

This paper has been accepted for publication at the
British Machine Vision Conference (BMVC), Cardiff, 2019.

Matching Features without Descriptors: Implicitly Matched Interest Points

Titus Cieslewski¹
titus@ifi.uzh.ch

Michael Bloesch²
m.bloesch@imperial.ac.uk

Davide Scaramuzza¹
sdavide@ifi.uzh.ch

¹ Dept. of Informatics and
Neuroinformatics
University of Zurich and ETH Zurich
Switzerland

² Dept. of Computing
Imperial College London
United Kingdom

Abstract

The extraction and matching of interest points is a prerequisite for many geometric computer vision problems. Traditionally, matching has been achieved by assigning descriptors to interest points and matching points that have similar descriptors. In this paper, we propose a method by which interest points are instead already implicitly matched at detection time. With this, descriptors do not need to be calculated, stored, communicated, or matched any more. This is achieved by a convolutional neural network with multiple output channels and can be thought of as a collection of a variety of detectors, each specialised to specific visual features. This paper describes how to design and train such a network in a way that results in successful relative pose estimation performance despite the limitation on interest point count. While the overall matching score is slightly lower than with traditional methods, the approach is descriptor free and thus enables localization systems with a significantly smaller memory footprint and multi-agent localization systems with lower bandwidth requirements. The network also outputs the confidence for a specific interest point resulting in a valid match. We evaluate performance relative to state-of-the-art alternatives.

Multimedia Material

Source code and data for this work are available at
https://github.com/uzh-rpg/imips_open.

1 Introduction

Many applications of computer vision, such as structure from motion and visual localization, rely on the generation of point correspondences between images. Correspondences can be

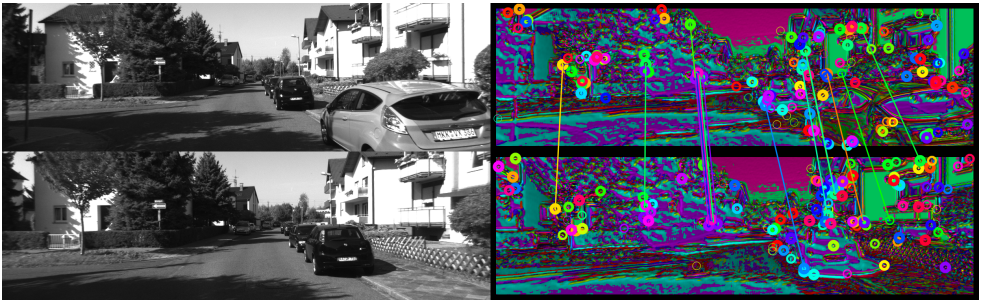


Figure 1: We propose a CNN interest point detector which provides implicitly matched interest points: descriptors are not needed for matching. This image illustrates the output of the network. Hue indicates which channel has the strongest response for a given pixel, and brightness indicates that response. Circles indicate the interest points, which are the global maxima of each channel. Lines indicate inlier matches after P3P localization.

found densely [0, 13, 29], where a correspondence is sought for every pixel, or with sparse feature matching, where correspondences are only established for a few distinctive points in the images. While dense correspondences capture more information, it is often of interest to establish them only sparsely. Sparse correspondences make algorithms like visual odometry or bundle adjustment far more tractable, both in terms of computation and memory.

Sparse feature matching used to be solved with hand-crafted descriptors [0, 21, 61], but has more recently been solved using learned descriptors [6, 28, 39]. In this paper, we propose a novel approach that exploits convolutional neural networks (CNNs) in a new way. Traditionally, features are matched by first detecting a set of interest points, then combining these points with descriptors that locally describe their surroundings, to form visual features. Subsequently, correspondences between images are formed by matching the features with the most similar descriptors. This approach, which has been developed with hand-crafted methods, has been directly adopted in newer methods involving CNNs. As a result, different CNNs have been used for the different algorithms involved in this pipeline, such as interest point detection, orientation estimation, and descriptor extraction.

We propose a method that uses only a single, convolution-only neural network that subsumes all of these algorithms. This network can be thought of as an extended interest point detector, but instead of outputting a single channel that allows the selection of interest points using non-maximum suppression, it outputs multiple channels (see Figure 1). For each channel only the global maximum is considered an interest point, and the network is trained in such a way that the points extracted by the same channel from different viewpoints are correspondences. As with traditional feature matching, geometric verification is required to reject outlier correspondences. The network can alternatively be thought of as a dense point descriptor, but instead of expressing descriptors along the channel axis of the output tensor, each channel represents the response to a function defined in the descriptor space.

