SKD: Unsupervised Keypoint Detecting for Point Clouds using Embedded Saliency Estimation

Georgi Tinchev Adrian Penate-Sanchez Maurice Fallon
Oxford Robotics Institute, University of Oxford
{gtinchev,adrian,mfallon}@robots.ox.ac.uk

December 12, 2019

Abstract

In this work we present a novel keypoint detector that uses saliency to determine the best candidates from point clouds. The approach can be applied to any differentiable deep learning descriptor by using the gradients of that descriptor with respect to the input to estimate an initial set of candidate keypoints. By using a neural network over the set of candidates we further learn to refine the point selection until the actual keypoints are obtained. The key intuition behind this approach is that keypoints need to be determined based on how the descriptor behaves and not just on the geometry that surrounds a point. To improve the performance of the learned keypoint descriptor we combine the saliency, the feature signal and geometric information from the point cloud to allow the network to select good keypoint candidates. The approach was evaluated on the two largest LiDAR datasets - the Oxford RobotCar dataset and the KITTI dataset, where we obtain up to 50% improvement over the state-of-the-art in both matchability score and repeatability.

1. Introduction

A key task for localization and reconstruction is the repeatable extraction of points that can be reliably matched with a map or another representation of the same environment or object. Image keypoint extraction has been well studied, but point cloud keypoint extraction is less explored. In this work, we present a novel method for keypoint extraction from point clouds that exploits the spatial information encoded within each feature to suggest points with higher likelihood of matching, improving the performance of simultaneous keypoint and feature pair.

Traditionally, keypoint extraction methods have only focused on the local geometry of the image, such as the Harris corner detector [9] or the more efficient Difference-of-Gaussians approach which is used in SIFT [18]. These methods focused on understanding the geometry of the image so as to match points present in both images, known as repeatability. If corresponding points are obtained from two different images of the same scene, one can now find matches between them without introducing incorrect matches (outliers). If the outlier percentage is low, the robust estimator [6] can converge faster and produce a better solution. The problem with this approach is that it ignores how the feature will react to disturbances. The fact that a point is the same does not mean that the descriptor will produce the same response. Therefore, one has to look at the combination of keypoint and feature performance — namely matchability. As evident by recent research [4], repeatability is not sufficient to evaluate keypoint quality. The only reliable metric is matchability, as it measures how useful the combination of keypoint and feature is. It is for this reason that we have focused our research in building a keypoint detector that learns to produce points that can be reliably matched given a deep learning descriptor. All our experiments are based on the state-of-the-art feature descriptor for point cloud data — 3DFeatNet [37]. We use...
the gradient response of the descriptor with respect to the input point cloud and combine it with contextual information from the point cloud in order to produce keypoints that generate a higher percentage of inlier matches.

The proposed approach leverages the idea, presented in [4], that state-of-the-art keypoint detectors can be built by using only the feature gradients with respect to the input. The gradients encode much of the local information used by the feature descriptor. The points of maximum gradient response provide a set of very promising candidate points. Using this initial set of keypoints, we enforce the notion of saliency that uses the combination of neural network layer activations and the gradients at that layer w.r.t. the input to select more characteristic keypoints (Fig. 1). The use of saliency has been shown to produce promising results both in images and in point clouds [30, 31, 42]. This subset of the original points will then be combined within a neural network that looks at the geometry of the point cloud and makes a decision as to which are the best points to select. The main contributions of our work are as follow:

- Unsupervised learning of keypoints: similarly to [15], we require only the rotation and translation between two point clouds when learning to predict reliable keypoints from a point cloud.

- A context agnostic approach: by combining the gradient response with the semantic information of a point cloud, our approach is less vulnerable to biases in the training data. This makes our approach robust when testing on a different context and allows the model to be used without retraining. We demonstrate this by training on the Oxford RobotCar dataset [19] and testing on the KITTI dataset [7] without retraining.

