Performance Evaluation of Learned 3D Features

Riccardo Spezialetti, Samuele Salti, and Luigi Di Stefano
Viale del Risorgimento 2, Bologna, Italy
{riccardo.spezialetti,samuele.salti,luigi.distefano}@unibo.it

Abstract. Matching surfaces is a challenging 3D Computer Vision problem typically addressed by local features. Although a variety of 3D feature detectors and descriptors has been proposed in literature, they have seldom been proposed together and it is yet not clear how to identify the most effective detector-descriptor pair for a specific application. A promising solution is to leverage machine learning to learn the optimal 3D detector for any given 3D descriptor [15]. In this paper, we report a performance evaluation of the detector-descriptor pairs obtained by learning a paired 3D detector for the most popular 3D descriptors. In particular, we address experimental settings dealing with object recognition and surface registration.

Keywords: 3D Computer Vision · Surface Matching · 3D Features.

1 Introduction

Surface matching is an ubiquitous task in 3D Computer Vision, where it helps to tackle major applications such as object recognition and surface registration. Nowadays, most surface matching methods follow a local paradigm based on establishing correspondences between 3D patches referred to as features. The typical feature-matching pipeline consists of three steps: detection, description and matching.

Although over the last decades many 3D detectors and descriptors have been proposed in literature, it is yet unclear how to effectively combine these proposals to create an effective pipeline. Indeed, unlike the related field of local image features, methods to either detect or describe 3D features have been designed and proposed separately, alongside with specific application settings and related datasets. This is also vouched by the main performance evaluation papers in the field, which address either repeatability of 3D detectors designed to highlight geometrically salient surface patches [14] or distinctiveness and robustness of popular 3D descriptors [2].

More recently, however, [9] and [15] have proposed a machine learning approach that allows for learning an optimal 3D keypoint detector for any given 3D descriptor so as to maximize the end-to-end performance of the overall feature-matching pipeline. The authors show that this approach provides effective pipelines across diverse applications and datasets. Moreover, their object recognition experiments show that, with the considered descriptors (SHOT [13], Spin Image (SI) [4], FPFH [8]), learning to detect specific keypoints leads to better performance than relying on existing general-purpose handcrafted detectors (ISS [17], Harris3D [10], NARF [11]).

By enabling an optimal detector to be learned for any descriptor, [15] sets forth a novel paradigm to maximize affinity between 3D detectors and descriptors. This opens
up the question of which learned detector-descriptor pair may turn out most effective in the main application areas. This paper tries to answer this question by proposing an experimental evaluation of learned 3D pipelines. In particular, we address object recognition and surface registration, and compare the performance attained by learning a paired feature detector for the most popular handcrafted 3D descriptors (SHOT [13], SI [4], FPFH [8], USC [12], RoPS [3]) as well as for a recently proposed descriptor based on deep learning (CGF-32 [5]).

2 3D Local Feature Detectors and Descriptors

This section reviews state-of-the-art methods for detection and description of 3D local features. Both tasks have been pursued through hand-crafted and learned approaches.

Hand-Crafted Feature Detectors  Keypoint detectors have traditionally been conceived to identify points that maximize a saliency function computed on a surrounding patch. The purpose of this function is to highlight those local geometries that turn out repeatedly identifiable in presence of nuisances such as noise, viewpoint changes, point density variations and clutter. State-of-the-art proposals mainly differ for the adopted saliency function. Detectors operate in two steps: first, the saliency function is computed at each point on the surface, then non-maxima suppression allows for sifting out saliency peaks. Intrinsic Shape Signature (ISS) [17] computes the eigenvalue decomposition of the scatter-matrix of the points within the supporting patch in order to highlight local geometries exhibiting a prominent principal direction, Harris3D [10] extends the idea of image corners by deploying surface normals rather than image gradients to calculate the saliency (i.e Cornerness) function. Normal Aligned Radial Feature (NARF) [11] first selects stable surface points, then highlights those stable points showing sufficient local variations. This leads to locate keypoints close to depth discontinuities.

Learned Feature Detectors  Unlike previous work in the field, Salti et al. [9] proposed to learn a keypoint detector amenable to identify points likely to generate correct matches when encoded by the SHOT descriptor. In particular, the authors cast keypoint detection as a binary classification problem tackled by a Random Forest and show how to generate the training set as well as the feature representation deployed by the classifier. Later, Tonioni et al. [15] have demonstrated that this approach can be applied seamlessly and very effectively to other popular descriptors such as SI [4] and FPFH [8].