An important benefit of our method is that descriptors do not need to be calculated, stored, communicated or matched any more. Previously, the minimal representation of an observation for relative pose estimation consisted of point coordinates and their associated descriptors. With our method, the same observation can be represented with as little as the point coordinates (3 bytes for up to 4096×4096 images), ordered consistently with the channel order.

We provide an evaluative comparison of our method with other state-of-the-art methods on indoor, outdoor and wide-baseline datasets. As our evaluations show, it is a viable alterna-

tive to methods involving explicit descriptors particularly with narrow baselines, achieving a similar pose estimation performance.

2 Related Work

Modern feature matching can be best understood by considering the sub-problems and their historical context. Calculating dense correspondences for images used to be prohibitive, so instead early work concentrated on finding sparse sets of points that could be used for correspondence generation. In a first interest point detector, [26] identified points which are explicitly *distinct* from the points that surround them—thus increasing the likelihood to be unambiguously matched in another image. Subsequently, faster approximations for distinctiveness have been found, whether using first-order approximations [12, 34], or convolutional filters [9, 21]. Alternatively, a detector can explicitly target a subset of distinctive points, such as corners and dots [30]. All of these methods calculate a response for every pixel in the image, and the n pixels with the largest response are selected as interest points. This process typically involves non-maximum suppression in order to prevent directly neighboring points from being selected.

Once a set of interest points is extracted in the images, they need to be matched between each other to establish correspondences. One can match points based directly on the surrounding image patches, but this is fragile to slight changes in illumination and viewpoint. Instead, descriptors can be used, which are functions of patches and whose output is typically lower-dimensional, but invariant to some amount of illumination and viewpoint change, yet still distinctive enough to differ between the different points extracted in one image. A popular class of traditional descriptors is histograms of gradients (HoG) [21]. Another example are binary descriptors, which are particularly efficient to calculate [9, 19, 31].

Most descriptors, however, are still sensitive to large affine transformations such as changes in scale and orientation. Consequently, modern feature matchers use multi-scale detection [19, 24], and orientation estimation [21, 31]. A wide variety of traditional feature matching systems comprising these components exist, see the survey in [27].

Recent success of convolutional neural networks (CNNs), however, has led the community to revisit these systems and replacing their components with CNN-based methods. For detection, the traditionally handcrafted “featureness” responses can directly be replaced with a fully convolutional neural network. Rather than just imitating traditional interest point detectors, CNN-based detectors can be trained to be invariant across different viewpoints [17], to present consistent ranking in the images in which they are extracted [32] and to provide particularly sharp and thus unambiguous responses [41]. A majority of these methods is compared in [18]. CNNs are furthermore proven function approximators for image patches and thus well suited for descriptor calculations. The output channels of a CNN can simply be interpreted as the coefficients of a descriptor [11, 15, 20, 25, 35, 38, 40]. A comparison of traditional and learned descriptors is provided in [33]. While the results of comparing CNN-based methods to traditional methods do not yet suggest absolute superiority of CNN-based methods [33], an advantage of CNN-based methods is that they are malleable: firstly, they can adapt to and learn from new data: Consider an application where the type of environment is known beforehand—CNN-based methods can be trained to work particularly well on that particular type of environment. Secondly, they can adapt to or be trained together with other components of a larger system.

Two recent systems that fully integrate CNN-based methods and do this kind of joint

training are LF-Net [28] and SuperPoint [6]. LF-Net [28] builds on top of a previous method by the same authors, LIFT [39], the first such system, in which the method was trained on a set of pre-extracted patches. [28], instead, is trained in a self-supervised and more unconstrained manner, only requiring an image sequence with ground truth depths and poses. Like [39], [28] uses separate CNNs for multi-scale interest point detection, feature orientation estimation and feature description. In contrast, SuperPoint [6] only contains an interest point detector and a feature descriptor network, both of which are sharing several encoder layers. It also does not explicitly express multi-scale detection, but rather trains multi-scale detection implicitly. It is first pre-trained on labeled synthetic images, then fine tuned on artificially warped real images. Both [28] and [6] still consider the traditional components of feature detection as separate functional units, even if the whole system is trained end-to-end.

In contrast, we offer a novel approach in which all components are subsumed into a single network. Beside having the benefit that all components can be jointly trained (from scratch) and thus tailored to one another, we also get rid of explicit descriptors. Instead, interest points are implicitly matched by the CNN output channel from which they originate. In practice, this results in memory, computation and potentially data transmission savings, as descriptors do not need to be stored, matched or communicated any more.

