ensemble des points clés correspondants entre les volumes comme l'ensemble des paires : $(p \in P_i, q \in t^{-1}(P_i^t))$, dont la distance p à q est inférieure à 4 mm. Ce seuil a été choisi pour obtenir un grand nombre de correspondances correctes. En utilisant cette méthode couplée à la détection de points de SURF3D, nous obtenons approximativement 20000 points répétés dans chaque patient transformé.

3.2 Entraînement du descripteur avec la fonction de coût par triplet

Nous apprenons un descripteur RNC mettant en correspondance une parcelle 3D de 10^3 voxels entourant un point clé avec un vecteur de descripteur. Des travaux récents en 2D ont montré que l'apprentissage de descripteurs en utilisant des triplets donne de meilleurs résultats qu'en utilisant des paires [12]. L'apprentissage de triplets nécessite la formation de triplets de parcelles d'image $\{a, p, n\}$ où a est nommée ancre, p nommée positive et représentant une parcelle différente, mais de la même classe que a, et n une négative représentant une parcelle d'image d'une classe différente. Dans notre cas, a et p sont deux parcelles d'image extraites autour de points clés correspondants provenant de volumes différents et n est une parcelle d'image extraite autour d'un point clé différent. L'objectif est d'optimiser les paramètres du RNC afin de rapprocher a et p dans l'espace des descripteurs et d'éloigner n de n. Ainsi, la perte par triplet est

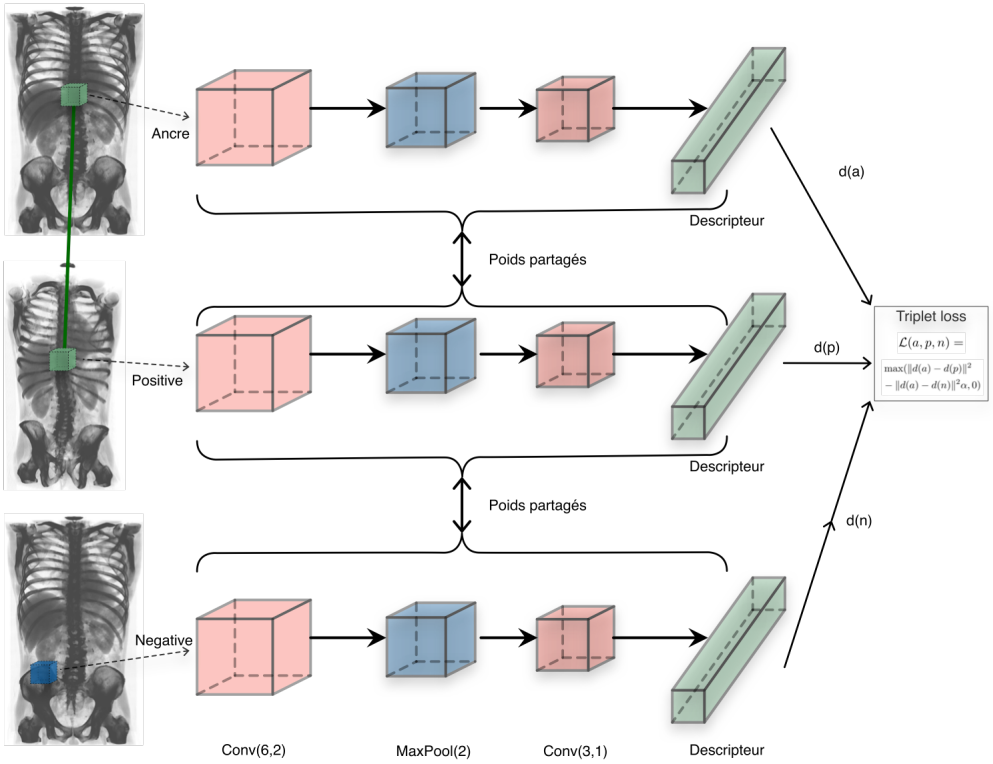

FIGURE 2 – Architecture de notre RNC en triplet. La taille des parcelles d'images est de 10^3 voxels. Le RNC est composé de deux convolutions avec la fonction d'activation $tanh$, un *MaxPooling* (mise en commun maximale en français) et une couche entièrement connectée, nommée *descripteur* sur ce schéma. Le triplet $\{a, p, n\}$ passe à travers le réseau, et le vecteur de description est utilisé pour l'apprentissage avec la fonction de perte par triplet.

