Noname manuscript No.
(will be inserted by the editor)

Learning 3D Medical Image Keypoint Descriptors with the Triplet Loss

Nicolas Loiseau–Witon · Razmig Kéchichian · Sébastien Valette · Adrien Bartoli

Received: date / Accepted: date

Abstract
Purpose. We propose to learn a 3D keypoint descriptor which we use to match keypoints extracted from full-body CT scans. Our methods are inspired by 2D keypoint descriptor learning, which was shown to outperform hand-crafted descriptors. Adapting these to 3D images is challenging because of the lack of labelled training data and high memory requirements.
Method. We generate semi-synthetic training data. For that, we first estimate the distribution of local affine inter-subject transformations using labelled anatomical landmarks on a small subset of the database. We then sample a large number of transformations and warp unlabelled CT scans, for which we can subsequently establish reliable keypoint correspondences using guided matching. These correspondences serve as training data for our descriptor, which we represent by a CNN and train using the triplet loss with online triplet mining.
Results. We carry out experiments on a synthetic data reliability benchmark and a registration task involving 20 CT volumes with anatomical landmarks used for evaluation purposes. Our learned descriptor outperforms the 3D-SURF descriptor on both benchmarks while having a similar runtime.
Conclusion. We propose a new method to generate semi-synthetic data and a new learned 3D keypoint descriptor. Experiments show improvement compared to a hand-crafted descriptor. This is promising as literature has shown that jointly learning a detector and a descriptor gives further performance boost.

N. LOISEAU–WITON, S. VALETTE, R. KECHICHIAN
Univ Lyon, INSA-Lyon, Université Claude Bernard Lyon 1, UJM-Saint Etienne, CNRS, Inserm, CREATIS UMR 5220, U1206, F-69100, LYON, France
E-mail: lastname@creatis.insa-lyon.fr

A. BARTOLI
Institut Pascal, UMR 6602 CNRS/UCA/CHU, Clermont-Ferrand, France
E-mail: Adrien.Bartoli@gmail.com

Keywords Medical imaging · Keypoints · Descriptors · Convolution Neural Network · Triplet loss

1 Purpose

Computational anatomy focuses on the analysis of the variability of the human anatomy. Typical applications are the discovery of differences across healthy and diseased subjects and the classification of anomalies. A fundamental tool in computational anatomy, which forms the central focus of this paper, is the computation of point correspondences across volumes (3D images) such as Computed Tomography (CT) volumes, for multiple subjects. Keypoint descriptors are used to find correspondences across images to further achieve image registration. More specifically, we consider automatically detected keypoints and their local descriptors, computed from the image or the 3D volume patch surrounding each keypoint. These descriptors are essential and must be discriminant and repeatable [5, 13]. Learned descriptors based on Convolutional Neural Networks (CNN) have recently shown great success for 2D images [4]. However, while classical 2D image descriptors were extended to volumes [1, 16], recent learning-based approaches have been limited to 2D detection and description. The extension to 3D descriptors was only proposed in [6], where a network is trained to generate descriptors with binary components, which allows a fast computation of a similarity metric adapted to an 3D image retrieval task. However, the implementation of this method is not publicly available, preventing direct empirical comparison. We propose a methodology to learn 3D keypoint descriptors from volumetric data suitable for image registration. The main difficulty is to define a sound training approach, combining a training dataset and a loss function. In short, we propose to generate semi-synthetic data by transforming real volumes and to use a triplet loss inspired by 2D descriptor learning. Our experimental results show that our learned descriptor outperforms the hand-crafted descriptor 3D-SURF [1], a 3D extension of SURF, with similar runtime.

2 Methods

Our first goal is to create a reference dataset defining keypoint correspondences between multiple volumes. In 2D, these correspondences can be established using Structure-from-Motion [3]. For 3D medical images, however, no such large annotated dataset is publicly available. We thus propose to create a semi-synthetic reference dataset by transforming real volumes.

2.1 Constructing a semi-synthetic dataset

We use two subsets from the Visceral dataset [12]. The first subset, named *Gold*, contains 20 CT volumes, each annotated with about 40 anatomical landmarks, placed by medical experts. These landmarks were visualised in [11]. The second subset, *Silver*, contains 60 CT volumes without any landmarks.