There have been some previous attempts to significantly reduce the amount of data associated with descriptors. In [67], the authors replace descriptors with word identifiers of the corresponding visual word in a Bag-of-Words visual vocabulary [36]. This can be used jointly with Bag-of-Words place recognition in order to facilitate multi-agent relative pose estimation with minimal data exchange. In [23], the authors propose highly compressed maps for visual-inertial localization in which binary descriptors are projected down to as little as one byte. In contrast, our approach circumvents the use of any explicit descriptor, by implicitly embedding a form of descriptor in the learned detection algorithm itself.

3 System Overview

In analogy to other state-of-the-art approaches for interest point detection, we employ a neural network to predict a per-pixel response from an input image. But instead of only predicting a single output score for every pixel, we predict n different activations, see Figure 2. The neural network consists of 14 layers of 3×3 convolutions with stride 1 and leaky ReLU activations, except for the final activation, which is a sigmoid. The first half layers output 64 channels, the second half 128. From each final output channel i , we extract the argmax as i -th interest point with coordinates c_i . The key concept is that we then *implicitly* match the interest point from the same output channel across multiple frames. This has the advantage of inherently solving the data association problem, without the need to use descriptors explicitly. Formally, point c_i from image I is matched with point c'_i from image I' . At test time, a relative pose between both images can then be computed based on the corresponding interest point coordinates.

During training, an inlier determination module (Section 3.1) processes the matches (c_i, c'_i) and determines which of them are inliers. This relies on ground truth correspondences $\Psi(c_i)$. Interest points, correspondences and inlier labels shape mini-batches that contribute to the loss for a given training step as described in Section 4. During evaluation and deployment, the inlier determination module is replaced with an application-specific geometric verifier, such as a perspective-n-point (PnP) localizer. In our experiments, we evaluate our system with P3P [9] localization using RANSAC [8], which produces a relative pose esti-

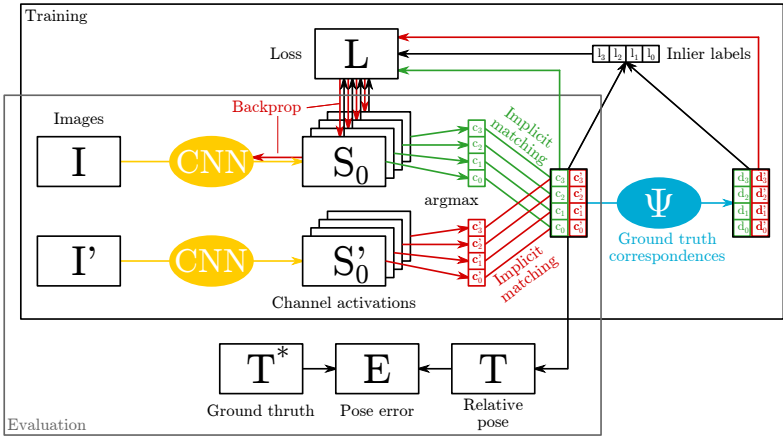


Figure 2: Overview of our approach. Given an image, a CNN computes n activations, the argmax of which are considered the n interest points. The interest points from two different images are matched by channel. During training, these matches are labeled as *inliers*, *outliers*, or *unassigned* using ground truth correspondences. Our loss promotes inliers, penalizes outliers, and suppresses redundancy. For evaluation, the correspondences are used to compute a relative pose between two images which is compared against ground truth.

mate. We compare this pose estimate to the ground truth relative pose to assess the viability of our method for visual pose estimation.

3.1 Inlier and True Correspondence Determination

Our training methodology requires inlier labels and correspondences in order to calculate the loss. Given a method Ψ to calculate true correspondences between the two images provided in a training step, we label an interest point c_i as *inlier* if the correspondence of the matched interest point from the other image $\Psi^{-1}(c_i)$ is within $3px$ of the matched interest point in the other image and vice versa. Otherwise, the match is either labeled an *outlier* if the correspondence lies somewhere else within the respective image, or *unassigned* if it is outside the image frame. In some datasets, a method to compute ground truth correspondence is provided. If such a method is not provided, correspondence can be calculated from ground truth depth and pose [62]. In case these are not provided either, the correspondence can be estimated using an SfM algorithm [28] given image sequences, or by using direct tracking such as KLT [27]. With this only image regions with sufficient texture can be tracked, which is acceptable since image regions without texture are unlikely to be of interest.