définie par l'équation suivante :

$$\mathcal{L}(a,p,n) = \max(\|f(a)-f(p)\|^2 - \|f(a)-f(n)\|^2 + \alpha, 0) \quad (1)$$

où $f(\cdot)$ est le RNC et α le paramètre de marge.
En minimisant cette perte, le réseau amène la distance entre a et p $d(a,p)$ à tendre vers 0, et la distance entre a et n $d(a,n)$ à tendre vers $d(a,p)+\alpha$. Pour un état donné du réseau, les triplets choisis au hasard ne sont pas tous utiles de façon égale pour l'entraînement. Un triplet avec une perte élevée au début peut avoir une perte nulle plus tard au cours de l'entraînement, et donc ne plus rien apporter à l'entraînement. Nous suivons donc une approche d'extraction de triplets en ligne pour un entraînement sélectif sur les triplets les plus utiles. Tout d'abord, nous définissons trois types de triplets [17] représentés sur la Figure 3 :

- *Triplet facile* avec une perte nulle, où $d(a,p)+\alpha < d(a,n)$;
- *Triplet semi-difficile* où $d(a,p) < d(a,n) < d(a,p)+\alpha$;
- *Triplet difficile* avec $d(a,n) < d(a,p)$; les *triplets semi-durs*.

En utilisant cette définition, nous excluons les *triplets faciles* du prochain cycle d'apprentissage en identifiant les *triplets difficiles* et *semi-difficiles* sur la base des valeurs de perte respectives, et nous alimentons un sous-ensemble au réseau : pour chaque mini-lot de taille b, nous formons tous les triplets possibles, et gardons les b triplets avec la valeur de perte la plus élevée.
Notre RNC est peu profond et défini par deux couches de convolution 3D, une couche de *max-pooling* entre les convolutions, et une couche entièrement connectée qui donne le descripteur final. L'architecture du réseau est illustrée dans la Figure 2. Le passage d'une parcelle d'image de taille 10^3 à travers ce RNC nous donne un vecteur de description de la taille désirée. Nous utilisons une taille de vecteur de description de 48 pour une comparaison directe avec 3D-SURF, cette taille a été déterminée après avoir entraîné notre réseau sur des tailles de vecteurs de sortie de la couche pleinement connectée différentes, Tableau 1.

4 Résultats

Fractionnement des données. Nous divisons le groupe de données *Silver* en un sous-ensemble d'entraînement et un autre de validation. Les données d'entraînement consistent en 55 patients avec 10 volumes transformés chacun, suivant notre procédure de génération de données semi-synthétiques. Les données de validation sont constituées de 5 patients avec 10 volumes transformés chacun. Pour les tests, nous utilisons le groupe *Gold* avec 20 sujets et les repères anatomiques associés.

Entraînement. L'optimisation est effectuée par descente de gradient stochastique (DGS), avec une taille de mini-lot de 1000 parcelles, un taux d'apprentissage de 0,1, un momentum de 0,9, une décroissance des poids de 10^{-6} et une marge pour la perte par triplet de 0.2. Nous utilisons également l'exploration de triplets en ligne pour trouver les meilleurs triplets pour l'apprentissage. Notre implémentation basée sur le CPU utilise la bibliothèque PyTorch. L'apprentissage d'une seule époque avec 10^6 triplets prend environ 30 minutes et environ 10 Go de mémoire sur une plateforme Linux 64-bit fonctionnant avec un processeur Intel Xeon 2,6 GHz. Notre modèle est suffisamment léger pour être entraîné sur un CPU ; l'entraînement par GPU n'a pas réduit de manière significative le temps d'entraînement car le processus d'entraînement est principalement lié aux E/S, c'est-à-dire que le chargement des parcelles d'images est le goulot d'étranglement. Notre recherche d'hyperparamètres a été effectuée avec Ray Tune. Nous avons suivi une stratégie de recherche par grille pour les hyperparamètres de taille de mini-lot et de taille de descripteur et une stratégie de recherche aléatoire pour les hyperparamètres de taux d'apprentissage, de marge et de momentum. Le Tableau 1 résume les plages de recherche des hyperparamètres.