- State-of-the-art performance: Compared to the state-of-the-art, our approach generates two times more correct matches and achieves 40% more relative repeatability. We have evaluated our algorithm on the two biggest and most used LIDAR datasets - the Oxford RobotCar dataset [19] and the KITTI dataset [7]. In total both contain more than 300 kilometers of point cloud data.

2. Related Work

The work presented in this paper focuses on obtaining keypoints from point clouds. This is a topic which is very closely related to image keypoint extraction, so we will summarize both in this section. We will also outline the relevant work that has been done on point cloud-based deep learning models and saliency methods as they are part of the core contributions of our approach.

2.1. Image Keypoint Detectors

Much of the recent work on keypoint and feature extraction from images employs deep learning architectures. In [8] a keypoint detector for depth images is presented. It uses a Siamese approach by pairing two Faster-RCNN networks [25] and then enforcing a contrastive loss. Quad-Networks [27] developed an unsupervised keypoint generator by learning to rank points and keeping the rank before and after transformation while looking at the top and bottom quartiles. Methods such as [10] and [41] extract unsupervised landmarks, they find the best possible cues to improve the performance of the task at hand. Approaching landmark generation in an unsupervised fashion is the equivalent to learning keypoints, because no ground-truth points exist, the regions that maximize the end-task performance must be manually selected. Specific to unsupervised keypoint learning, [14] presents an approach that generates reliable keypoints by learning from the temporal consistency of the network activations in short videos. A successful approach to obtain good keypoints is to optimize how they respond to the content of the image and not just the geometry. TILDE [34] learns to make the points more reliable by understanding how changes in weather and lighting modify the performance of the point. An important distinction is that a keypoint is no longer focused on finding reliable geometry, but also looking at the content and the context of the point. In LIFT [38], the keypoint detector and the feature descriptor are learned together by leveraging structure from motion sequences to generate great amounts of training data. By understanding that the job of a keypoint is to increase the probability of a descriptor matching, LIFT focused on matching performance and at the time outperformed all previous approaches. Very recently, ELF [4] applied a simple approach by using the gradient response of deep learning features to produce the keypoints. They managed to outperform both TILDE and LIFT without the need to train. Their approach obtains very reliable embedding information by looking at the feature activations.

2.2. Point Cloud Keypoint Detectors

The family of solutions popularized by PointNet [23] and PointNet++ [24] introduced new ways to efficiently understand unstructured sets of points (point clouds). By learning a symmetry function approximated by a multi-layer perceptron (MLP) the authors proposed a new kind of layer that could learn an approximation of a convolutional operator over sets of unordered points. Similarly, [16] managed to increase the performance of the basic neurons by also approximating a convolution using a MLP and by performing aggregation of spatial data within each neuron. [33] presented a method to perform place recognition from point clouds. The point clouds were described using a combination of networks trained with a metric learning loss to pro-
duce a feature vector. Due to the success of the aforementioned approaches, many novel methods leveraged PointNet layers in their work [35, 42, 21, 17, 40, 5, 36, 22]. Recently, in [35], the authors studied how adversarial attacks could affect the performance of the PointNet layers. It also studied how introducing or removing points affected the performance. In [42] the authors built a saliency map to understand the effect of each point on the final prediction. They test the performance of their saliency score by performing point dropping operations to verify that they perform better than a method based on the critical-subset theory. Frustum PointNets [22] use PointNet layers as the building blocks for their approach and apply it to object detection in point clouds. Their approach produced good quality performance by combining image inputs with point cloud inputs. Also combining images and point clouds for 3D object detection, PointFusion [36] presents a method that focused on bounding box prediction by proposing a novel dense fusion architecture. The most relevant methods to our work are 3DFeatNet [37] and USIP [15]. 3DFeatNet [37] is the state-of-the-art in feature extraction for point clouds. It uses PointNet++ as a building block and learned a detector and a descriptor by using a two stage network. The main problem with this approach is that the keypoint extraction network does not perform as well as desired. Therefore, recently a novel keypoint extraction method USIP [15] was recently proposed. It focused on obtaining keypoints with high level of repeatability. This increases the performance of the USIP detector with respect to the 3DFeatNet detector significantly.