In order to generate new volumes, we estimate the probability density of possible inter-volume transformations, and sample from this density to create new CT volumes by warping original volumes. We subsequently define keypoint correspondences between original and generated volumes using guided matching. We first estimate a local affine transformation between two subjects using the minimal amount of 4 point correspondences. We use the anatomical landmarks, providing reliable correspondences. Landmarks are points placed in images by experts on certain anatomical locations, contrary to keypoints which are automatically detected by handcrafted or learned detectors and do not necessarily correspond to similar anatomical locations. For each landmark in each volume of the *Gold* subset, we use the landmark, its three closest landmarks and the four corresponding landmarks in another volume to estimate a local transformation. When the landmark and its three neighbours are almost collinear (e.g. vertebral landmarks), the problem is ill-conditioned and we therefore discard the transformation. Thus we obtain at most $Lk(k-1)/2$ affine transformation matrices of size 4×4, where $L = 40$ is the number of landmarks and $k = 20$ is the number of volumes. Now with these affine transformation matrices obtained on each subject, we compute the element-wise Pearson correlation coefficients, excluding elements corresponding to translation, and obtain a 9x9 coefficient matrix. Inspecting this matrix confirms that its non diagonal elements are close to 0, hence that the estimated transformation parameters are pairwise independent. This allows us to sample each parameter independently. We apply Kernel Density Estimation (KDE) to these matrices to estimate the density of inter-volume transformations. Student's t-test shows that a Gaussian Kernel is a good fit for the KDE. We estimate the kernel bandwidth via Scott's rules [15]. To generate our semi-synthetic dataset, we sample transformations from this density and apply them to volumes in the *Silver* subset. More specifically, for a sampled transformation t and a silver volume V_i, we obtain the volume V_i^t. We detect the keypoints in V_i and V_i^t using 3D-SURF and obtain two keypoint sets P_i and P_i^t. Note that 3D-SURF is both a detector and a descriptor. We then apply the inverse transform t^{-1} to the keypoints from V_i^t. Finally we implement guided matching using a k-d tree to construct the set of corresponding keypoints between the volumes as the set of pairs: $(p \in P_i, q \in t^{-1}(P_i^t))$, where the distance p to q is lower than 8 mm. This threshold was chosen to obtain a large number of correct correspondences. As the voxels have side length of about 1 mm, 8 mm is slightly smaller than 10 voxels in our images.

2.2 Training the descriptor with the triplet loss

We learn a descriptor CNN mapping a 3D patch of 10^3 voxels surrounding a keypoint to a descriptor vector. We use a patch size of 10^3 for two reasons: memory requirements which grow cubically in 3D, and our use of the output of

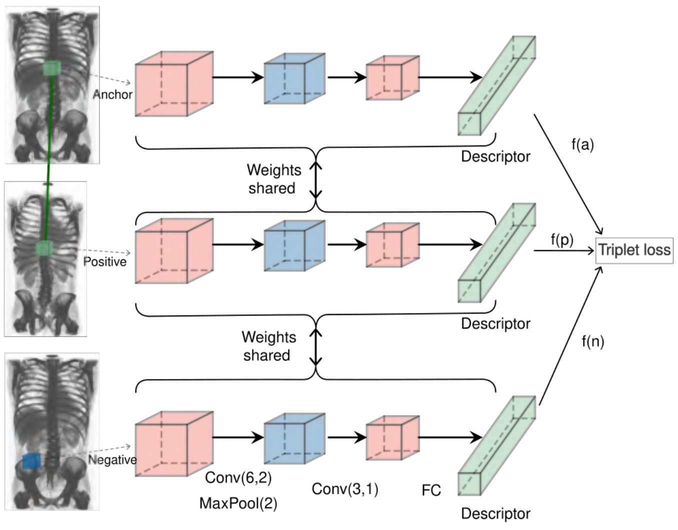


Fig. 1: Architecture of our CNN and the triplet-loss. The input patches are 10^3 voxels. The CNN has two convolutions with *tanh* activation, one max-pooling and a fully-connected layer. The triplet $\{a, p, n\}$ is passed through the network, and descriptor vectors are fed to the triplet loss.