Hand-Crafted Feature Descriptors  Many hand-crafted feature descriptors represent the local surface by computing geometric measurements within the supporting patch and then accumulating values into histograms. Spin Images (SI) [4] relies on two coordinates to represent each point in the support: the radial coordinate, defined as the perpendicular distance to the line trough the surface normal at the keypoint, and the elevation coordinate, defined as the signed distance to the tangent plane at the keypoint. The space formed by this two values is then discretized into a 2D histogram.

In 3D Shape Context (3DSC) [11] the support is partitioned by a 3D spherical grid centered at the keypoint with the north pole aligned to the surface normal. A 3D his-
togram is built by counting up the weighted number of points falling into each spatial bin along the radial, azimuth and elevation dimensions. Unique Shape Context (USC) [12] extends 3DSC with the introduction of a unique and repeatable canonical reference frame borrowed from [13].

SHOT [13], alike, deploys both a unique and repeatable canonical reference frame as well as a 3D spherical grid to discretize the supporting patch into bins along the radial, azimuth and elevation axes. Then, the angles between the normal at the keypoint and those at the neighboring points within each bins are accumulated into local histograms. Rotational Projection Statistics (RoPS) [3] uses a canonical reference frame to rotate the neighboring points on the local surface. The descriptor is then constructed by rotationally projecting the 3D points onto 2D planes to generate three distribution matrices. Finally, a histogram encoding five statistics of distribution matrices is calculated. Fast Point Feature Histograms (FPFH) [8] operates in two steps. In the first, akin to PFH [7], four features, refereed to as SPFH, are calculated using the Darboux frame and the surface normals between the keypoint and its neighbors. In the second step, the descriptor is obtained as the weighted sum between the SPFH of the keypoint and the SPFHs of the neighboring points.

Learned Feature Descriptors The success of deep neural networks in so many challenging image recognition tasks has motivated research on learning representations from 3D data. One of the pioneering works is 3D Match [16], where the authors deploy a siamese network trained on local volumetric patches to learn a local descriptor. The input to the network consists of a Truncated Signed Distance Function (TSDF) defined on a voxel grid. In [5], the authors deploy a fully-connected deep neural network together with a feature learning approach based on the triplet ranking loss in order to learn a very compact 3D descriptor, referred to as CGF-32. Their approach does not rely on raw data but on an hand-crafted input representation similar to [1], canonicalized by the local reference frame presented in [13].

3 Keypoint Learning

In order to carry out the performance evaluation proposed in this paper, for most local descriptors reviewed in section 2 we did learn the corresponding optimal detector according to the keypoint learning methodology [15]. We provide here a brief overview of this methodology and refer the reader to [9,15] for a detailed description.

The idea behind keypoint learning is to learn to detect keypoints that can yield good correspondences when coupled with a given descriptor. To this end, keypoint detection is cast as binary classification, i.e. a point can either be a good candidate or not when used to create matches by means of the given descriptor, and a Random Forest is used as classifier. Training of the classifier requires to define the training set, i.e. both positive (good) and negative (not good) points, as well as the feature representation.

As for positive samples, the method tries to sift out those points that, when described by a chosen descriptor, can be matched correctly across different 2.5D views of a 3D object. Thus, starting from a set of 2.5D views \( \{ V_i \}, i = 1, \ldots, N \) of an object from a 3D dataset, each point \( p \in V_i \) in each view \( V_i \) is embedded by the chosen descriptor. Then, for each view \( V_i \), a subset of overlapping views is selected based on an
overlap threshold $\tau$. A two-step positive samples selection is performed on $V_i$ and each overlapping view $V_j$. In the first step, a list of correspondences between descriptors is created by searching for all descriptors $d \in V_i$ the nearest neighbor in the descriptor space between all descriptors $g \in V_j$. A preliminary list of positive samples $P^j_i$ for view $V_i$ is created by taking only those points that have been correctly matched in $V_j$, i.e., the points belonging to the matched descriptors in the two views correspond to the same 3D point of the object according to threshold $\epsilon$. The list is then filtered removing non-maxima local extrema within $\epsilon_{nms}$ using the descriptor distance as saliency. In the second step, the list of positive samples is refined by keeping only the points in $V_i$ that can be matched correctly also in those others overlapping views that have not been used in the first step. Negative samples are then extracted on each view, sampling random points among those points which are not included in the positive set. A distance threshold $\epsilon_{neg}$ is used to avoid a negative being too close to a positive and to other negative samples, and also to balance the size of the positive and negative sets.