2.3. Saliency Estimation

Saliency has been studied as a way to quantify and understand what is relevant within a neural network. The definition of saliency in this context has been studied by a number of works, [30, 20, 28, 2, 1] amongst others, that seek to understand why machine learning models behave as they do. Even before the growth in popularity of neural networks, methodologies such as [3] were being designed to understand why classifiers made specific decisions. This became more prevalent with the adoption of deep learning models for most perception tasks. Approaches such as [30, 1] seek to understand what a neural network finds relevant by looking at how the gradients of a given prediction behave. The interpretation is that gradients with higher magnitude in specific areas influence the prediction. The use of saliency has been demonstrated recently [4] to be capable of generating state of the art keypoint extractors in images. ELF [4] computes the gradient of the feature map given an image then used the Kapur threshold [12] to select keypoints. In a similar fashion, Grad-CAM [29] used the gradient maps of a classification score to produce regions of interest for a given image that can aid tasks like classification, image captioning or visual question answering.

3. Methodology

We present Salient Keypoint Detection (SKD) - a method for unsupervised keypoint extraction based on the saliency of point cloud data. We define saliency as the combination of the feature activation signal at a specific layer of a pre-trained descriptor and the gradients of the same
layer with respect to the input point cloud. We leverage a trained descriptor network to produce robust keypoints directly from 3D data. We project the saliency to the spherical coordinate system of the input point cloud and extract the most informative regions. We combine those regions with context-aware features and the features of the original descriptor in order to extract robust and repeatable keypoints. Using that data, we train a neural network that learns to predict the likelihood of a point to be a keypoint. Similarly to what [4] does on images, we extract the gradients of the features at different levels of the architecture, and choose the best performing one. An evaluation of the performance at each layer is presented in Sec. 4.3. In contrast to [4], we select keypoints by combining the gradient information and the activations with the points in Euclidean space in order to determine critical points using the criterion defined by [42].

### 3.1. Pointcloud Saliency

For a given point cloud \( P \in \mathbb{R}^3 \), we extract the gradients of a pre-trained network \( \nabla F \) at a specific layer \( l \) with respect to the input, defined as \( \nabla F^P_l \). We define the *initial saliency* \( S(P) \), as the product of the feature activations of that layer \( F^P_l \) with the gradients, formally defined as:

\[
S(P) = F^P_l \cdot \nabla F^P_l
\]  

(1)

In this way the extracted initial saliency corresponds to specific points in the point cloud with good activations and are valuable based on the gradient of that layer w.r.t. the input. From a geometric perspective this can be thought of projecting the feature signal through to the input point cloud, determining how good individual points are. An example projection can be seen in Fig. 3. The initial saliences are of the same dimension as the input point cloud, \( S(P) \in \mathbb{R}^3 \).

We take these saliences and determine a secondary saliency score per point, defined as \( s_i \) for each point \( i \in P \). The score weights the contribution of \( S(P) \) by the distance from the center of the point cloud in the Spherical Coordinate system. The score is formally defined as:

\[
s_i = -\sum_{j=1}^{3} \left[ S_j(P) \odot (x_{ij} - \text{median}(x_{ij})) \right] r_i
\]

(2)

where \( j \in \{1, 2, 3\} \) defines each of the Cartesian coordinates \( x, y, z \) of point \( i \in P \), \( \odot \) is the Hadamard Product and \( r_i = \sqrt{\sum_{j=1}^{3} (x_{ij} - \text{median}(x_{ij}))^2} \) is the distance of point \( i \) to the median of the spherical core of the point cloud [42]. In other words, we transfer the gradient and its activations under the orthogonal coordinates and measure the offset from the center of the point cloud.