4 Training Methodology

We use standard iterative training using Adam updates [14]. In each iteration, a training sample, which consists of two images of the same scene is forwarded through the system to obtain a set of matches $\{(c_i, c'_i), i \in \{0, \dots, n-1\}\}$ and an associated set of true correspondences $\{(d_i, d'_i) = (\Psi(c_i), \Psi^{-1}(c'_i))\}$ according to Figure 2. During training, the loss is only applied at these sparse locations. In order to allow for efficient gradient backpropagation, patches are gathered from these locations. In image I , two mini-batches are formed: one from stacking *interest point patches* $P(c_i)$ centered around c_i and shaped according to the

chn.	label	$P(c_0)$	$P(c_1)$	$P(c_2)$	$P(d'_0)$	$P(d'_1)$	$P(d'_2)$
0	inlier	↑					n/a
1	outlier	↓	↓			↑	n/a
2	unass.	↓					n/a

Table 1: Mini-batch toy example to illustrate losses. For inliers, the activation of the maxima is strengthened while suppressing the activation in the other channels. For outliers, the activation of the maxima is weakened while promoting the response of the true correspondence.

receptive field, $r \times r$, of a single pixel at the output. The other from stacking *correspondence patches* $P(d'_i)$ centered around the correspondences d'_i . Both batches have a shape $[n, r, r, 1]$. The network transforms both of them into output tensors of shape $[n, 1, 1, n]$. The training loss is now applied to these tensors. Since they are flat along the height and width dimensions, we can conceptualize them as square matrices along the batch and channel dimensions, and visualize how the loss is applied to them in Table 1 for a toy example with $n = 3$. Note that the diagonal in the first tensor contains the responses of patch $P(c_i)$ at channel i , which is the maximum value in channel i and the value that caused c_i to be selected as interest point. Similarly, the diagonal in the second tensor contains the responses that *should* be the maximum in the given channel considering the correspondence from the interest point selected in the other image. The training loss that is applied to these tensors has three components with their specific purpose:

- *Inlier reinforcement* reinforces interest points that are inliers in a given training sample, and suppresses interest points that are outliers.
- *Redundancy suppression* ensures that different channels do not converge to the same points.
- *Correspondence reinforcement* reinforces true correspondences of all points which are outliers in a given training sample.

The entire loss formulation is symmetrically applied to the other image I' .

Let p_{ij} be the scalar response of channel j to patch $P(c_i)$ and l_i the inlier label of that patch. Then, the inlier reinforcement loss is simply the cross-entropy loss according to that label, applied where $i = j$:

$$L_{inl}(p_{ij}, l_i) = \begin{cases} -\log(p_{ij}), & l_i = \text{inlier}, i = j, \\ -\log(1 - p_{ij}), & l_i = \text{outlier}, i = j, \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

This is the loss responsible for learning a high response for points that are likely to result in inliers. In Table 1 the inlier reinforcement effect is observed for channel 0 on $P(c_0)$ and the outlier suppression effect is present for channel 1 on $P(c_1)$.

To prevent channels from converging to the same interest points, a loss is applied on inlier patches that suppresses the response on all channels, except the one which gave rise to the inlier:

$$L_{red}(p_{ij}) = \begin{cases} -\log(1 - p_{ij}), & l_i = \text{inlier}, i \neq j, \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

In Table 1 the redundancy suppression is present on channels 1 & 2 on $P(c_0)$. We found that without redundancy suppression, all channels tend to converge to a single feature, all selecting the same interest point in every image.

Finally, we have found that our network does not converge with the above losses alone. Thus, for channels with outliers, we promote p'_{ii} , the response of channel i to patch $P(d'_i)$:

$$L_{cor}(p'_{ii}) = \begin{cases} -\log(p'_{ii}), & l_i = \text{outlier} \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

In Table 1 the correspondence reinforcement can be observed in channel 1 on $P(d'_1)$, which is the patch extracted at the correspondence of the maximum activation of the paired image.

4.1 Training Pair Selection

In datasets such as HPatches [10], pairs are pre-selected. For image sequences, we extract pairs as follows: Given one image, points are densely sampled and subjected to KLT tracking for as far into subsequent images as possible. A pair is then formed between this initial image and a random subsequent image in which at least a fraction o of the initial points is still tracked. o thus reflects the minimum scene overlap between the two images. The benefit of this method is that it can be applied to uncalibrated image sequences, while providing good guarantees regarding minimum scene overlap. For training, we use $o = 0.3$, for selecting pairs during evaluation $o = 0.5$.