As far as the representation input to the classifier is concerned, the method relies on histograms of normal orientations inspired by SHOT [13]. However, to avoid computation of the local Reference Frame while still achieving rotation invariance, the spherical support is divided only along the radial dimension so as to compute a histogram for each spherical shell thus obtained. [15] showed that, although inspired by SHOT, such representation can be used to learn an effective detector also for other descriptors.

4 Evaluation Methodology

The performance evaluation proposed in this paper aims to compare different learned detector-descriptor pairs while addressing two main application settings, namely object recognition and surface registration. In this section, we highlight the key traits and nuisances which characterize the two tasks, present the datasets and performance evaluation metrics used in the experiments and, finally, provide the relevant implementation details.

4.1 Object Recognition

In typical object recognition settings, one wishes to recognize a set of given 3D models into scenes acquired from an unknown vantage point and featuring an unknown arrangement of such models. Peculiar nuisances in this scenario are occlusions and, possibly, clutter, as objects not belonging to the model gallery may be present in the scenes. In our experiments we rely on the following popular object recognition datasets:

- **UWA** dataset, introduced by Mian et al. [6]. This dataset consists of 4 full 3D models and 50 scenes wherein models significantly occlude each other. To create some clutter, scenes contain also an object which is not included in the model gallery. As scenes are scanned by a Minolta Vivid 910 scanner, they are corrupted by real sensor noise.

- **Random Views** dataset, based on the Stanford 3D scanning repository[^1] and originally proposed in [14]. This dataset comprises 6 full 3D models and 36 scenes

[^1]: [http://graphics.stanford.edu/data/3Dscanrep/](http://graphics.stanford.edu/data/3Dscanrep/)
obtained by synthetic renderings of random model arrangements. Scenes feature occlusions but no clutter. Moreover, scenes are corrupted by different levels of synthetic noise. In the experiments we consider scenes with Gaussian noise equal to \( \sigma = 0.1 \) mesh resolution units.

To evaluate the effectiveness of the considered learned detector-descriptor pairs we rely on descriptor matching experiments. Specifically, for both datasets, we run keypoint detection on synthetically rendered views of all models. Then, we compute and store into a single kd-tree all the corresponding descriptors. Keypoints are detected and described also in the set of scenes provided with the dataset, \( \{S_j\}, j = 1, \ldots, N_S \). Eventually, a correspondence is established for each scene descriptor by finding the nearest neighbor descriptor within the models kd-tree and thresholding the distance between descriptors to accept a match as valid. Correct correspondences can be identified based on knowledge of the ground-truth transformations which bring views and scenes into a common reference frame and checking whether the matched keypoints lay within a 3D distance \( \epsilon \). Indeed, denoting as \((k_j, k_{n,m})\) a correspondence between a keypoint \(k_j\) detected in scene \(S_j\) and a keypoint \(k_{n,m}\) detected in the \(n\)-th view of model \(m\), as \(T_{j,m}\) the transformation from \(S_j\) to model \(m\), as \(T_{n,m}\) the transformation from the \(n\)-th view and the canonical reference frame of model \(m\), the set of correct correspondences for scene \(S_j\) is given by:

\[
C_j = \{(k_j, k_{n,m}) : \|T_{j,m}k_j - T_{n,m}k_{n,m}\| \leq \epsilon\}
\]

From \(C_j\), we can compute True Positive and False Positive matches for each scene and, by averaging them across scenes, for each of the considered datasets. The final results for each dataset are provided as Recall vs. 1-Precision curves, with curves obtained by varying the threshold on the distance between descriptors.