This transforms the original point cloud to a spherical coordinate system, and thus, further away points will have higher score, and points closer to the centre will have lower score (see Fig. 3). The reasoning for this is that under any rotation and translation of the input, the centre of the
Figure 4. Matching score evaluated on the Oxford RobotCar dataset (top row) and the KITTI dataset (bottom row). The first two columns present the percentage of matched keypoints using the 3DFeatNet descriptor when varying the distance between correspondences. The first column shows a zoomed-in version of the second column within 1 m of distance, considered relevant for geometric registration. The third column shows the performance of different approaches while varying the number of detected keypoints.

point cloud will not change, and is therefore less informative. Furthermore, when performing geometric registration, selected models (points) further apart will provide a more robust solution in comparison to points very close together.

Finally, the saliences are normalized to have a zero mean and unit variance within a single point cloud. The saliency ensures a good spatial distribution of the selected points while also selecting points with good activations based on the descriptor network. Fig. 3 illustrates the saliences of an example input point cloud at each stage of the computation.

3.2. Network Architecture

Our network architecture is depicted in Fig. 2. It consists of three parts. The first part is called the point cloud context features. This is an ensemble of four X-Conv layers [16] and two fully connected layers that have been pretrained on a feature extraction task to create stable initial estimations. These layers will learn to provide a description of the context around any given point. We use a 2 dimensional size for the context latent space. The second component is the saliency as described in the previous section. And the final component is a PCA dimensionality reduction of the original per point features. The three components are concatenated and fed to two additional fully connected layers to produce the final keypoint prediction. The network learns to infer a score per point determining the probability of it being a robust and repeatable keypoint for the original descriptor. Note that our model is descriptor-agnostic and thus can be applied to any descriptor network in order to improve the performance. The code for the TensorFlow implementation will be made available to the general public.

3.2.1 Training

During training, the input to our model consists of a tuple of point clouds and the ground truth transformation between them, \( (P_h, P_l, T) \). Both \( P_h \) and \( P_l \) are \( N \times 3 \) dimensional vectors, where \( N \) denotes the cardinality of the point cloud set. In addition, we assume to have a pre-trained model for the point descriptors. To this end we estimate the saliency \( s_i \) and features \( f_i \) for each point in both point clouds. Due to the large dimensionality of the feature space \( f_i \), we performed PCA and transformed the features that explained \( \approx 90\% \) of the data. This smooths the feature space, which lead to better results.

In addition, given the ground truth transformation between the two point clouds, we determine the bidirectional correspondence for each point, conditioned on the descrip-
tor. These correspondences are used in our loss to select matching pairs of points. Note that these correspondences need not be injective nor surjective. In other words, if for a point \( i \), the closest neighbour under the ground truth transformation, \( T \), is \( j \), it does not mean the reverse applies, nor that limits \( i \) to have a unique neighbour from \( P \).

For each point in both point clouds we extract what we have called context-aware features, \( f_c \), and concatenate them to the saliencies \( s \), and features \( f_i \). The term context-aware is used as we expect the layers that lead to this descriptor to contain information about the local geometry around each point, helping the network to understand the correlations between the local geometry and the point descriptors. We pretrain these layers on a feature learning task to obtain stable initial features in the training. We chose a small feature space of only two dimensions in order to force the network to learn rough estimates of the shape of objects that can generalize better when moving to a different dataset. Afterwards, the full architecture is trained end-to-end with the saliency and feature concatenation in the middle to create a rich and informative latent space.

The concatenated saliences, PCA features and context-aware features are fed into two additional fully connected layers to estimate the probability of each of the points being a keypoint. To this end we use a standard softmax cross entropy loss between the stacked \( P_k, P_l \) clouds and the determined correspondences, given the ground truth transformation. Due to the smaller number of correct keypoint correspondences between the two clouds, we balance the loss function terms given the keypoint to non-keypoint ratio determined by the ground truth correspondences.

### 3.2.2 Inference

During the forward pass of the network we estimate a probability of each being a correct keypoint. We can then either extract the top \( K \) keypoints, based on this probability or select all the keypoints based on the probability of each point. The extracted keypoints produced by the described approach are better suited to the descriptor as the learning iterations optimize its performance. Note, that neither Non-Maximum Suppression, nor any other threshold is applied to obtain the final set of keypoints.