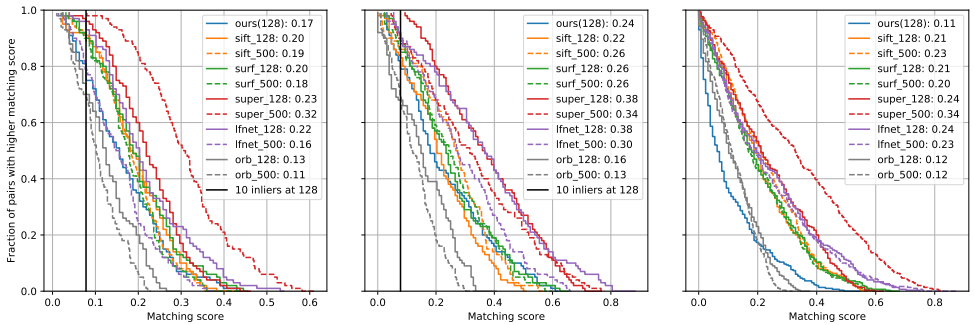
5 Experiments

We subject our method to evaluation and comparison to state-of-the-art on three datasets: The viewpoint-variant part of the wide-baseline HPatches benchmark [10, 18], sequence 00 of the outdoor autonomous driving dataset KITTI [100] and sequence v1_01 of the indoor drone dataset EuRoC [9]. For the two sequences, 100 image pairs are randomly selected using the method in Section 4.1. KITTI and EuRoC are stereo datasets, and we only use the left camera for the image pairs. Rotation and translation differences within these pairs are plotted in the supplementary material. For the HPatches benchmark, we train our network using the provided training split, while for KITTI and EuRoC, we train our network on a separate dataset, TUM mono [6], where we use the rectified images of sequences 01, 02, 03 (indoors) and 48, 49, 50 (outdoors). We found that adding sequences 04 to 47 does not result in better performance. For the sequences, we use KITTI 05 as validation dataset. Ground truth correspondences are given by ground truth homographies in the HPatches training data. In TUM mono, they are instead established using KLT as discussed in Section 3.1.

We compare our method to SIFT [20], SURF [8], ORB [30], LF-Net [28] and SuperPoint [5]. For SIFT, SURF and ORB, we use the OpenCV implementation, while for LF-Net and Superpoint, we use the publicly available code and pre-trained weights. All baselines are evaluated both at 128 interest points, like our method, and at a more native interest point count of 500.

5.1 Matching score

For all datasets, we evaluate matching score, which is the fraction of inlier matches among all matches. In HPatches, the ground truth homography is used to distinguish inlier from outlier matches. In KITTI and EuRoC we instead use the inliers that are determined during pose estimation, see Section 5.2. As a consequence, the matching score here more closely reflects the matching score that is achieved in practice when doing relative pose estimation,



(a) KITTI

(b) Euroc

(c) HPatches

Figure 3: Matching score distributions (higher is better).

whether in SLAM or in map localization. Figure 3 shows the matching score obtained in the three datasets. Both on KITTI and on EuRoC, our method performs slightly worse than all baselines except ORB, especially if the latter are permitted to extract more points than our method. Note, however, that SURF, SIFT, SuperPoint and LF-NET use 64, 128, 256 and 256 floating points to describe each interest point, respectively, in addition to point locations. Instead, our method represents a visual location using only 128 properly ordered point locations. The results on HPatches show that our method is still not very well suited to the significant viewpoint and scale changes present in that dataset. Nevertheless, we believe that implicit interest point matching opens a new research direction in terms of more closely integrated detector training. Future work to address strong viewpoint and scale changes could include considerations such as scale and rotation invariance, whether by modeling inside the neural network [28] or by data augmentation and curricular learning [9].

5.2 Pose Estimation Accuracy

To understand how matching score translates to pose estimation accuracy, we compare the pose estimated using our method and SIFT with the ground truth relative pose on both KITTI and EuRoC. Both in SLAM and map localization absolute relative pose is typically obtained from interest points matched to 3D point locations by P3P localization [9] with RANSAC [8]. To obtain 3D point locations in one of the two images, we use epipolar stereo matching of the interest points extracted using our method or a baseline with the corresponding image from the right camera. Rotation and translation error are measured with the geodesic (angle in angle-axis) and euclidean distances. In Figure 4 these errors are compared to the inlier count for each pair in the KITTI testing set, using both our interest points and SIFT. The same plot for EuRoC can be found in the supplementary material. We can see that both with our method and with SIFT, an inlier count above 10 indicates a good relative pose (rotation error below 1° , translation error below 30cm). Estimates get slightly better up to an inlier count of 20, beyond which they do not improve by much, even if much more interest points are extracted. We transfer this insight to Figure 3, where we indicate the matching score corresponding to 10 inliers with a vertical line. From this, we can see that our method results in good relative poses in 80% of the tested image pairs.