### 4.2 Surface Registration

The goal of surface registration is to align into a common 3D reference frame several partial views (usually referred to as scans) of a 3D object obtained by a certain optical sensor. This is achieved through rather complex procedures that, however, typically rely on a key initial step, referred to as Pairwise Registration, aimed at estimating the rigid motion between any two views by a feature-matching pipeline. Differently from object recognition scenarios, the main nuisances deal with missing regions, self-occlusions, limited overlap area between views and point density variations. In our experiments we rely on the following surface registration dataset:

- **Laser Scan** dataset, recently proposed in [5]. This dataset includes 8 public-domain 3D models, i.e. 3 taken from the AIM@SHAPE repository (*Bimba, Dancing Children* and *Chinese Dragon*), 4 from the Stanford 3D Scanning Repository (*Armadillo, Buddha, Bunny, Stanford Dragon*) and *Berkeley Angel* According to the protocol described in [5], training should be carried out based on synthetic views generated from *Berkeley Angel, Bimba, Bunny and Chinese Dragon*, while the test data consists of the the real scans available for the remaining 3 models (*Armadillo, Buddha* and *Stanford Dragon*).
Thus, given a set of $M$ real scans available for a test model, we compute all the possible $N = \frac{M(M-1)}{2}$ view pairs $\{V_i, V_j\}$. For each pair, we run keypoint detection on both views. Due to partial overlap between the views, a keypoint belonging to $V_i$ may have no correspondence in $V_j$. Hence, denoted as $T_i$ and $T_j$ the ground-truth transformations that, respectively, bring $V_i$ and $V_j$ into a canonical reference frame, we can compute the set $O_{i,j}$ that contains the keypoints in $V_i$ that have a corresponding point in $V_j$. In particular, given a keypoint $k_i \in V_i$

$$O_{i,j} = \{k_i : \|T_i k_i - N(N(T_i k_i, T_j V_j))\| \leq \epsilon_{ovr}\},$$  

(2)

where $N(N(T_i k_i, T_j V_j))$ denotes the nearest neighbor of $T_i k_i$ in the transformed view $T_j V_j$. If the number of points in $O_{i,j}$ is less than 20% of the keypoints in $V_i$, the pair $(V_i, V_j)$ is not considered in the evaluation experiments due to insufficient overlap. Conversely, for all the view pairs $(V_i, V_j)$ exhibiting sufficient overlap, a list of correspondences between all the keypoints detected in $V_i$ and all the keypoints extracted from $V_j$ is established by finding the nearest neighbor in the descriptor space via kdtree matching. Then, given a pair of matched keypoints $(k_i, k_j)$, $k_i \in V_i, k_j \in V_j$, the set of correct correspondences, $C_{i,j}$, can be identified based on the available ground-truth transformations by checking whether the matched keypoints lay within a certain distance $\epsilon$ in the canonical reference frame:

$$C_{i,j} = \{(k_i, k_j) : \|T_i k_i - T_j k_j\| \leq \epsilon\}$$  

(3)

Then, the precision of the matching process can be computed as a function of the distance threshold $\epsilon$ [5]:

$$\text{precision}_{i,j}(\epsilon) = \frac{|C_{i,j}|}{|O_{i,j}|}$$  

(4)

The precision score associated with any given model is obtained by averaging across all view pairs. We also average across all test models so as to get the final score associated to the Laser Scan dataset.

| Dataset          | $r_{desc}$ (mm) | $r_{det}$ (mm) | $\tau$ | $\epsilon$ (mm) | $r_{nms}$ (mm) | $r_{neg}$ (mm) | $r_{nms}$ (mm) | $s_{min}$ (mm) |
|------------------|-----------------|----------------|--------|-----------------|----------------|----------------|----------------|----------------|
| UWA              | 40              | 20             | 0.85   | 7               | 4              | 2              | 4              | 0.8            |
| Random Views     | 40              | 20             |        | 7               | -              | -              | 4              | 0.8            |

4.3 Implementation

For all handcrafted descriptors considered in our evaluation, we use the implementation available in the PCL library. For CGF-32, we use the public implementation made available by the authors [5]. As for the Keypoint Learning (KPL) framework described in section 3, we use the publicly available original code for the generation of the training set [2]. During the detection phase, each point of a point cloud is passed through the

2 http://github.com/CVLAB-Unibo/Keypoint-Learning.
Table 2: Parameters for surface registration dataset.