### 4. Results

In this section we discuss the datasets and metrics we used to evaluate our approach and then present our findings.

#### 4.1. Datasets

In our study we used two datasets - the Oxford RobotCar dataset [19] and the KITTI odometry dataset [7]. In order to provide a fair comparison, our experiments are based on the preprocessed test data provided by [37] and [15] and we also use their evaluation scripts to make the comparison as fair as possible. We train our model using the same sequences from the RobotCar dataset as [37], and test our approach using the same test set of 3, 426 point cloud pairs which the authors provided. Furthermore, we do not train our method nor the baselines on the KITTI dataset, in order to test the generalization ability of the proposed approach. The evaluation part of the KITTI dataset, used by both [37, 15], provides only 2, 369 point clouds out of the total dataset. So as to increase the size of the KITTI evaluation dataset, we extended it using the 11 training sequences. This is possible only because the RobotCar dataset is used for all model training. The extended dataset is processed in a denser manner: for each point cloud aligning the next consecutive 10 point clouds to it using the ground truth transformation. By doing this we expanded the number of testing point cloud pairs from 2, 831 to 207, 917, which allows us to more fully study the proposed approach.
Figure 6. Top down view of a single point cloud (brown) from the KITTI dataset and 1024 generated keypoints (blue) from each of the methods. SKD was not implicitly trained to ignore the ground, but rather uses the feature signal of the descriptor and the context information to determine informative areas in the environment - corners, edges and structure.

4.2. Metrics

We used three metrics to compare the performance of our method. We focused our analysis on the keypoint extraction methods, therefore, we use the same 3DFeatNet [37] descriptor for all the methods in order to compare the performance of the keypoint detectors. Firstly, we utilize the matching score as proposed by [37]. We detect keypoints separately for two point clouds. Given the ground truth transformation, we project the keypoints from the first point cloud into the second one. Keypoints that do not have a nearest neighbour in the second point cloud are ignored from the final result (i.e. no overlap). For the rest of the keypoints, the descriptors are compared and matched to establish a correspondence. The precision is measured as the number of correct correspondences against the total number of possible matches. The metric estimates the percentage of correct correspondences based on the distance between them. We follow the convention of presenting results up to 1 m distance, as it serves as a desirable upper limit when performing registration between two point clouds. For the second metric we chose to compare the normalized relative repeatability, as proposed in [15]. The metric compares the keypoints detected in one point cloud to the keypoints detected in its corresponding point cloud. If both overlap within certain distance it is considered a match. While this saturates with high number of keypoints [4], we chose to use the metric to have a fair comparison against USIP [15]. We note that the ability to match keypoints on its own is not sufficient, therefore, we consider matching score as the more pertinent metric. Finally, we evaluated the geometric registration of our approach on the Oxford RobotCar dataset using RANSAC [6]. Similarly to [37], we consider a successful alignment all deviations from the ground truth transformation that are below 2 meters and 5 degrees.

For the successful registrations we present relative translation error (RTE), relative rotation error (RRE), success rate as the percentage of successful registrations over the entire dataset, the average number of iterations it took RANSAC to find a suitable candidate within 99% confidence (capped at 10,000 iterations) and the inlier ratio of how many points were considered when obtaining a correct registration.

4.3. Baselines

Our approach is general and can be applied to any point cloud descriptor network. For simplicity we chose to use the descriptor of [37] due to its open source availability, ease of use and being the state-of-the-art in point cloud descriptors. We select the best performing layer from the descriptor to generate the gradients on which we compute the saliency values, as evidenced in Fig. 7. We have compared against the learned keypoint detector methods of 3DFeatNet [15] and USIP [37], as well as hand-engineered keypoint extraction methods such as SIFT-3D [18], ISS [43] and Harris-3D [9], and a 3D interpretation of the ELF [4] detector. All the learning methods are trained on the Oxford RobotCar dataset and tested on both RobotCar and KITTI data. We have used the models provided online, and trained our own network. For USIP, we took the models provided by the authors trained on the Oxford RobotCar dataset. As ELF does not need training, we took the best performing layer, in accordance to Fig. 7. We adapt the approach to work on point cloud data by performing Non-Maximum Suppression in 3D space and choosing keypoints based on the Kapur Threshold [12].