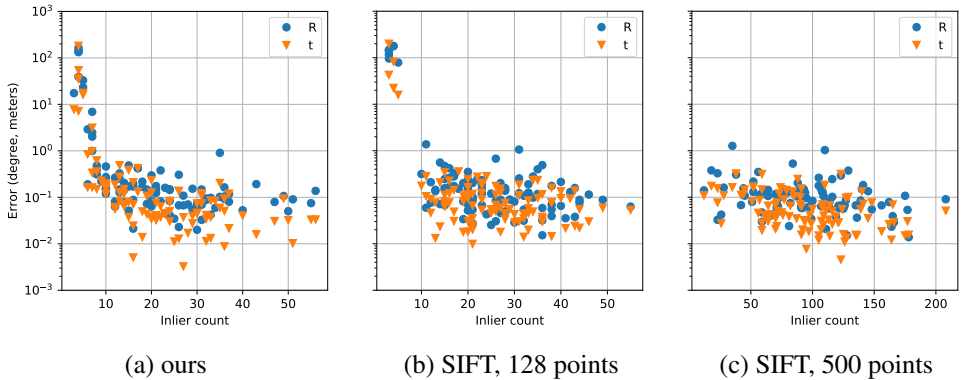


Figure 4: Pose error versus inlier count on KITTI. R: rotation error. t: translation error.

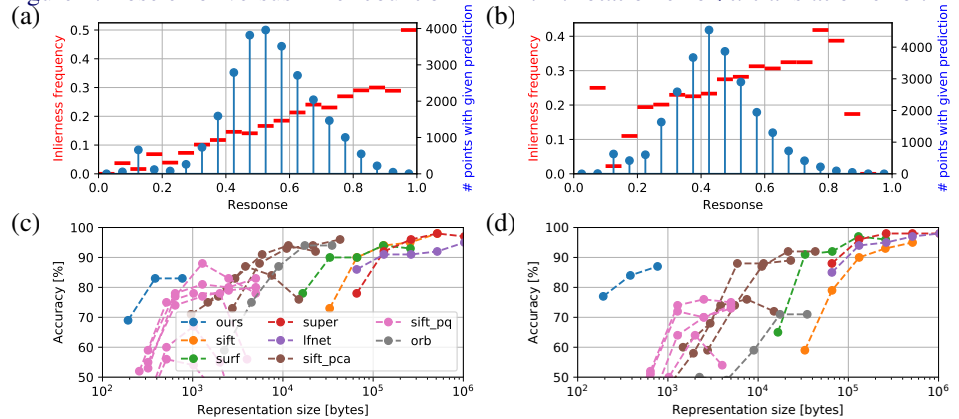


Figure 5: (a), (b) Empiric “inlierness” frequency versus response, and histogram of interest point responses. Inlierness frequency is calculated per histogram bin by dividing the amount of inliers within a response range by the total amount of interest points within that range. (c), (d) Accuracy versus representation size for our method and baselines, including compression schemes for SIFT. Note the logarithmic scale. Left: KITTI, right: EuRoC.

5.3 Other results

Some additional insights are shown in Figure 5. The parts (a) and (b) show how the probability of a point being an inlier is correlated with the response of the corresponding channel at that point, using our network. As can be seen, there is a good correlation, which indicates that the response has some predictive power regarding the probability of a point resulting in an inlier. This could potentially be exploited, for example in RANSAC model sampling. In (c) and (d), we show how pose estimation accuracy compares to the amount of data needed to represent a visual frame for our method, the previously used baselines and two compression methods applied to SIFT descriptors. Here, accuracy represents the fraction of pose estimates with rotation and translation errors below 1° and 30cm (KITTI) and 3° and 10cm (EuRoC). The two compression methods are principle component analysis (PCA) projection of SIFT descriptors and product quantization [14, 23], see the supplementary material for details. For our method, we evaluate networks with output channel counts $\in \{64, 128, 256\}$. For all other methods, we evaluate $\{64, 128, 256, 500, 1000\}$ interest points per image. As can be seen, our method outperforms baselines in the trade-off between accuracy and representation size.