| Model Name       | $r_{desc}$ (mm) | $r_{det}$ (mm) | $\tau$ (mm) | $\epsilon_{nms}$ (mm) | $\epsilon_{neg}$ (mm) | $\epsilon_{ovr}$ (mm) | $s_{min}$ (mm) |
|------------------|-----------------|----------------|-------------|------------------------|------------------------|----------------------|---------------|
| Angel            | 40              | 20             | 0.85        | 7                      | 4                      | 2                    | -             |
| Bimba            | 40              | 20             | 0.85        | 7                      | 4                      | 2                    | -             |
| Bunny            | 40              | 20             | 0.65        | 7                      | 4                      | 2                    | -             |
| Chinese Dragon   | 40              | 20             | 0.65        | 7                      | 4                      | 2                    | -             |
| Armadillo        | 40              | 20             | -           | 7                      | -                      | -                    | 2             | 4             | 0.5           |
| Buddha           | 40              | 20             | -           | 7                      | -                      | -                    | 2             | 4             | 0.5           |
| Stanford Dragon  | 40              | 20             | -           | 7                      | -                      | -                    | 2             | 4             | 0.5           |

Random Forest classifier which produces a score. A point is identified as a keypoint if it exhibits a local maximum of the scores in a neighborhood of radius $r_{nms}$ and the score is higher than a threshold $s_{min}$. For each descriptor considered in our evaluation, we train its paired detector according to the KPL framework. As a result, we obtain six detector-descriptor pairs, referred to from now on as KPL-CGF32, KPL-FPFH, KPL-ROPS, KPL-SHOT, KPL-SI, KPL-USC.

In object recognition experiments, the training data for all detectors are generated from the 4 full 3D models present in the UWA dataset. According to the KPL methodology [9,15], for each model we render views from the nodes of an icosahedron centered at the centroid.

Then, the detectors are used in the scenes of the UWA dataset as well as in those of the Random Views dataset. Thus, similarly to [9,15], we do not retrain the detectors on Random Views in order to test the ability of the considered detector-descriptor pairs to generalize well to unseen models in object recognition settings. A coherent approach was pursued for the CGF-32 descriptor. As the authors do not provide a model trained on the UWA dataset, we trained the descriptor on the synthetically rendered views of the 4 UWA models using the code provided by the authors and following the protocol described in the paper in order to generate the data needed by their learning framework based on the triplet ranking loss. Thus, KPL-CGF32 was trained on UWA models and, like all other detector-descriptor pairs, tested on both UWA and Random Views scenes.

In surface registration experiments we proceed according to the protocol proposed in [5]. Hence, detectors are trained with rendered views of the train models provided within the Laser Scanner dataset (Angel, Bimba, Bunny, Chinese Dragon) and tested on the real scans of the test models (Armadillo, Buddha, Stanford Dragon). As CGF-32 was trained exactly on the abovementioned train models [5], to carry out surface registration experiments we did not retrain the descriptor but used the trained network published by the authors.

The values of the main parameters of the detector-descriptor pairs used in the experiments are summarized in Table 1 and Table 2. As it can be observed from Table 1 train parameters for Random Views dataset are not specified as we did not train KPL detectors on this dataset. For surface registration, since models belong to different repositories, we report parameters grouped by model. Test parameters for Angel, Bimba, Bunny and Chinese Dragon are not reported as they are only used in train. Similarly, we omit train parameters for Armadillo, Buddha and Stanford Dragon. Due to the different shapes

3 https://github.com/marckhoury/CGF
of the models in the dataset, \( \tau \) is tuned during the train stage so that the number of overlapping views remains constant across all models.

5 Experimental Results

5.1 Object Recognition

Results on the UWA dataset are shown in Figure 1. First, we wish to highlight how the features based on descriptors which encode just the spatial densities of points around a keypoint outperform those relying on higher order geometrical attributes (such as, \( e.g. \), normals). Indeed, KPL-CGF32, KPL-USC and KPL-SI yield significantly better results than KPL-SHOT and KPL-FPFH. These results are coherent with the findings and analysis reported in the performance evaluation by Guo et al.\[2\], which pointed out the former feature category being more robust to clutter and sensor noise. It is also worth observing how the representation based on the spatial tessellation and point density measurements proposed in \[1\] together with the local reference frame proposed in \[13\] turn out particularly amenable to object recognition, as it is actually deployed by both features yielding neatly the best performance, namely KPL-CGF32 and KPL-USC. Yet, learning a dataset-specific non-linear mapping by a deep neural network on top of this good representation does improve performance quite a lot, as vouched by KPL-CGF32 outperforming KPL-USC by a large margin. Indeed, the results obtained in this paper by learning both a dataset-specific descriptor as well as its paired optional detector, \( i.e. \) the features referred to as KPL-CGF32, turn out significantly superior to those previously published on UWA object recognition dataset (see \[9\] and \[15\]).