4.4. Matching Score Experiments

Fig. 4 illustrates the performance of SKD in comparison to other state-of-the-art methods. The first row shows
In this paper, we present a novel method for keypoint extraction that uses saliency information to extract informative regions in a point cloud. The method concatenates signals from the gradients w.r.t. the input, context-aware features and the descriptor features and learns to predict which descriptors have a higher chance of being matched correctly. The proposed approach is descriptor-agnostic and outperforms the state-of-the-art by up to 50% in matchability and repeatability compared to the second-best method.

### 4.6. Geometric Verification Experiments

To this end, we present the results of the geometric registration on the Oxford RobotCar dataset in Tab. 1. SKD performs commensurately to the state-of-the-art in terms of relative rotation and translation error - within the standard deviation of the best-performing method, while being more than two times faster than the second best in terms of RANSAC iterations. Also, our algorithm has the highest inlier ratio by more than 50% compared to the second best.

### 5. Conclusions

In this paper, we present a novel method for keypoint extraction that uses saliency information to extract informative regions in a point cloud. The method concatenates signals from the gradients w.r.t. the input, context-aware features and the descriptor features and learns to predict which descriptors have a higher chance of being matched correctly. The proposed approach is descriptor-agnostic and outperforms the state-of-the-art by up to 50% in matchability and repeatability compared to the second-best method.
References

[1] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. In Advances in Neural Information Processing Systems, 2018.

[2] Marco Ancona, Enea Ceolini, Cengiz ztiireli, and Markus Gross. Towards better understanding of gradient-based attribute methods for deep neural networks. In International Conference on Learning Representations, 2018.

[3] David Bachrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. How to explain individual classification decisions. J. Mach. Learn. Res., 11:1803–1831, Aug. 2010.

[4] Assia Benbihi, Matthieu Geist, and Cédric Pradalier. ELF: embedded localisation of features in pre-trained CNN. In ICCV, 2019.

[5] Haowen Deng, Tolga Birdal, and Slobodan Ilic. Ppfnet: Global context aware local features for robust 3d point matching. In The IEEE Conference on Computer Vision and Pattern Recognition, 2018.

[6] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM, 24(6):381–395, 1981.

[7] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The KITTI dataset. International Journal of Robotics Research, 32(11):1231 – 1237, 2013.

[8] G. Georgakis, S. Karanam, Z. Wu, J. Ernst, and J. Koeck. End-to-end learning of keypoint detector and descriptor for pose invariant 3d matching. In CVPR, 2018.

[9] C. Harris and M. Stephens. A combined corner and edge detector. In Fourth Alvey Vision Conference, 1988.

[10] Tomas Jakab, Ankush Gupta, Hakan Bilen, and Andrea Vedaldi. Unsupervised learning of object landmarks through conditional image generation. In Advances in Neural Information Processing Systems, 2018.

[11] Andrew E. Johnson and Martial Hebert. Using spin images for efficient object recognition in cluttered 3D scenes. IEEE Transactions on pattern analysis and machine intelligence, 21(5):433–449, 1999.

[12] J.N. Kapur, P.K. Sahoo, and A.K.C. Wong. A new method for gray-level picture thresholding using the entropy of the histogram. Computer Vision, Graphics, and Image Processing, 29(3):273 – 285, 1985.

[13] Marc Khoury, Qian-Yi Zhou, and Vladlen Koltun. Learning compact geometric features. In Proceedings of the IEEE International Conference on Computer Vision, pages 153–161, 2017.