Evaluation. Nous évaluons le descripteur en utilisant deux métriques différentes. La première métrique est le taux de faux positifs au point 0.95 du rappel des vrais positifs (FPR95) [7]. Nous calculons le FPR95 en nous basant sur les distances du descripteur de 10^5 paires de points clés sélectionnés aléatoirement avec 50% de paires correspondantes et 50% de paires non correspondantes. Un FPR95 faible indique de bons résultats. La deuxième métrique est la distance moyenne entre les points de repère anatomiques calculée sur les points de repère réels dans les volumes *Gold* après les avoir recalés dans un espace commun à l'aide de l'algorithme de recalage FROG basé sur les points clés [2]. Pour comparaison, nous remplaçons le descripteur 3D-SURF utilisé dans cet algorithme par notre descripteur appris. Une faible distance moyenne entre les points de repère anatomiques indique de bons résultats. Le Tableau 2 compare les résultats entre le descripteur 3D-SURF et notre modèle entraîné avec différentes tailles de descripteurs. Notre descripteur donne les meilleurs résultats avec une taille de 48, et nos résultats sont les meilleurs en termes de FPR95 et de distance moyenne entre les points de repère par rapport au descripteur 3D-SURF, quelle que soit la taille du descripteur.

5 Conclusions et travaux futurs

Nos résultats, bien que préliminaires, montre qu'un descripteur 3D entraîné sur des données semi-synthétiques, peut surpasser un descripteur traditionnel. Nous avons l'intention d'approfondir ces résultats prometteurs en élargissant notre ensemble de données d'entraînement, et en menant d'autres expériences. Les recherches futures porteront sur l'entraînement d'un détecteur de points clés en 3D.

Remerciements

Ce travail a été financé par le projet TOPACS ANR-19-CE45-0015 de l'Agence Nationale de la Recherche (ANR).

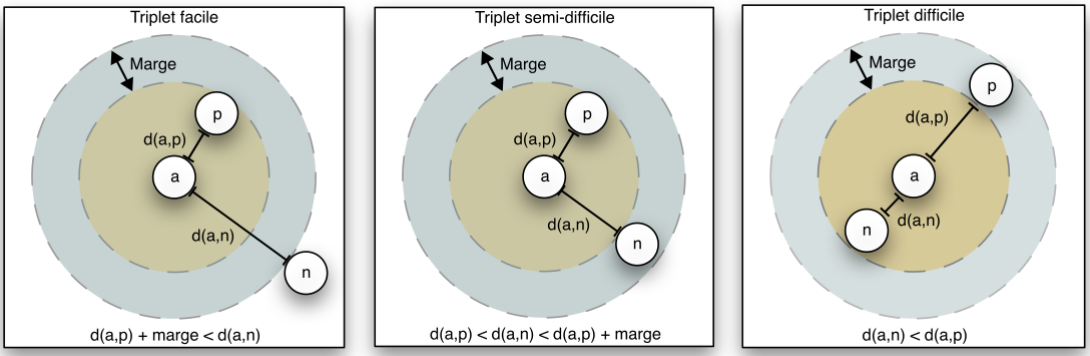

FIGURE 3 – Schéma montrant les trois types de triplets. De gauche à droite, les triplets faciles, les semi-difficiles et les difficiles.

TABLE 1 – Intervalles de recherche pour la recherche des hyperparamètres.

Hyperparamètres	Type de recherche	Intervalle
Marge	Grille	$[0.1, 0.2, 0.4, 0.8, 1.5]$
Taille de descripteur	Grille	$[24, 48, 64, 90, 128]$
Taille du lot	Grille	$[10, 50, 100, 200, 500, 1000]$
Taux d'apprentissage	Aléatoire	$[10^-5, 0.9]$
Moment	Aléatoire	$[0.1, 0.9]$

TABLE 2 – Comparaison des performances entre le descripteur 3D-SURF et notre descripteur appris, en fonction de l'hyperparamètre sur la taille de descripteur.