6 Conclusion

In this paper, we have introduced a descriptor-free approach for detection and matching of sparse visual features. We rely on a convolutional neural network to predict multiple activation layers and we define the location of the maximal response in each layer to be an interest point. The key novelty is that instead of relying on descriptors for matching, interest points are uniquely associated to the activation layer they are extracted from. This setup allows us to train the traditionally modularized interest point detection, description, and matching processes jointly in a simple setup, while at the same time getting rid of the requirement for explicit descriptors. Without descriptors, visual features can be stored, communicated and matched at a highly reduced cost. For this system, we have devised a self-supervised training methodology that reinforces interest points that result in inliers, ensures that each channel specializes on different features, and uses ground truth correspondences to ensure that channels not resulting in inliers find good candidate features. This training can be achieved with uncalibrated image sequences. Albeit achieving slightly lower matching scores when compared to other approaches that do use descriptors, we demonstrated the applicability of our descriptor-free approach in a visual pose estimation setup.

7 Acknowledgments

This work was supported by the National Centre of Competence in Research (NCCR) Robotics through the Swiss National Science Foundation and the SNSF-ERC Starting Grant. The Titan Xp used for this research was donated by the NVIDIA Corporation.

References

- [1] Vassileios Balntas, Karel Lenc, Andrea Vedaldi, and Krystian Mikolajczyk. HPatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2017.
- [2] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. *Comput. Vis. Image. Und.*, 110(3):346–359, 2008. doi: 10.1016/j.cviu.2007.09.014.
- [3] Michael Burri, Janosch Nikolic, Pascal Gohl, Thomas Schneider, Joern Rehder, Sammy Omari, Markus W. Achtelik, and Roland Siegwart. The EuRoC micro aerial vehicle datasets. *Int. J. Robot. Research*, 35(10):1157–1163, 2015. doi: 10.1177/0278364915620033.
- [4] M. Calonder, V. Lepetit, M. Ozuysal, T. Trzcinski, C. Strecha, and P. Fua. BRIEF: Computing a local binary descriptor very fast. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(7):1281–1298, 2012.
- [5] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperPoint: Self-supervised interest point detection and description. In *IEEE Conf. Comput. Vis. Pattern Recog. Workshops (CVPRW)*, 2018.
- [6] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct Sparse Odometry. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(3):611–625, March 2018. doi: 10.1109/TPAMI.2017.2658577.
- [7] Gunnar Farneback. Two-frame motion estimation based on polynomial expansion. In *Scandinavian Conf. on Im. Analysis (SCIA)*, pages 363–370, 2003.
- [8] 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. doi: 10.1145/358669.358692.
- [9] Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng. Complete solution classification for the perspective-three-point problem. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(8):930–943, August 2003. doi: 10.1109/TPAMI.2003.1217599.
- [10] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The KITTI dataset. *Int. J. Robot. Research*, 32(11):1231–1237, 2013. doi: 10.1177/0278364913491297.
- [11] Xufeng Han, Thomas Leung, Yangqing Jia, Rahul Sukthankar, and Alexander C. Berg. MatchNet: Unifying feature and metric learning for patch-based matching. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2015.
- [12] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Proc. Fourth Alvey Vision Conf.*, volume 15, pages 147–151, 1988. doi: 10.5244/C.2.23.
- [13] Berthold K.P. Horn and Brian G. Schunck. Determining optical flow. *J. Artificial Intell.*, 17(1):185 – 203, 1981. doi: 10.1016/0004-3702(81)90024-2.