3D-SURF in training and testing, which is of dimension of 10^3. Recent work in 2D has shown that learning descriptors using triplets yields better results than using pairs [10]. Triplet learning requires forming triplets of patches $\{a, p, n\}$ where a is an anchor, p a positive representing a different patch of the same class as a, and n a negative representing a patch of a different class. In our case a and p are two patches around corresponding keypoints from different volumes and n is a patch around a different keypoint. The aim is to optimize the CNN parameters in order to bring a and p close together in descriptor space and to push n away from a. Thus the triplet loss is defined by:

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

where $f(\cdot)$ is the CNN and α the margin parameter.

By minimizing this loss, the network brings the distance between a and p, $d(a, p)$, towards 0 and the distance between a and n, $d(a, n)$, to greater than $d(a, p) + \alpha$. For a given network state, randomly selected triplets are not equally useful for training. A triplet with high loss in the beginning can have zero loss later in the training. We thus follow an online triplet mining approach for selective training on the most useful triplets. First, we define three types of triplets: easy triplets with a zero loss, where $d(a, p) + \alpha < d(a, n)$; hard triplets with $d(a, n) < d(a, p)$; semi-hard triplets, where $d(a, p) < d(a, n) < d(a, p) + \alpha$. Using this definition, we exclude easy triplets from the next training round by identifying hard and semi-hard triplets based on respective loss values, and feed

Table 1: Ranges used for hyperparameter tuning.

Hyperparameter	Search type	Range
Margin	Grid	$[0.1, 0.2, 0.4, 0.8, 1.5]$
Descriptor size	Grid	$[24, 48, 64, 90, 128]$
Batch size	Grid	$[10, 50, 100, 200, 500, 1000]$
Learning rate	Random	$[10^{-5}, 0.9]$
Momentum	Random	$[0.1, 0.9]$

a subset to the network: for every mini-batch of size b, we form all possible triplets, and keep the b triplets with the highest loss value.

Our CNN is defined by two 3D convolution layers, the first one is a convolution with a kernel size of 3 and with 32 output channels, the second one has a kernel size of 2 and 64 output channels. The max-pooling layer between the convolutions has a kernel size of 2 and a stride of 2. The last layer is a fully-connected layer which gives the final descriptor. The network architecture is illustrated in Figure 1. Passing a patch of size 10^3 through this CNN gives us a descriptor vector of the desired size. We use a size of 48 for direct comparability with 3D-SURF; this size of 48 corresponds to the descriptor size which is the vector size of a described patch after passing through the CNN. We also tested alternative descriptor sizes experimentally. The descriptor size is unrelated to the patch size; it is dimensionless, hence unitless. For example, the well-known SIFT and 3D-SURF descriptors have sizes of 128 and 48 respectively [5, 13].

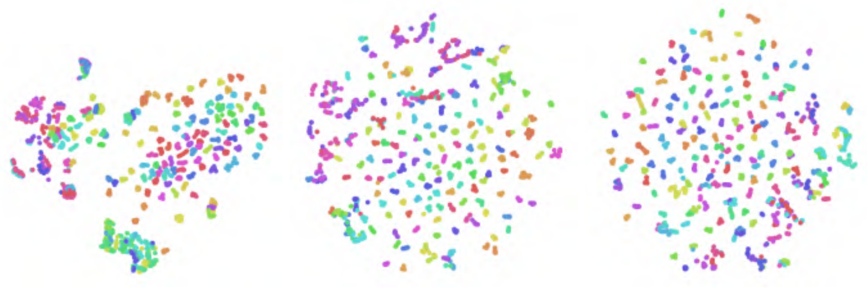
3 Results

3.1 Data split

We divide the *Silver* dataset into training and validation subsets. The training data consists of 55 subjects with 10 transformed volumes each, following our procedure of semi-synthetic data generation. The validation data consist of 5 subjects with 10 transformed volumes each. For testing, we use the *Gold* subset with 20 subjects and the associated anatomical landmarks.

3.2 Training

Optimization is performed via Stochastic Gradient Descent, with a batch size of 1000 patches, a learning rate of 0.1, a momentum of 0.9, a weight decay of 10^{-6} (L_2 regularization weight) and a loss margin α of 0.2. We also use online triplet mining to find the best triplets for learning. Our CPU-based implementation uses the PyTorch library. The training of a single epoch with 10^6 triplets takes about 30 minutes and approximately 10 GB of memory on



