LCD: Learned Cross-Domain Descriptors for 2D-3D Matching

Quang-Hieu Pham,[1] **Mikaela Angelina Uy,**[2] **Binh-Son Hua,**[3]
Duc Thanh Nguyen,[4] **Gemma Roig,**[5] **Sai-Kit Yeung**[6]

[1]Singapore University of Technology and Design, [2]Stanford University, [3]The University of Tokyo,
[4]Deakin University, [5]Geothe University of Frankfrut am Main, [6]Hong Kong University of Science and Technology
quanghieu_pham@mymail.sutd.edu.sg, mikacuy@stanford.edu, hua.binhson@mail.u-tokyo.ac.jp,
duc.nguyen@deakin.edu.au, gemmar@mit.edu, saikit@ust.hk

Abstract

In this work, we present a novel method to learn a *local cross-domain descriptor* for 2D image and 3D point cloud matching. Our proposed method is a dual auto-encoder neural network that maps 2D and 3D input into a shared latent space representation. We show that such local cross-domain descriptors in the shared embedding are more discriminative than those obtained from individual training in 2D and 3D domains. To facilitate the training process, we built a new dataset by collecting ≈ 1.4 millions of 2D-3D correspondences with various lighting conditions and settings from publicly available RGB-D scenes. Our descriptor is evaluated in three main experiments: 2D-3D matching, cross-domain retrieval, and sparse-to-dense depth estimation. Experimental results confirm the robustness of our approach as well as its competitive performance not only in solving cross-domain tasks but also in being able to generalize to solve sole 2D and 3D tasks. Our dataset and code are released publicly at https://hkust-vgd.github.io/lcd.

Introduction

Computer vision tasks such as structure-from-motion, visual content retrieval require robust descriptors from both 2D and 3D domains. Such descriptors, in their own domain, can be constructed from low-level features, *e.g*., colors, edges, etc. In image matching, a well-known task in computer vision, several hand-crafted local descriptors, *e.g*., SIFT (Lowe 2004), SURF (Bay, Tuytelaars, and Van Gool 2006) have been proposed. With the advent of deep learning, many robust 2D descriptors are learned automatically using deep neural networks (Simo-Serra et al. 2015; Kumar et al. 2016). These learned descriptors have shown their robustness and advantages over the hand-crafted counterparts. The same phenomenon can also be observed in 3D domain. For example, hand-crafted 3D descriptors, *e.g*., FPFH (Rusu, Blodow, and Beetz 2009), SHOT (Tombari, Salti, and Di Stefano 2010), as well as deep learning based descriptors (Zeng et al. 2017) have been used in many 3D tasks, such as 3D registrations (Choi, Zhou, and Koltun 2015; Zhou, Park, and Koltun 2016) and structure-from-motion (Hartley and Zisserman 2003).

While 2D and 3D descriptors are widely available, finding the association between these representations is a challenging task. There also lacks a descriptor that can capture features in both domains and tailored for cross-domain tasks, for example, 2D to 3D content retrieval. In general, there is a large discrepancy between 2D and 3D representations. Data in 2D, *i.e*., images, can simply be represented by regular grids. Meanwhile, 3D data can be represented by either meshes, volumes, or point clouds and obtained via an image formation model that is governed by laws of physics and optics. Even with the recent advent of deep learning, these issues still remain the same: features learned on 2D domain may not be applicable in 3D space and vice versa.

In this work, we attempt to bridge the gap between 2D and 3D descriptors by proposing a novel approach to learn a cross-domain descriptor that works on both 2D and 3D domains. In particular, we make the following contributions:

- A novel learned cross-domain descriptor (LCD) that is learned using a dual auto-encoder architecture and a triplet loss. Our setup enforces the 2D and 3D auto-encoders to learn a cross-domain descriptor in a shared latent space representation. This shared latent space not only provides a common space for 2D-3D matching, but also improves the descriptor performance in single-domain settings.

- A new public dataset of ≈ 1.4 millions of 2D-3D correspondences for training and evaluating the cross-domain descriptors matching. We built our dataset based on SceneNN (Hua et al. 2016) and 3DMatch (Zeng et al. 2017).

- Applications to verify the robustness of our cross-domain descriptor. Specifically, we apply the descriptor to solve a sole 2D (image matching) and a sole 3D task (3D registration), and then to a 2D-3D content retrieval task (2D-3D place recognition). Experimental results show that our descriptor gives comparable performance to other state-of-the-art methods in all the tasks even when it is not purposely tailored to such particular tasks.

Related work

Local descriptors are crucial components in many applications such as localization (Sattler et al. 2017), registration (Choi, Zhou, and Koltun 2015; Zhou, Park, and

Koltun 2016), Structure-from-Motion (SfM) (Hartley and Zisserman 2003), Simultaneous Localization and Mapping (SLAM) (Durrant-Whyte and Bailey 2006), and pose estimation (Haralick et al. 1991). In general, 2D descriptors are obtained from the 2D local patches of images, whereas 3D descriptors are often computed from 3D point clouds.

2D descriptors. Image descriptors, both fully hand-crafted (Lowe 2004; Bay, Tuytelaars, and Van Gool 2006; Rublee et al. 2011) and partially learned (Brown, Hua, and Winder 2010; Tola, Lepetit, and Fua 2009), have been well studied in early days of computer vision. Recently, deep learning has been applied for end-to-end learning of 2D descriptors (Chopra, Hadsell, and LeCun 2005). The robustness of the learned descriptors over the handcrafted ones have been proven clearly in image matching. For example, (Zagoruyko and Komodakis 2015) and (Han et al. 2015) proposed a Siamese architecture to learn a similarity score between a given pair of image patches. However, these methods are computationally expensive as the image patches need to be pairwise passed into the network. To make the solution tractable, in (Simo-Serra et al. 2015; Tian, Fan, and Wu 2017), the descriptor was learned with the same Siamese architecture but matched using Euclidean distance. This allows the learned descriptor to be used as a direct replacement to traditional hand-crafted descriptors, and nearest neighbor queries could be done efficiently in matching. Our work is built upon this idea, but instead we learn a cross-domain 2D-3D descriptor.