In \[15\], the results achieved on Random Views by the detectors trained on UWA prove the ability of the KPL methodology to learn to detect general rather than dataset-specific local shapes amenable to provide good matches alongside with the paired descriptor, and even more effectively, in fact, than the shapes found by handcrafted detectors. Thus, when comparing the different features, we can assume here that descriptors are feed by detectors with optimal patches and focus on the ability of the former to handle the specific nuisances of the Random Views dataset. As shown in Figure 1, KPL-FPFH and KPL-SHOT perform slightly better than KPL-USC, KPL-CGF32 and KPL-SI. Again, this is coherent with previous findings reported in literature (see \[2\] and \[15\]), which show how descriptors based on higher order geometrical attributes turn out more effective on Random Views due to the lack of clutter and real sensor noise. As for KPL-CGF32, although it performs still overall better than the other descriptors based on point densities, we observe quite a remarkable performance drop compared to the results on the UWA dataset, much larger, indeed, than that observed for KPL-USC, which shares with KPL-CGF32 a very similar input representation. This suggests that the non-linear mapping learned by KPL-CGF32 is highly optimized to tell apart the features belonging to the objects present in the training dataset (\( i.e. \) UWA) but turns out quite less effective when applied to unseen features, like those found on the objects belonging to Random Views. This \textit{domain shift} issue is a peculiar wick trait of learned features, which may cause them to yield less stable performance across diverse datasets than handcrafted representations.
5.2 Surface Registration

First, it is worth pointing out how, unlike in object recognition settings, in surface registration it is never possible to train any machine learning operator, either detector or descriptor, on the very same objects that would then be processed at test time. Indeed, should one be given either a full 3D model or a set of scans where ground-truth transformations are known, as required to train 3D feature detectors (i.e. KPL) or descriptors (e.g. CGF-32), there would be no need to carry out any registration for that object. Surface registration is about stitching together several scans of a new object than one wishes to acquire as a full 3D model. As such, any learning machinery is inherently prone to the domain shift issue.

As mentioned in subsection 4.2, our experiments rely on the Laser Scan dataset [5] and follow the split into train and test objects proposed by the authors. As shown in Figure 2, when averaging across all test objects, the detector-descriptor pair based on the learned descriptor CGF-32 provides the best performance. This validates the findings reported in [5], where the authors introduce CGF-32 and prove its good registration performance on Laser Scan, also in our experimental setting where an optimal detector is learned for every descriptor.

6 Conclusion and Future Work

Object recognition settings turn out quite amenable to deploy learned 3D features. Indeed, one can train upon a set of 3D objects available beforehand, e.g. due to scanning by some sensor or as CAD models, and then seek to recognize them into scenes featuring occlusions and clutter. These settings allow for learning an highly specialized descriptor alongside its optimal paired detector so to achieve excellent performance. In particular, the learned pair referred to in this paper as KPL-CGF32 sets the new state of the art in descriptor matching on the UWA benchmark dataset. Although the learned representation may not exhibit comparable performance when transferred to unseen
objects, in object recognition it is always possible to retrain on the objects at hand to improve performance. An open question left to future work concerns whether the input parametrization deployed by CGF-32 may enable to learn an highly effective non-linear mapping also in datasets characterized by different nuisances (e.g. Laser Scan) or one should better try to learn 3D representations directly from raw data, as vouched by the success of deep learning from image recognition. Features based on learned representations, such as KPL-CGF32, are quite effective also in surface registration, although this scenario is inherently more prone to the domain shift issue and, indeed, features based on handcrafted descriptors, like in particular KPL-SHOT and KPL-USC, turn out very competitive.