(a) patches around keypoints (b) 3D-SURF descriptors (c) learned descriptors

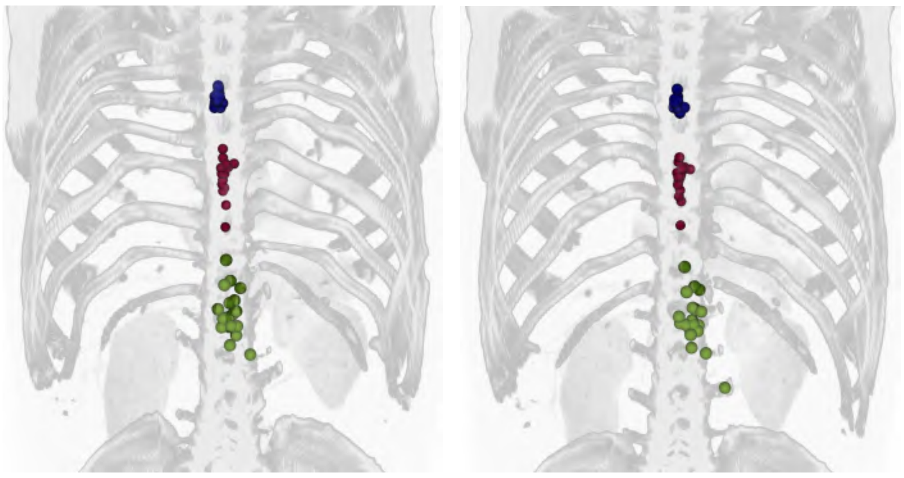
Fig. 2: 2D t-SNE embeddings of 200 randomly selected sets of 10 corresponding keypoints (identical colors) detected by 3D-SURF, (a) patches around detected keypoints, (b) SURF descriptors of the same keypoints, (c) descriptors of the same keypoints, calculated by our CNN.

a Linux 64-bit platform running on an Intel Xeon 2.6 GHz CPU. Our model is light enough to be trained on a CPU; GPU-based training did not significantly reduce training time as the training process is mostly I/O-bound i.e. loading image patches is the bottleneck. Our hyperparameter search was done with Ray Tune. We followed a grid search strategy for batch size, descriptor size parameters and a random search strategy for learning rate, margin and momentum parameters. Table 1 summarizes parameter ranges.

3.3 Evaluation

To illustrate the effect of learning descriptors, Figure 2 shows the 2D projection of keypoint descriptors computed via the t-SNE method [14]. In this figure, 200 keypoints were randomly selected in the validation dataset. As each keypoint is present in 10 transformed volumes, there is a total of 2000 keypoints. Corresponding keypoints are displayed with the same color. Figure 2a shows the 2D projection of original patches, while Figure 2c shows the projection of descriptors computed by our CNN. Our descriptors yield tighter clusters, with a better overall spatial distribution, similar to that of the 3D-SURF descriptors, shown in Figure 2b.

We have carried out quantitative evaluation with two different metrics. The first metric, measuring the reliability of the CNN, is the false positive rate at 0.95 true positive recall (FPR95) [7]. This metric indicates performances in terms of the percentage of false matches when 95% of all correct matches are detected. To compute the FPR95, 100k pairs of keypoints with 50k corresponding and 50k non corresponding pairs from the generated dataset were randomly selected. We extract patches around these 100k pairs of keypoints, compute their descriptor and compute euclidean distance between each pair of descriptors. We can now compute FPR95 with the percentage of false and



(a) Registration with 3D-SURF descriptors (b) Registration with CNN descriptors

Fig. 3: 3 different landmark types in the registration common space for 20 patients: tracheal bifurcation (best case, blue, top), thoracic vertebra #9 (intermediate case, red, middle) and xiphoid (worst case, green, bottom), registered with 3D-SURF descriptors (left) and our CNN descriptors (right).

correct matches. A low FPR95 indicates good results. The second metric is the mean landmark distance (in millimeters) calculated on ground-truth anatomical landmarks in *Gold* volumes after registering them in a common space using the keypoint-based FROG registration algorithm [2]. For comparison, we replace the 3D-SURF descriptor used in this algorithm with our learned descriptor. Low mean distance between landmarks indicates good results. Table 2 compares the results between the 3D-SURF descriptor and our learned descriptor with different descriptor sizes. Our descriptor gives the best results: it outperforms the 3D-SURF descriptor in terms of both FPR95 and mean landmark distance, regardless of the descriptor size.