More recently, triplet networks (Balntas et al. 2017) taking three image patches as input for learning descriptors have been introduced. These networks showed that learning with a triplet loss (Schroff, Kalenichenko, and Philbin 2015; Hermans, Beyer, and Leibe 2017) resulted in a better embedding space. Further improvements to the triplet loss were studied in (Mishchuk et al. 2017; Keller et al. 2018). Joint learning of feature detector and descriptor was explored by (Yi et al. 2016). In general, all of these works take image patches as input and learn a feature space for 2D descriptors. In contrast, our work aims to learn a shared latent space for both 2D and 3D descriptors. In addition to leveraging metric learning, we also utilize auto-encoders to learn a more discriminative space.

3D descriptors. Unlike 2D descriptors, 3D descriptors for point clouds such as PFH (Rusu et al. 2008), FPFH (Rusu, Blodow, and Beetz 2009), and SHOT (Tombari, Salti, and Di Stefano 2010) do not reach the same level of robustness and maturity. These methods either require stable surfaces or sufficient point densities. Deep learning solutions have also been proposed to tackle these problems. For example, 3DMatch (Zeng et al. 2017) used voxelized patches to calculate 3D local descriptors in convolutions to register RGB-D scans. (Dewan, Caselitz, and Burgard 2018) also applied convolutions on 3D voxels to learn local descriptors for LiDAR scans. However, such architectures cannot be used directly on point cloud due to their irregular input structure that disables convolutions. To address this issue, (Khoury,

Zhou, and Koltun 2017) proposed to reduce hand-crafted point cloud descriptor dimension through deep learning for efficient matching.

Recently, PointNet (Qi et al. 2017) introduced the first deep neural network that can directly operate on point clouds. This network then became the backbone for multiple point-based networks (Deng, Birdal, and Ilic 2018b; 2018a; Yew and Lee 2018). In particular, PPFNet (Deng, Birdal, and Ilic 2018b) used point pair features to learn local point cloud descriptors for registration. 3DFeat-Net (Yew and Lee 2018) proposed a weakly-supervised approach to learn local descriptors with only GPS/INS tags for outdoor data. PPF-FoldNet (Deng, Birdal, and Ilic 2018a) made use of an auto-encoder to learn point cloud descriptors in an unsupervised manner. Other deep learning descriptors on point cloud include KeypointNet (Suwajanakorn et al. 2018), USIP (Li and Lee 2019), and PointNetVLAD (Angelina Uy and Hee Lee 2018). These methods address the problem of 3D keypoint detection and LiDAR-based place recognition, and thus, are designed to match 3D structures only. On the other hand, our work handles both 2D image patches and 3D point clouds in a unified manner.

2D-3D cross-domain descriptors. (Li et al. 2015) proposed a joint global embedding of 3D shapes and images to solve the retrieval task. The 3D embeddings were first hand-craftedly constructed and image embeddings were learned to adhere to the 3D embeddings. In contrast, our network jointly learns both 2D and 3D embeddings for local descriptors. (Xing et al. 2018) proposed a network, called 3DTNet, that receives both 2D and 3D local patches as input. However, 3DTNet was only designed for 3D matching and the network used 2D features as auxiliary information to make 3D features more discriminative. Some other works also established the connection between 2D and 3D for specific applications such as object pose estimation (Lim, Pirsiavash, and Torralba 2013; Xiao, Russell, and Torralba 2012) and 3D shape estimation (Hejrati and Ramanan 2012). Recently, (Feng et al. 2019) proposed a deep network to match 2D and 3D patches with a triplet loss for outdoor localization. Our work differs from this method in that our goal is not to learn an application-specific descriptor, but a cross-domain descriptor for generalized 2D-3D matching that can be used in various tasks as proven in our experiments.

Learned cross-domain descriptors (LCD)
Problem definition
Let $I \in \mathbb{R}^{W \times H \times 3}$ be a colored image patch of size $W \times H$ and represented in conventional RGB color space. Similarly, let $P \in \mathbb{R}^{N \times 6}$ be a colored point cloud of N points, each point is represented by its coordinates $(x, y, z) \in \mathbb{R}^3$ and RGB color.

Our goal of learning a cross-domain descriptor is to find two mappings $f : \mathbb{R}^{W \times H \times 3} \mapsto \mathcal{D}$ and $g : \mathbb{R}^{N \times 6} \mapsto \mathcal{D}$ that maps the 2D and 3D data space to a shared latent space $\mathcal{D} \subseteq \mathbb{R}^D$, where D is the dimension of the embedding such that for each pair of 2D-3D correspondence (I, P), their mappings are as similar as possible. Mathematically, given
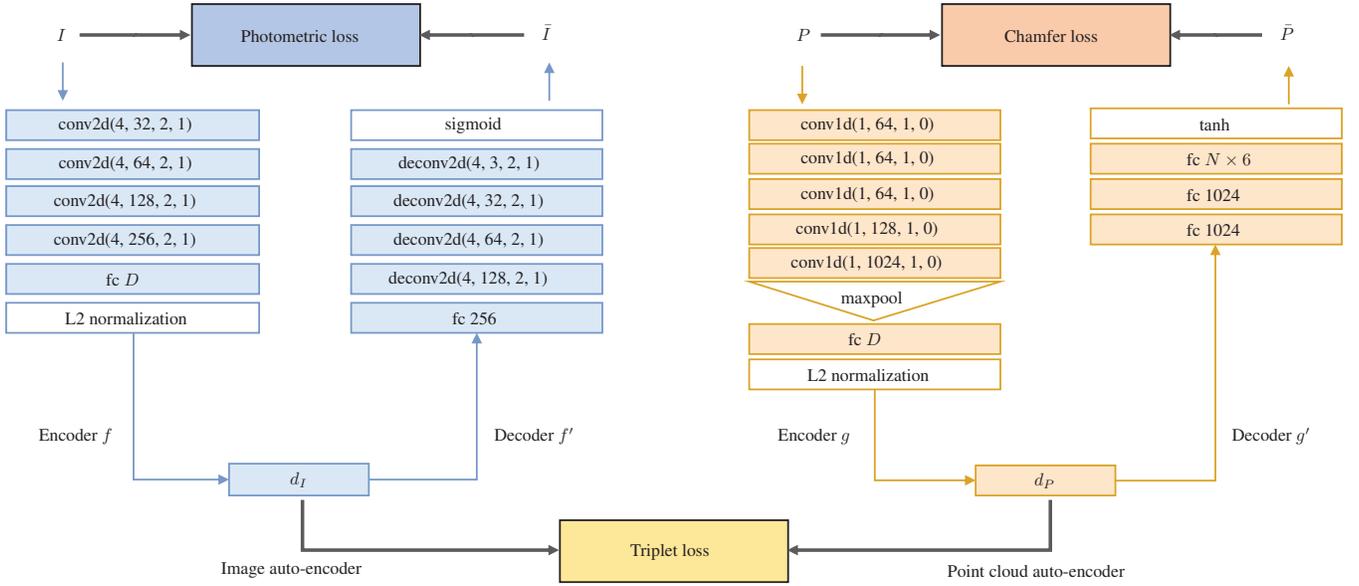
Figure 1: Our proposed network consists of a 2D auto-encoder and a 3D auto-encoder. The input image and point cloud data is reconstructed with a photometric and a Chamfer loss, respectively. The reconstruction losses ensures features in the embedding to be discriminative and representative. The similarity between the 2D embedding d_I and the 3D embedding d_P is further regularized by a triplet loss. Diagram notation: `fc` for fully-connected, `conv/deconv`(`kernel_size`, `out_dim`, `stride`, `padding`) for convolution and deconvolution, respectively. Each convolution and deconvolution is followed by a ReLU activation and a batch normalization by default.