- [14] Herve Jegou, Matthijs Douze, and Cordelia Schmid. Product quantization for nearest neighbor search. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(1):117–128, January 2011. doi: 10.1109/TPAMI.2010.57.
- [15] Michel Keller, Zetao Chen, Fabiola Maffra, Patrik Schmuck, and Margarita Chli. Learning deep descriptors with scale-aware triplet networks. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018.
- [16] Diederik P. Kingma and Jimmy L. Ba. Adam: A method for stochastic optimization. In *Int. Conf. Learn. Representations (ICLR)*, 2015.
- [17] Karel Lenc and Andrea Vedaldi. Learning covariant feature detectors. In *Eur. Conf. Comput. Vis. Workshops (ECCVW)*, pages 100–117, 2016.
- [18] Karel Lenc and Andrea Vedaldi. Large scale evaluation of local image feature detectors on homography datasets. In *British Mach. Vis. Conf. (BMVC)*, 2018.
- [19] S. Leutenegger, M. Chli, and R.Y. Siegwart. BRISK: Binary robust invariant scalable keypoints. In *Int. Conf. Comput. Vis. (ICCV)*, pages 2548–2555, 2011. doi: 10.1109/ICCV.2011.6126542.
- [20] Antonio Loquercio, Marcin Dymczyk, Bernhard Zeisl, Simon Lynen, Igot Gilitschenski, and Roland Siegwart. Efficient descriptor learning for large scale localization. In *IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 3170–3177, 2017. doi: 10.1109/ICRA.2017.7989359.
- [21] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.*, 60(2):91–110, November 2004. doi: 10.1023/B:VISI.0000029664.99615.94.
- [22] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Int. Joint Conf. Artificial Intell. (IJCAI)*, pages 674–679, 1981.
- [23] Simon Lynen, Torsten Sattler, Michael Bosse, Joel Hesch, Marc Pollefeys, and Roland Siegwart. Get out of my lab: Large-scale, real-time visual-inertial localization. In *Robotics: Science and Systems (RSS)*, 2015. doi: 10.15607/RSS.2015.XI.037.
- [24] K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. *Int. J. Comput. Vis.*, 60(1):63–86, 2004. doi: 10.1023/B:VISI.0000027790.02288.f2.
- [25] Anastasiia Mishchuk, Dmytro Mishkin, Filip Radenovic, and Jiří Matas. Working hard to know your neighbors margins: Local descriptor learning loss. In *Conf. Neural Inf. Process. Syst. (NeurIPS)*, pages 4826–4837. 2017.
- [26] H. P. Moravec. *Obstacle Avoidance and Navigation in the Real World by Seeing Robot Rover*. PhD thesis, Carnegie-Mellon University, Pittsburgh, Pennsylvania, September 1980.
- [27] Dibyendu Mukherjee, Q. M. Jonathan Wu, and Guanghui Wang. A comparative experimental study of image feature detectors and descriptors. *Mach. Vis. and Applications*, 26(4):443–466, May 2015. doi: 10.1007/s00138-015-0679-9.

- [28] Yuki Ono, Eduard Trulls, Pascal Fua, and Kwang Moo Yi. LF-Net: Learning local features from images. In *Conf. Neural Inf. Process. Syst. (NeurIPS)*, pages 6234–6244, 2018.
- [29] Ignacio Rocco, Mircea Cimpoi, Relja Arandjelović, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Neighbourhood consensus networks. In *Conf. Neural Inf. Process. Syst. (NeurIPS)*, pages 1651–1662, 2018.
- [30] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *Eur. Conf. Comput. Vis. (ECCV)*, pages 430–443, 2006. doi: 10.1007/11744023_34.
- [31] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. ORB: An efficient alternative to SIFT or SURF. In *Int. Conf. Comput. Vis. (ICCV)*, 2011.
- [32] Nikolay Savinov, Akihito Seki, L’ubor Ladický, Torsten Sattler, and Marc Pollefeys. Quad-networks: Unsupervised learning to rank for interest point detection. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2017.
- [33] Johannes L. Schonberger, Hans Hardmeier, Torsten Sattler, and Marc Pollefeys. Comparative evaluation of hand-crafted and learned local features. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 1482–1491, 2017.
- [34] Jianbo Shi and Carlo Tomasi. Good features to track. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 593–600, 1994. doi: 10.1109/CVPR.1994.323794.
- [35] Edgar Simo-Serra, Eduard Trulls, Luis Ferraz, Iasonas Kokkinos, Pascal Fua, and Francesc Moreno-Noguer. Discriminative learning of deep convolutional feature point descriptors. In *Int. Conf. Comput. Vis. (ICCV)*, 2015.
- [36] J. Sivic and A. Zisserman. Video Google: a text retrieval approach to object matching in videos. In *Int. Conf. Comput. Vis. (ICCV)*, 2003. doi: 10.1109/ICCV.2003.1238663.
- [37] Danilo Tardioli, Eduardo Montijano, and Alejandro R. Mosteo. Visual data association in narrow-bandwidth networks. In *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, pages 2572–2577, 2015. doi: 10.1109/IROS.2015.7353727.
- [38] Xing Wei, Yue Zhang, Yihong Gong, and Nanning Zheng. Kernelized subspace pooling for deep local descriptors. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018.
- [39] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. LIFT: Learned invariant feature transform. In *Eur. Conf. Comput. Vis. (ECCV)*, pages 467–483, 2016.
- [40] Sergey Zagoruyko and Nikos Komodakis. Learning to compare image patches via convolutional neural networks. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 4353–4361, 2015. doi: 10.1109/CVPR.2015.7299064.
- [41] Linguang Zhang and Szymon Rusinkiewicz. Learning to detect features in texture images. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018.