[14] Tejas Kulkarni, Ankush Gupta, Catalin Ionescu, Sebastian Borgeaud, Malcolm Reynolds, Andrew Zisserman, and Volodymyr Mnih. Unsupervised learning of object keypoints for perception and control. In Advances in Neural Information Processing Systems, 2019.

[15] Jiaxin Li and Gim Hee Lee. Usip: Unsupervised stable interest point detection from 3d point clouds. In ICCV, 2019.

[16] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. In NeurIPS, Advances in Neural Information Processing Systems, 2018.

[17] Zhijian Liu, Haotian Tang, Yujin Lin, and Song Han. Point-voxel CNN for efficient 3d deep learning. In Advances in Neural Information Processing Systems, 2019.

[18] David G. Lowe. Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vision, 2004.

[19] Will Maddern, Geoff Pascoe, Chris Linegar, and Paul Newman. 1 Year, 1000km: The Oxford RobotCar Dataset. The International Journal of Robotics Research (IJRR), 36(1):3–15, 2017.

[20] Aravindh Mahendran and Andrea Vedaldi. Salient deconvolutional networks. In European Conference on Computer Vision, 2016.

[21] Jiageng Mao, Xiaogang Wang, and Hongsheng Li. Interpolated convolutional networks for 3d point cloud understanding. In The IEEE International Conference on Computer Vision (ICCV), 2019.

[22] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[23] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[24] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In NeurIPS, Advances in Neural Information Processing Systems, 2017.

[25] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in Neural Information Processing Systems 28, 2015.

[26] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In 2009 IEEE International Conference on Robotics and Automation, pages 3212–3217. IEEE, 2009.

[27] Nikolay Savinov, Akihito Seki, Lubor Ladicky, Torsten Sattler, and Marc Pollefeys. Quad-networks: unsupervised learning to rank for interest point detection. In CVPR, 2017.

[28] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In The IEEE International Conference on Computer Vision (ICCV), Oct 2017.

[29] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In 2017 IEEE International Conference on Computer Vision (ICCV), 2017.

[30] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In Workshop at Inter-
[31] J.T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. Striving for simplicity: The all convolutional net. In Workshop at International Conference on Learning Representations, 2015.

[32] Federico Tombari, Samuele Salti, and Luigi Di Stefano. Unique shape context for 3D data description. In Proceedings of the ACM workshop on 3D object retrieval, pages 57–62. ACM, 2010.

[33] Mikaela Angelina Uy and Gim Hee Lee. PointNetVLAD: Deep Point Cloud Based Retrieval for Large-Scale Place Recognition. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[34] Yannick Verdie, Kwang Moo Yi, Pascal Fua, and Vincent Lepetit. TILDE: A temporally invariant learned detector. In CVPR, 2014.

[35] Chong Xiang, Charles R. Qi, and Bo Li. Generating 3d adversarial point clouds. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2019.

[36] Danfei Xu, Dragomir Anguelov, and Ashesh Jain. Pointfusion: Deep sensor fusion for 3d bounding box estimation. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[37] Zi Jian Yew and Gim Hee Lee. 3dfeat-net: Weakly supervised local 3d features for point cloud registration. In ECCV, 2018.

[38] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. Lift: Learned invariant feature transform. In European Conference on Computer Vision, 2016.

[39] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3Dmatch: Learning local geometric descriptors from RGB-D reconstructions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1802–1811, 2017.

[40] Wenxiao Zhang and Chunxia Xiao. PCAN: 3d attention map learning using contextual information for point cloud based retrieval. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[41] Yuting Zhang, Yijie Guo, Yixin Jin, Yijun Luo, Zhiyuan He, and Honglak Lee. Unsupervised discovery of object landmarks as structural representations. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[42] Tianhang Zheng, Changyou Chen, Junsong Yuan, and Kui Ren. Pointcloud saliency maps. In ICCV, 2019.