a distance function $\mathcal{F}$ and two descriptors $d_I, d_P \in \mathcal{D}$, if I and P are represented the same underlying geometry, then $\mathcal{F}(d_I, d_P) < m$, where m is a predefined margin.

In addition to mapping data to descriptors, we also aim to learn the inverse mapping functions $f' : \mathcal{D} \mapsto \mathbb{R}^{W \times H \times 3}$ and $g' : \mathcal{D} \mapsto \mathbb{R}^{N \times 6}$. Being able to reconstruct data from descriptors, these inverse mappings are beneficial in downstream applications such as 3D sparse-to-dense depth estimation, as shown later in our experiments.

Network architecture

Inspired by the success of using auto-encoders in construction of descriptors (Deng, Birdal, and Ilic 2018a), we propose a novel dual auto-encoder architecture to learn descriptors. Our model is a two-branch network architecture, where one branch encodes 3D features, and the other branch encodes 2D features. The two branches are then jointly optimized using a triplet loss enforcing the similarity of embeddings generated by the two branches as well as the 2D/3D reconstruction losses. Our network architecture is illustrated in Figure 1.

For the 2D branch, our 2D auto-encoder takes input a colored image patch of size 64×64 and processes it through a series of convolutions with ReLU activation in order to extract image features. For the 2D decoder, we use a series of transpose convolutions with ReLU to reconstruct the image patch. For the 3D branch, we adopt the well-known PointNet architecture (Qi et al. 2017), employing a series of fully-connected layers then max-pooling to calculate a global feature. To reconstruct the colored point cloud, we employ another series of fully-connected layers which outputs a colored

point cloud of size $N \times 6$. To enforce a shared representation, the two auto-encoders are tied up between their bottlenecks by optimizing a triplet loss. The final training loss combines photometric loss, Chamfer loss, and triplet loss as follows.

Photometric loss. The 2D auto-encoder loss is defined by the photometric loss, which is the mean squared error between the input 2D patch I and the reconstructed patch $\bar{I}$:

$$\mathcal{L}_{mse} = \frac{1}{W \times H} \sum_{i=1}^{W \times H} \left\| I_i - \bar{I}_i \right\|^2, \quad (1)$$

where I_i and $\bar{I}_i$ denote the ith pixel in the input and reconstructed image patches, respectively.

Chamfer loss. To optimize the 3D auto-encoder network, we need to compute the distance between the input point set P and the reconstructed point set $\bar{P}$. We measure this distance via the well known Chamfer distance:

$$\mathcal{L}_{chamfer} = \max \left\{ \frac{1}{|P|} \sum_{p \in P} \min_{q \in \bar{P}} \|p - q\|_2 , \right.$$
$$\left. \frac{1}{|\bar{P}|} \sum_{q \in \bar{P}} \min_{p \in P} \|p - q\|_2 \right\}. \quad (2)$$

Triplet loss. To enforce the similarity in the embeddings generated by the 2D and 3D branch, *i.e.*, a 2D image patch

and its corresponding 3D structures should have similar embeddings, we employ the triplet loss function. This loss minimizes the distance between an anchor and a positive, while maximizes the distance between the anchor and a negative. Following (Hermans, Beyer, and Leibe 2017), we perform online batch-hardest negative mining that can improve both train and test performance. The triplet loss function can be written as follows:

$$\mathcal{L}_{triplet} = \max(\mathcal{F}(d_a, d_p) - \mathcal{F}(d_a, d_n) + m, 0), \quad (3)$$

where m is the margin and $\mathcal{F}$ is the distance function. (d_a, d_p, d_n) is a triplet consisting of an anchor, a positive, and a hardest negative, respectively.

Training loss. In summary, the loss function that is used to train our network is defined as:

$$\mathcal{L} = \alpha \cdot \mathcal{L}_{mse} + \beta \cdot \mathcal{L}_{chamfer} + \gamma \cdot \mathcal{L}_{triplet}, \quad (4)$$

where α, β, and γ are the weights to emphasis the importance of each sub-network in the training process. We set $\alpha = \beta = \gamma = 1$ in our implementation.

The proposed architecture holds several advantages. First, the 2D and 3D branch capture important features in 2D and 3D domains. When these branches are trained jointly, domain invariant features would be learned and integrated into the embeddings. Second, having auto-encoders in the architecture enables the transformation of descriptors across 2D and 3D domains as shown in our experiments.