We believe that these findings may pave the way for further research on the recent field of learned 3D representations, in particular in order to foster addressing domain adaptation issues, a topic investigated more and more intensively in nowadays deep learning literature concerned with image recognition. Indeed, 3D data are remarkably diverse in nature due to the variety of sensing principles and related technologies and we witness a lack of large training datasets, e.g. at a scale somehow comparable to ImageNet, that may allow learning representations from a rich and varied corpus of 3D models. Therefore, how to effectively transfer learned representations to new scenarios seems a key issue to the success of machine/deep learning in the most challenging 3D Computer Vision tasks.

Finally, KPL has established a new framework whereby one can learn an optimal detector for any given descriptor. In this paper we have shown how applying KPL to a learned representation (CGF-32) leads to particularly effective features (KPL-CGF32), in particular when pursuing object recognition. Yet, according to the KPL methodology, the descriptor (e.g. CGF-32) has to be learned before its paired detector: one might be willing to investigate on whether and how a single end-to-end paradigm may allow learning both component jointly so as to further improve performance.
References

1. Frome, A., Huber, D., Kolluri, R., Bülow, T., Malik, J.: Recognizing objects in range data using regional point descriptors. In: European conference on computer vision. pp. 224–237. Springer (2004)
2. Guo, Y., Bennamoun, M., Sohel, F., Lu, M., Wan, J., Kwok, N.M.: A comprehensive performance evaluation of 3d local feature descriptors. International Journal of Computer Vision 116(1), 66–89 (2016)
3. Guo, Y., Sohel, F., Bennamoun, M., Lu, M., Wan, J.: Rotational projection statistics for 3d local surface description and object recognition. International journal of computer vision 105(1), 63–86 (2013)
4. Johnson, A.E., Hebert, M.: Using spin images for efficient object recognition in cluttered 3d scenes. IEEE Transactions on pattern analysis and machine intelligence 21(5), 433–449 (1999)
5. Khoury, M., Zhou, Q.Y., Koltun, V.: Learning compact geometric features. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition. pp. 153–61 (2017)
6. Mian, A., Bennamoun, M., Owens, R.: On the repeatability and quality of keypoints for local feature-based 3d object retrieval from cluttered scenes. International Journal of Computer Vision 89(2-3), 348–361 (2010)
7. Rusu, R.B., Blodow, N., Marton, Z.C., Beetz, M.: Aligning point cloud views using persistent feature histograms. In: Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on. pp. 3384–3391. IEEE (2008)
8. Rusu, R.B., Bradski, G., Thibaux, R., Hsu, J.: Fast 3d recognition and pose using the viewpoint feature histogram. In: Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on. pp. 2155–2162. IEEE (2010)
9. Salti, S., Tombari, F., Spezialetti, R., Di Stefano, L.: Learning a descriptor-specific 3d keypoint detector. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2318–2326 (2015)
10. Sipiran, I., Bustos, B.: Harris 3d: a robust extension of the harris operator for interest point detection on 3d meshes. The Visual Computer 27(11), 963 (2011)
11. Steder, B., Rusu, R.B., Konolige, K., Burgard, W.: Narf: 3d range image features for object recognition. In: Workshop on Defining and Solving Realistic Perception Problems in Personal Robotics at the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS). vol. 44 (2010)
12. Tombari, F., Salti, S., Di Stefano, L.: Unique shape context for 3d data description. In: Proceedings of the ACM workshop on 3D object retrieval. pp. 57–62. ACM (2010)
13. Tombari, F., Salti, S., Di Stefano, L.: Unique signatures of histograms for local surface description. In: Proceedings of the 11th European Conference on Computer Vision Conference on Computer Vision: Part III. pp. 356–369. ECCV’10. Springer-Verlag, Berlin, Heidelberg (2010), http://dl.acm.org/citation.cfm?id=1927006.1927035
14. Tombari, F., Salti, S., Di Stefano, L.: Performance evaluation of 3d keypoint detectors. International Journal of Computer Vision 102(1-3), 198–220 (2013)
15. Tonioni, A., Salti, S., Tombari, F., Spezialetti, R., Di Stefano, L.: Learning to detect good 3d keypoints. International Journal of Computer Vision pp. 1–20 (2018)
16. Zeng, A., Song, S., Nießner, M., Fisher, M., Xiao, J., Funkhouser, T.: 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In: Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on. pp. 199–208. IEEE (2017)
17. Zhong, Y.: Intrinsic shape signatures: A shape descriptor for 3d object recognition. In: Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on. pp. 689–696. IEEE (2009)