Type de descripteur	FPR95	Distance moyenne entre les points de repères	Taille de descripteurs
3D-SURF	0.077	8.74	48
Appris	0.01	8.74	24
	0.007	**8.54**	48
	0.008	8.64	64
	0.022	8.64	128

Références

[1] Rémi Agier, Sébastien Valette, Laurent Fanton, Pierre Croisille, and Rémy Prost. Hubless 3d medical image bundle registration. In *VISAPP 2016 11th Joint Conference*, 2016.

[2] Rémi Agier, Sébastien Valette, Razmig Kéchichian, Laurent Fanton, and Rémy Prost. Hubless keypoint-based 3d deformable groupwise registration. *Medical image analysis*, 59 :101564, 2020.

[3] Hani Altwaijry, Andreas Veit, Serge J Belongie, and Cornell Tech. Learning to detect and match keypoints with deep architectures. In *BMVC*, 2016.

[4] Vassileios Balntas, Edgar Riba, Daniel Ponsa, and Krystian Mikolajczyk. Learning local feature descriptors with triplets and shallow convolutional neural networks. In *Bmvc*, page 3, 2016.

[5] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf : Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006.

[6] Maximilian Blendowski and Mattias Heinrich. 3d-cnns for deep binary descriptor learning in medical volume data. In *Bildverarbeitung für die Medizin 2018*, pages 23–28. Springer, 01 2018.

[7] Matthew Brown, Gang Hua, and Simon Winder. Discriminative learning of local image descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(1) :43–57, 2010.

[8] Michael Calonder, Vincent Lepetit, Mustafa Ozuysal, Tomasz Trzcinski, Christoph Strecha, and Pascal Fua. Brief : Computing a local binary descriptor very fast. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7) :1281–1298, 2012.

[9] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint : Self-supervised interest point detection and description, 2018.

[10] Philipp Fischer, Alexey Dosovitskiy, and Thomas Brox. Descriptor matching with convolutional neural networks : a comparison to sift, 2015.

[11] Xufeng Han, Thomas Leung, Yangqing Jia, Rahul Sukthankar, and Alexander C. Berg. Matchnet : Unifying feature and metric learning for patch-based matching. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3279–3286, 2015.

[12] Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. In *International Workshop on Similarity-Based Pattern Recognition*, pages 84–92. Springer, 2015.

[13] M. Jahrer, Michael Grabner, and Horst Bischof. Learned local descriptors for recognition and matching. In *Proceedings of the Computer Vision Winter Workshop 2008*, pages 39–46. ., 2008.

[14] Georg Langs, Allan Hanbury, Bjoern Menze, and Henning Müller. Visceral : towards large data in medical imaging—challenges and directions. In *MICCAI international workshop on medical content-based retrieval for clinical decision support*, pages 92–98. Springer, 2012.

[15] Stefan Leutenegger, Margarita Chli, and Roland Y. Siegwart. Brisk : Binary robust invariant scalable keypoints. In *2011 International Conference on Computer Vision*, pages 2548–2555, 2011.

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

[17] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet : A unified embedding for face recognition and clustering. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2015.

[18] Ivan Sipiran and Benjamin Bustos. Harris 3d : A robust extension of the harris operator for interest point detection on 3d meshes. *The Visual Computer*, 27 :963–976, 11 2011.

[19] Jiang Wang, Yang song, Thomas Leung, Chuck Rosenberg, Jinbin Wang, James Philbin, Bo Chen, and Ying Wu. Learning fine-grained image similarity with deep ranking, 2014.

[20] Kilian Q. Weinberger and Lawrence K. Saul. Distance metric learning for large margin nearest neighbor classification. *J. Mach. Learn. Res.*, 10 :207–244, 2009.

[21] Sergey Zagoruyko and Nikos Komodakis. Learning to compare image patches via convolutional neural networks, 2015.