Implementation details

Our network requires a dataset of 2D-3D correspondences to train. To the best of our knowledge, there is no such publicly available dataset. Therefore, we build a new dataset of 2D-3D correspondences by leveraging the availability of several 3D datasets from RGB-D scans. In this work, we use the data from SceneNN (Hua et al. 2016) and 3DMatch (Zeng et al. 2017). SceneNN is a comprehensive indoor dataset scanned by handheld RGB-D sensor with fine-grained annotations. 3DMatch dataset is a collection of existing RGB-D scenes from different works (Glocker et al. 2013; Xiao, Owens, and Torralba 2013; Valentin et al. 2016; Dai et al. 2017; Henry et al. 2013; Halber and Funkhouser 2017). We follow the same train and test splits from (Zeng et al. 2017) and (Hua, Tran, and Yeung 2018). Our training dataset consists of 110 RGB-D scans, of which 56 scenes are from SceneNN and 54 scenes are from 3DMatch. The models presented in our experiments are all trained on the same dataset.

The 2D-3D correspondence data is generated as follows. Given a 3D point which is randomly sampled from a 3D point cloud, we extract a set of 3D patches from different scanning views. To find a 2D-3D correspondence, for each 3D patch, we re-project its 3D position into all RGB-D frames for which the point lies in the camera frustum, taking occlusion into account. We then extract the corresponding local 2D patches around the re-projected point. In total, we collected $1,465,082$ 2D-3D correspondences, with varying lighting conditions and settings.

Our network is implemented in PyTorch. The network is trained using SGD optimizer, with learning rate set to 0.01.

We train our network on a cluster equipped with NVIDIA V100 GPUs and 256 GB of memory. It takes around 17 hours to train our network, stopping after 250 epochs.

Experiments

In this section, we evaluate our proposed cross-domain descriptor under a wide range of applications, showing that the learned descriptor can work on both 2D and 3D domains. We also explore the effect of the output feature dimension D on the descriptor's performance. In our experiments, we train and test with $D \in \{64, 128, 256\}$, denoted as *LCD-D** in the results. We evaluate the performance of our 2D descriptor on the task of image matching. Then, we demonstrate the capability of our 3D descriptor in the global registration problem. Our cross-domain descriptor also enables unique applications, such as 2D-3D place recognition and sparse-to-dense depth estimation. Finally, we validate our network design by conducting in-depth ablation study.

2D image matching

We first evaluate our descriptor in a classic 2D computer vision task — image matching. We use the SceneNN dataset, which contains around $100K$ of RGB-D frames with ground-truth pose estimation. We use 20 scenes for testing, following the same split from (Hua, Tran, and Yeung 2018). We consider the following descriptors as our competitors: traditional hand-crafted descriptors *SIFT* (Lowe 2004), *SURF* (Bay, Tuytelaars, and Van Gool 2006), and a learned descriptor *SuperPoint* (DeTone, Malisiewicz, and Rabinovich 2018). We also show the result from *PatchNetAE*, which is the single 2D branch of our network trained only with image patches.

To evaluate the performance over different baselines, we sample image pairs at different frame difference values: 10, 20, and 30. SuperPoint is an end-to-end keypoint detector and descriptor network. Since we are only interested in the performance of descriptors, to give a fair comparison, we use the same keypoints extracted by SuperPoint for all of the descriptors. Matching between two images is done by finding the nearest descriptor. We use precision as our evaluation metric, which is the number of true matches over the number of predicted matches. Table 1 shows the performance of our descriptor compared to other methods. Overall, our method outperforms other traditional handcrafted descriptors by a margin, and gives a favorable performance compared to other learning-based method, *i.e.*, SuperPoint. An example of 2D matching visualization is provided in Figure 2. As can be seen, our descriptor gives a stronger performance with more correct matches.

3D global registration

To demonstrate a practical use of our descriptor, we combine it with RANSAC for the 3D global registration task. Given two 3D fragments from scanning, we uniformly downsample the fragments to obtain the keypoints. For every interest point, we form a local patch by taking points within a neighborhood of $30\,\mathrm{cm}$. The 3D descriptors are then computed for all of these keypoints. We match the two sets of keypoints with

Table 1: 2D matching results (precision) on the SceneNN dataset. Best results are marked in bold.

Frame difference	SIFT	SURF	SuperPoint	PatchNetAE	LCD-D256	LCD-D128	LCD-D64
10	0.252	0.231	0.612	0.613	**0.625**	0.604	0.591
20	0.183	0.157	**0.379**	0.360	0.373	0.364	0.347
30	0.125	0.098	0.266	0.245	**0.267**	0.256	0.239
Average	0.187	0.162	0.419	0.406	**0.422**	0.408	0.392

Table 2: 3D registration results (recall) on the 3DMatch benchmark. Best results are marked in bold.

	CZK	FGR	3DMatch	3DSmoothNet	PointNetAE	LCD-D256	LCD-D128	LCD-D64
Kitchen	0.499	0.305	0.853	0.871	0.766	**0.891**	0.889	0.891
Home 1	0.632	0.434	0.783	**0.896**	0.726	0.783	0.802	0.757
Home 2	0.403	0.283	0.610	**0.723**	0.579	0.629	0.616	0.610
Hotel 1	0.643	0.401	0.786	0.791	0.786	0.808	0.813	**0.841**
Hotel 2	0.667	0.436	0.590	**0.846**	0.680	0.769	0.821	0.821
Hotel 3	0.577	0.385	0.577	**0.731**	0.731	0.654	0.654	0.692
Study	0.547	0.291	0.633	0.556	0.641	**0.662**	0.628	0.650
MIT Lab	0.378	0.200	0.511	0.467	0.511	**0.600**	0.533	0.578
Average	0.543	0.342	0.668	**0.735**	0.677	0.725	0.720	0.730

(a) SIFT

(b) Ours

Figure 2: Qualitative 2D matching comparison between SIFT and our proposed descriptor. Our descriptor can correctly identify features from the wall and the refrigerator, while SIFT (Lowe 2004) fails to differentiate them.

nearest neighbor search and use RANSAC to estimate the final rigid transformation.