To illustrate the landmark spread after registration, Figure 3 shows 3 different landmark types in the common space: tracheal bifurcation (blue), thoracic vertebra #9 (red) and xiphoid (green). The registration was carried out with the FROG algorithm using 3D-SURF (left) and our learned descriptors (right). As each landmark is present in 20 subjects from the *Gold* group, there is a total of 60 landmarks. The best case of landmark registration, for both descriptors, corresponds to the tracheal bifurcation (blue), with a mean distance of 4.8 mm for CNN and 4.6 mm for 3D-SURF. The worst case, for both descriptors, corresponds to the xyphoid (red) with a mean distance of 14.4 mm for CNN and 12.7 mm for 3D-SURF. The thoracic vertebra #9 (blue) corresponds to an intermediate case, with a mean distance of 8.4 mm for CNN and 10.5 mm for 3D-SURF. Analysis shows that our method yields better results than 3D-SURF for all vertebral landmarks.

Table 2: Performance comparison of the 3D-SURF and our learned descriptor with varying descriptor size.

Type of descriptor	FPR95	Mean landmark distance (mm)	Descriptor size
3D-SURF	0.077	8.74	48
Learned	0.010	8.74	24
	0.007	**8.54**	48
	0.008	8.64	64
	0.022	8.64	128

4 Conclusions and future work

Our results, although preliminary, show that a learned 3D descriptor, trained on semi-synthetic data, can outperform a carefully hand-crafted one. We intend to further explore these promising results by extending our training dataset and conducting more experiments. Future research will address training a 3D keypoint detector. Indeed, as shown in recent work [9, 8], simultaneously learning a detector and a descriptor has given better results than learning them separately.

Declarations

Funding

This work was funded by the TOPACS ANR-19-CE45-0015 project of the French National Research Agency (ANR).

Conflicts of interest/Competing interests

The authors declare that they have no conflict of interest.

Ethical approval

All procedures performed in studies involving human participants were in accordance with the ethical standards of the institutional and/or national research committee and with the 1964 Helsinki declaration and its later amendments or comparable ethical standards.

Informed consent

Informed consent was obtained by the VISCERAL project from all individual participants included in the study

References

1. Agier R, Valette S, Fanton L, Croisille P, Prost R (2016) Hubless 3d medical image bundle registration. In: VISAPP 2016 11th Joint Conference
2. Agier R, Valette S, Kéchichian R, Fanton L, Prost R (2020) Hubless keypoint-based 3d deformable groupwise registration. Medical image analysis 59:101564
3. Altwaijry H, Veit A, Belongie SJ, Tech C (2016) Learning to detect and match keypoints with deep architectures. In: BMVC
4. Balntas V, Riba E, Ponsa D, Mikolajczyk K (2016) Learning local feature descriptors with triplets and shallow convolutional neural networks. In: Bmvc, vol 1, p 3
5. Bay H, Tuytelaars T, Van Gool L (2006) Surf: Speeded up robust features. In: European conference on computer vision, Springer, pp 404–417
6. Blendowski M, Heinrich M (2018) 3d-cnns for deep binary descriptor learning in medical volume data. In: Bildverarbeitung für die Medizin 2018, Springer, pp 23–28
7. 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
8. DeTone D, Malisiewicz T, Rabinovich A (2018) Superpoint: Self-supervised interest point detection and description. 1712.07629
9. 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
10. Hoffer E, Ailon N (2015) Deep metric learning using triplet network. In: International Workshop on Similarity-Based Pattern Recognition, Springer, pp 84–92
11. Krenn M, Grünberg K, Jimenez-del Toro O, Jakab A, Fernandez TS, Winterstein M, Weber MA, Langs G (2017) Datasets created in visceral. In: Cloud-Based Benchmarking of Medical Image Analysis, Springer, Cham, pp 69–84
12. Langs G, Hanbury A, Menze B, Müller H (2012) Visceral: towards large data in medical imaging—challenges and directions. In: MICCAI international workshop on medical content-based retrieval for clinical decision support, Springer, pp 92–98
13. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. International journal of computer vision 60(2):91–110
14. van der Maaten L, Hinton G (2008) Visualizing data using t-sne. Journal of Machine Learning Research 9(86):2579–2605, URL http://jmlr.org/papers/v9/vandermaaten08a.html
15. Scott D (2015) Multivariate density estimation: Theory, practice, and visualization: Second edition
16. 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