[43] Yu Zhong. Intrinsic shape signatures: A shape descriptor for 3d object recognition. In 2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops, pages 689–696. IEEE, 2009.
Supplementary Material

1. Introduction

This supplemental material addresses further issues that could not be included in the paper. It aims to give a more qualitative assessment of the behavior of the proposed approach by itself and compared to the main baselines. First, we present a study to quantify the contribution of the full approach with respect to a baseline which consists on just the use of the saliencies to estimate keypoints. Second, we present several qualitative comparisons of keypoint generated by our approach in comparison to 3DFeatNet [37] and USIP [15] with incremental number of generated keypoints on point clouds from the KITTI [7] and the Oxford RobotCar [19] datasets. Also, videos have been generated to showcase how keypoints are being generated in our approach and the main baselines.

2. Network Contribution

In Fig. 8, we show the comparison between using only the saliency values to estimate the keypoints and the proposed approach — SKD. The metrics and the test samples are the same as in Figure 4 in the paper. It can be seen that the contribution of the neural network that learns from the combined information gives a substantial increase in performance as is to be expected. What the figure outlines is the actual magnitude of the contribution which gives us further insights and context on how the method functions.

3. KITTI Odometry Results

We have performed registrations experiments for all the training sequences of the KITTI dataset in the same fashion as we did for the Oxford RobotCar Dataset, see Tab. 2. The same procedures and metrics as reported of the Table 1 are used for this experiment. We have used the increased sampling that we reported in the paper on the KITTI data to have a more thorough experimental validation due to the increased number of testing samples - nearly a 100 times more. The geometric registration results showcase the same behavior that was observed in the Oxford RobotCar dataset, with the proposed method obtaining a high percentage of correct matches (inliers), that enable a much faster convergence of the robust estimator - RANSAC [6].

4. Qualitative Results

Fig. 9 and Fig. 10 present qualitative results showing the different behavior of our approach and the most relevant baselines. The experiment consists on generating an increasing number of keypoints in the point cloud to see how each method focuses the point selection. 3DFeatNet selects points in areas with high density of points, USIP generates keypoints in a widespread fashion representing all the areas of the point cloud, our approach focuses on generating keypoints in areas where the feature descriptor performs the best and ignores less discriminative areas, such as the ground.

5. Video

We have also included in this supplemental material video showcasing how each methods keypoints are generated over the entire sequence 00 from the KITTI dataset. The video shows two consecutive point clouds at any frame with 128 detected corresponding keypoints in each. We note that the keypoints extracted by our approach are consistent between point clouds, whereas the baselines’ keypoints fluctuate between observations. The video is available here.
Figure 8. Network contribution of our method. We compare the use of the combined saliencies with respect to the full approach. The results displayed in the figure show the clear contribution the neural network is making to the keypoint generation process.

| Detector + Descriptor Method | RTE (m) | RRE (°) | Success Rate | Avg # iter | Inlier ratio |
|------------------------------|---------|---------|--------------|------------|-------------|
| 3DFeatNet [37] + 3DFeatNet [37] | 0.142 ± 0.120 | 0.533 ± 0.410 | 97.80% | 3917 | 12.7% |
| USIP [15] + 3DFeatNet [37] | 0.203 ± 0.193 | 0.637 ± 0.517 | 97.12% | 5324 | 11.0% |
| **SKD + 3DFeatNet [37]** | **0.140 ± 0.134** | **0.579 ± 0.480** | **96.52%** | **594** | **32.2%** |

Table 2. Geometric registration evaluation on the KITTI dataset as evaluated by RANSAC. The proposed method performs commensurately to the state-of-the-art, within the standard deviation, while managing to find a correct transformation six times faster with 150% more inliers compared to the best baseline.
Figure 9. Qualitative results for the Oxford RobotCar Dataset. For all methods the number of generated keypoints is increased. This shows what relative importance each method gives to certain areas of the point cloud.
Figure 10. Qualitative results for the KITTI Dataset. For all methods the number of generated keypoints is increased. This shows what relative importance each method gives to certain areas of the point cloud.