We use the 3DMatch Benchmark (Zeng et al. 2017) to evaluate the 3D matching performance of our descriptor, which contains 8 scenes for testing. 3DMatch already provided test fragments fused from consecutive depth frames. However, these fragments lack color information, which is required for our descriptor, so we modified the pipeline to generate another version with color.

We follow the same evaluation process introduced by (Choi, Zhou, and Koltun 2015), using recall as the evaluation metric. Given two non-consecutive scene fragments P_i and P_j, the predicted relative rigid transformation T_{ij} is a true positive if (1) over 30% of $T_{ij}P_i$ overlaps with P_j and (2) T_{ij} is sufficiently close to the ground-truth transformation $\hat{T}_{ij}$. Specifically, T_{ij} is correct if the RMSE of ground-truth correspondences $\hat{\mathcal{K}}_{ij}$ is below a threshold $\tau = 0.2\,\mathrm{m}$:

$$\frac{1}{|\hat{\mathcal{K}}_{ij}|} \sum_{(\hat{p},\hat{q})\in\hat{\mathcal{K}}_{ij}} \|T_{ij}\hat{p} - \hat{q}\|^2 < \tau^2. \qquad (5)$$

Table 2 lists the recall of different algorithms on the 3DMatch benchmark. *CZK* (Choi, Zhou, and Koltun 2015) uses FPFH descriptor (Rusu, Blodow, and Beetz 2009) with RANSAC to prune false matches. *3DMatch* (Zeng et al. 2017) employs the same RANSAC-based pipeline, using their own voxel-based learned descriptor. *FGR* (Zhou, Park, and Koltun 2016) is a fast global registration algorithm which does not rely on iterative sampling. *3DSmoothNet* (Gojcic et al. 2019) is a newly proposed method that uses a voxelized smooth density value representation. *PointNetAE* is just the single 3D branch of our network trained only with the 3D data in the dataset. Overall, our descriptor with RANSAC outperforms others by a significant margin. We also show additional qualitative results in Figure 3.

2D-3D place recognition

We further evaluated our local cross-domain descriptor with 2D-to-3D place recognition. Unlike previous works in single-domain place recognition (Torii et al. 2015; Arandjelovic et al. 2016), our task is to find the corresponding 3D geometry submap in the database given a query 2D image. We only assume that raw geometries are given, without additional
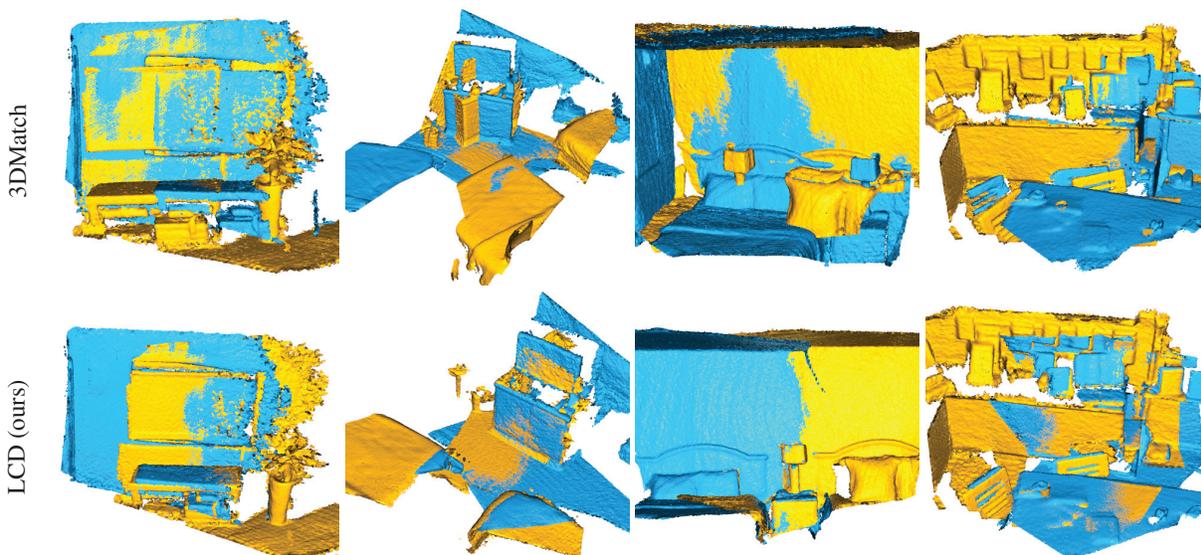
Figure 3: Qualitative results on the 3DMatch benchmark. Our method is able to successfully align pair of fragments in different challenging scenarios by matching local 3D descriptors, while 3DMatch (Zeng et al. 2017) fails in cases when there are ambiguities in geometry.
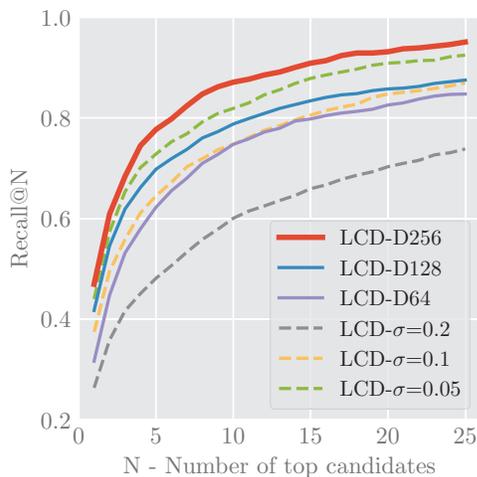
Figure 4: Results of the 2D-3D place recognition task. *LCD-D256*, *LCD-D128*, and *LCD-D64* indicate descriptor with different dimensions. While being effective, our cross-domain descriptor also demonstrates the robustness to input noise, with *LCD-σ* indicating the results when adding Gaussian noise with standard deviation σ into the query images.

associated image descriptors and/or camera information as often seen in the camera localization problem (Zeisl, Sattler, and Pollefeys 2015; Sattler et al. 2015). With increasing availability of 3D data, such 2D-3D place recognition becomes practical as it allows using Internet photos to localize a place in 3D. To the best of our knowledge, there has been no previous report about solving this cross-domain problem.

Here we again use the SceneNN dataset (Hua et al. 2016).

Following the split from (Hua, Tran, and Yeung 2018), we use 20 scenes for evaluation. To generate geometry submaps, we integrate every 100 consecutive RGB-D frames. In total, our database is consisted of $1,191$ submaps from various lighting conditions and settings such as office, kitchen, bedroom, etc. The query images are taken directly from the RGB frames, such that every submap has at least one associated image.

We cast this 2D-3D place recognition problem as an retrieval task. Inspired by the approach from Dense-VLAD (Torii et al. 2015), for each 3D submap, we sample descriptors on a regular voxel grid. These descriptors are then aggregate into a single compact VLAD descriptor (Jégou et al. 2010), using a dictionary of size 64. To extract the descriptor for an image, we follow the same process, but on a 2D grid. The query descriptor are matched with the database to retrieve the final results.

We follow the standard place recognition evaluation procedure (Torii et al. 2015; Arandjelovic et al. 2016). The query image is deemed to be correctly localized if at least one of the top N retrieved database submaps is within $d = 0.5\,\mathrm{m}$ and $\theta = 30°$ from the ground-truth pose of the query.

We plot the fraction of correct queries (recall@N) for different value of N, as shown in Figure 4. Representative top-3 retrieval results are shown in Figure 5. It can be seen that our local cross-domain descriptor are highly effective in this 2D-to-3D retrieval task.

Sparse-to-dense depth estimation

As we have the inverse mapping from the shared latent space back to 2D and 3D due to the use of auto-encoders, we can apply the local cross-domain descriptors for dense depth estimation. Particularly, here we show how to enrich depth from a color image with local predictions from the color channel.
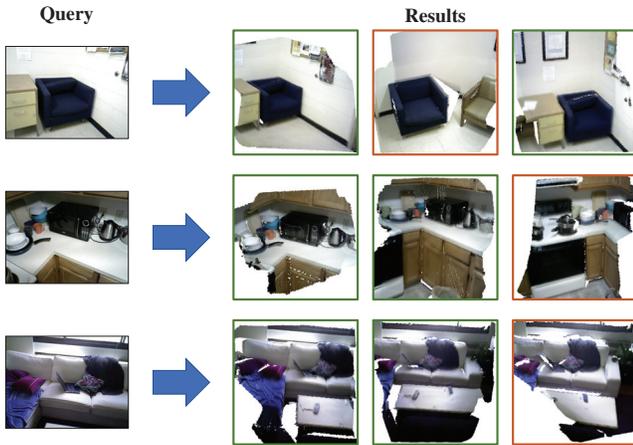
Figure 5: Top-3 retrieval results of the 2D-3D place recognition task using our descriptor. Green/red borders mark correct/incorrect retrieval. Best view in color.

Figure 6: Sparse-to-dense depth estimation results. Inputs are a RGB image and 2048 sparse depth samples. Our network estimates dense depth map by reconstructing local 3D points. Best view in color.

Table 3: Quantitative results of indoor depth estimation on the SceneNN dataset. Our method outperforms conventional depth estimation method FCRN (Laina et al. 2016) by a margin. Note that our method uses both RGB image and sparse depth samples, while FCRN only use RGB image as input. Best results are marked in bold.

	REL	RMSE	δ_1	δ_2	δ_3
FCRN	0.458	0.548	0.467	0.791	**0.935**
LCD-D64	**0.187**	**0.446**	**0.861**	**0.894**	0.908
LCD-D128	0.193	0.458	0.857	0.890	0.903
LCD-D256	0.194	0.459	0.858	0.890	0.903

We perform sparse-to-dense depth estimation, where dense depth is predicted from sparse samples (Mal and Karaman 2018). This can be beneficial in robotics, augmented reality, or 3D mapping, where the resolution of depth sensor is limited but the resolution of the color sensor is very high.

Given an RGB-D image, we first take the RGB and sample uniform 2D patches on a 50×50 regular grid. From these 2D patches, we first encode them using the 2D encoder, and decode using the 3D decoder to reconstruct the local point clouds. We assemble these local point clouds into one coherent point cloud by using the input depth samples. Finally, the dense depth prediction is calculated by projecting the dense 3D point cloud back to the image plane. Figure 6 shows some qualitative results from cross-domain network.

Here we also compare our method against *FCRN* (Laina et al. 2016) on the SceneNN dataset. Note that FCRN only uses RGB images as input, instead of RGB and sparse depth. We use the same $1,191$ RGB-D frames in the 2D-3D place recognition experiment for this evaluation, keeping the same training configuration. Evaluation metrics include: absolute mean relative error (REL), root-mean-square error (RMSE), and percentage of pixels within a threshold (δ_i). We report the evaluation results in Table 3.
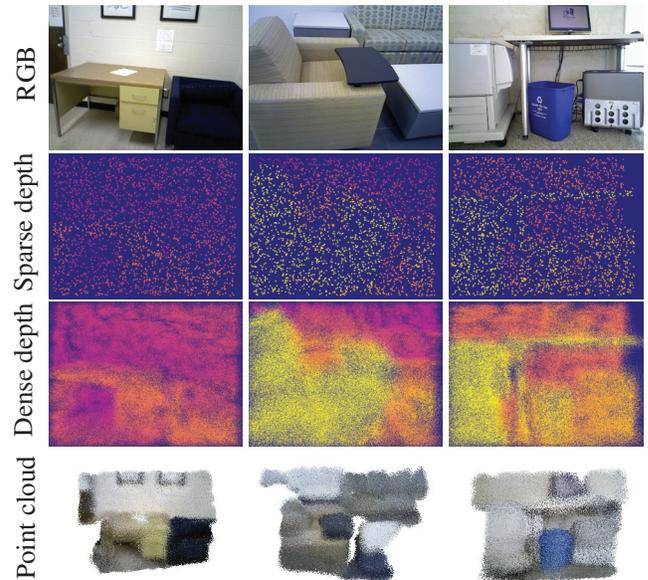
Ablation study

Robustness to noise. Since our descriptor uses color information in the training process, there is a risk that it might be over-fitting to color information, and not robust to the lighting changes in either 2D or 3D domain. To demonstrate the robustness of our method, we simulate color changes by adding Gaussian noise of standard deviation σ to the input image patches. We then run the 2D-3D place recognition experiment again with different levels of noisy input, and compare to the original result. Figure 4 shows the performance of our descriptor under different Gaussian noises, named LCD-σ. With a moderate level of noise ($\sigma = 0.1$), our descriptor still get a very good performance, achieving $\approx 75\%$ recall at 10 top candidates. This study shows that there is no need for the color in 2D patch and 3D point cloud to be identical, and our proposed descriptor is robust to input noise.

Single-domain vs. cross-domain. We also compare our cross-domain descriptor with descriptors trained on single domain. As shown in Table 1 and Table 2, we train two single auto-encoder models on either 2D or 3D domain, denoting as *PatchNetAE* and *PointNetAE*, respectively. Our learned cross-domain descriptor consistently outperforms single-domain models by a margin. This result implies that not only learning cross-domain descriptors benefits downstream applications, but it also improves the performance on single-domain tasks.

Efficiency. Our proposed descriptor is very efficient and can be used in real-time applications. Generating descriptors only requires a forward pass on the decoder, which only takes $3.4\,\text{ms}$ to compute 2048 descriptors, compared to $4.5\,\text{ms}$

when using ORB (Rublee et al. 2011). The network only uses around 1 GB of GPU memory for inference.

Conclusion

We propose a new local cross-domain descriptor that encodes 2D and 3D structures into representations in the same latent space. Our learned descriptor is built based on auto-encoders that are jointly trained to learn domain invariant features and representative and discriminative features of 2D/3D local structures. We found that local cross-domain descriptors are more discriminative than single-domain descriptors. We demonstrated that cross-domain descriptors are effective to 2D-3D matching and retrieval tasks. While our descriptor is task agnostic, we demonstrated that the descriptors are robust when being used for image matching and 3D registration, achieving competitive results with other state-of-the-art methods. A potential future work is to integrate our descriptor with keypoint detectors to make full image matching and retrieval becomes possible.

References

Angelina Uy, M., and Hee Lee, G. 2018. PointNetVLAD: Deep point cloud based retrieval for large-scale place recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Arandjelovic, R.; Gronat, P.; Torii, A.; Pajdla, T.; and Sivic, J. 2016. NetVLAD: CNN architecture for weakly supervised place recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Balntas, V.; Lenc, K.; Vedaldi, A.; and Mikolajczyk, K. 2017. HPatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Bay, H.; Tuytelaars, T.; and Van Gool, L. 2006. SURF: Speeded up robust features. In *European Conference on Computer Vision*.

Brown, M.; Hua, G.; and Winder, S. 2010. Discriminative learning of local image descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Choi, S.; Zhou, Q.-Y.; and Koltun, V. 2015. Robust reconstruction of indoor scenes. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Chopra, S.; Hadsell, R.; and LeCun, Y. 2005. Learning a similarity metric discriminatively with application to face verification. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Dai, A.; Nießner, M.; Zollhöfer, M.; Izadi, S.; and Theobalt, C. 2017. BundleFusion: Real-time globally consistent 3D reconstruction using on-the-fly surface reintegration. *ACM Transactions on Graphics* 36(3):24.

Deng, H.; Birdal, T.; and Ilic, S. 2018a. PPF-FoldNet: Unsupervised learning of rotation invariant 3D local descriptors. In *European Conference on Computer Vision*.

Deng, H.; Birdal, T.; and Ilic, S. 2018b. PPFNet: Global context aware local features for robust 3D point matching. In *IEEE Conference on Computer Vision and Pattern Recognition*.

DeTone, D.; Malisiewicz, T.; and Rabinovich, A. 2018. Super-Point: Self-supervised interest point detection and description. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*.

Dewan, A.; Caselitz, T.; and Burgard, W. 2018. Learning a local feature descriptor for 3D LiDAR scans. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Durrant-Whyte, H., and Bailey, T. 2006. Simultaneous localization and mapping: Part i. *IEEE Robotics & Automation Magazine*.

Feng, M.; Hu, S.; Jr, M. H. A.; and Lee, G. H. 2019. 2D3D-MatchNet: Learning to match keypoints across 2D image and 3D point cloud. In *IEEE International Conference on Robotics and Automation*.

Glocker, B.; Izadi, S.; Shotton, J.; and Criminisi, A. 2013. Real-time RGB-D camera relocalization. In *IEEE International Symposium on Mixed and Augmented Reality*.

Gojcic, Z.; Zhou, C.; Wegner, J. D.; and Wieser, A. 2019. The perfect match: 3D point cloud matching with smoothed densities. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Halber, M., and Funkhouser, T. 2017. Fine-to-coarse global registration of RGB-D scans. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Han, X.; Leung, T.; Jia, Y.; Sukthankar, R.; and Berg, A. C. 2015. MatchNet: Unifying feature and metric learning for patch-based matching. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Haralick, R. M.; Lee, D.; Ottenburg, K.; and Nolle, M. 1991. Analysis and solutions of the three point perspective pose estimation problem. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Hartley, R., and Zisserman, A. 2003. *Multiple view geometry in computer vision*. Cambridge University Press.

Hejrati, M., and Ramanan, D. 2012. Analyzing 3D objects in cluttered images. In *Advances in Neural Information Processing Systems*.

Henry, P.; Fox, D.; Bhowmik, A.; and Mongia, R. 2013. Patch volumes: Segmentation-based consistent mapping with RGB-D cameras. In *International Conference on 3D Vision*.

Hermans, A.; Beyer, L.; and Leibe, B. 2017. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*.

Hua, B.-S.; Pham, Q.-H.; Nguyen, D. T.; Tran, M.-K.; Yu, L.-F.; and Yeung, S.-K. 2016. SceneNN: A scene meshes dataset with annotations. In *International Conference on 3D Vision*.

Hua, B.-S.; Tran, M.-K.; and Yeung, S.-K. 2018. Pointwise convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Jégou, H.; Douze, M.; Schmid, C.; and Pérez, P. 2010. Aggregating local descriptors into a compact image representation. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Keller, M.; Chen, Z.; Maffra, F.; Schmuck, P.; and Chli, M. 2018. Learning deep descriptors with scale-aware triplet networks. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Khoury, M.; Zhou, Q.-Y.; and Koltun, V. 2017. Learning compact geometric features. In *IEEE International Conference on Computer Vision*.

Kumar, B.; Carneiro, G.; Reid, I.; et al. 2016. Learning local image descriptors with deep siamese and triplet convolutional networks by minimising global loss functions. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Laina, I.; Rupprecht, C.; Belagiannis, V.; Tombari, F.; and Navab, N. 2016. Deeper depth prediction with fully convolutional residual networks. In *International Conference on 3D Vision*.

Li, J., and Lee, G. H. 2019. USIP: Unsupervised stable interest point detection from 3D point clouds. In *IEEE International Conference on Computer Vision*.

Li, Y.; Su, H.; Qi, C. R.; Fish, N.; Cohen-Or, D.; and Guibas, L. J. 2015. Joint embeddings of shapes and images via cnn image purification. *ACM Transactions on Graphics* 34(6):234.

Lim, J. J.; Pirsiavash, H.; and Torralba, A. 2013. Parsing IKEA objects: Fine pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Lowe, D. G. 2004. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* 60(2):91–110.

Mal, F., and Karaman, S. 2018. Sparse-to-dense: Depth prediction from sparse depth samples and a single image. In *IEEE International Conference on Robotics and Automation*.

Mishchuk, A.; Mishkin, D.; Radenovic, F.; and Matas, J. 2017. Working hard to know your neighbor's margins: Local descriptor learning loss. In *Advances in Neural Information Processing Systems*.

Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017. PointNet: Deep learning on point sets for 3D classification and segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Rublee, E.; Rabaud, V.; Konolige, K.; and Bradski, G. R. 2011. ORB: An efficient alternative to SIFT or SURF. In *IEEE International Conference on Computer Vision*.

Rusu, R. B.; Blodow, N.; Marton, Z. C.; and Beetz, M. 2008. Aligning point cloud views using persistent feature histograms. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Rusu, R. B.; Blodow, N.; and Beetz, M. 2009. Fast point feature histograms (FPFH) for 3D registration. In *IEEE International Conference on Robotics and Automation*.

Sattler, T.; Havlena, M.; Radenovic, F.; Schindler, K.; and Pollefeys, M. 2015. Hyperpoints and fine vocabularies for large-scale location recognition. In *IEEE International Conference on Computer Vision*.

Sattler, T.; Torii, A.; Sivic, J.; Pollefeys, M.; Taira, H.; Okutomi, M.; and Pajdla, T. 2017. Are large-scale 3D models really necessary for accurate visual localization? In *IEEE Conference on Computer Vision and Pattern Recognition*.

Schroff, F.; Kalenichenko, D.; and Philbin, J. 2015. FaceNet: A unified embedding for face recognition and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Simo-Serra, E.; Trulls, E.; Ferraz, L.; Kokkinos, I.; Fua, P.; and Moreno-Noguer, F. 2015. Discriminative learning of deep convolutional feature point descriptors. In *IEEE International Conference on Computer Vision*.

Suwajanakorn, S.; Snavely, N.; Tompson, J. J.; and Norouzi, M. 2018. Discovery of latent 3D keypoints via end-to-end geometric reasoning. In *Advances in Neural Information Processing Systems*.

Tian, Y.; Fan, B.; and Wu, F. 2017. L2-Net: Deep learning of discriminative patch descriptor in Euclidean space. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Tola, E.; Lepetit, V.; and Fua, P. 2009. DAISY: An efficient dense descriptor applied to wide-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Tombari, F.; Salti, S.; and Di Stefano, L. 2010. Unique signatures of histograms for local surface description. In *European Conference on Computer Vision*.

Torii, A.; Arandjelovic, R.; Sivic, J.; Okutomi, M.; and Pajdla, T. 2015. 24/7 place recognition by view synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Valentin, J.; Dai, A.; Nießner, M.; Kohli, P.; Torr, P.; Izadi, S.; and Keskin, C. 2016. Learning to navigate the energy landscape. In *International Conference on 3D Vision*.

Xiao, J.; Owens, A.; and Torralba, A. 2013. SUN3D: A database of big spaces reconstructed using SfM and object labels. In *IEEE International Conference on Computer Vision*.

Xiao, J.; Russell, B.; and Torralba, A. 2012. Localizing 3D cuboids in single-view images. In *Advances in Neural Information Processing Systems*.

Xing, X.; Cai, Y.; Lu, T.; Cai, S.; Yang, Y.; and Wen, D. 2018. 3DT-Net: Learning local features using 2D and 3D cues. In *International Conference on 3D Vision*.

Yew, Z. J., and Lee, G. H. 2018. 3DFeat-Net: Weakly supervised local 3D features for point cloud registration. In *European Conference on Computer Vision*.

Yi, K. M.; Trulls, E.; Lepetit, V.; and Fua, P. 2016. LIFT: Learned invariant feature transform. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Zagoruyko, S., and Komodakis, N. 2015. Learning to compare image patches via convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Zeisl, B.; Sattler, T.; and Pollefeys, M. 2015. Camera pose voting for large-scale image-based localization. In *IEEE International Conference on Computer Vision*.

Zeng, A.; Song, S.; Nießner, M.; Fisher, M.; Xiao, J.; and Funkhouser, T. 2017. 3DMatch: Learning local geometric descriptors from RGB-D reconstructions. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Zhou, Q.-Y.; Park, J.; and Koltun, V. 2016. Fast global registration. In *European Conference on Computer Vision*.