Matching Features without Descriptors: Implicitly Matched Interest Points (IMIPs)

Titus Cieslewski[1], Michael Bloesch[2], and Davide Scaramuzza[1]
[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 visual pose estimation and related 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. Thanks to 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. The i-th interest point is the location of the maximum of the i-th channel, and the i-th interest point in one image is implicitly matched with the i-th interest point in another image. This paper describes how to design and train such a network in a way that results in successful relative pose estimation performance with as little as 128 output channels. While the overall matching score is slightly lower than with traditional methods, the network also outputs the confidence for a specific interest point resulting in a valid match. Most importantly, the approach completely gets rid of descriptors and thus enables localization systems with a significantly smaller memory footprint and multi-agent localization systems that require significantly less bandwidth. We evaluate performance relative to state-of-the-art alternatives.

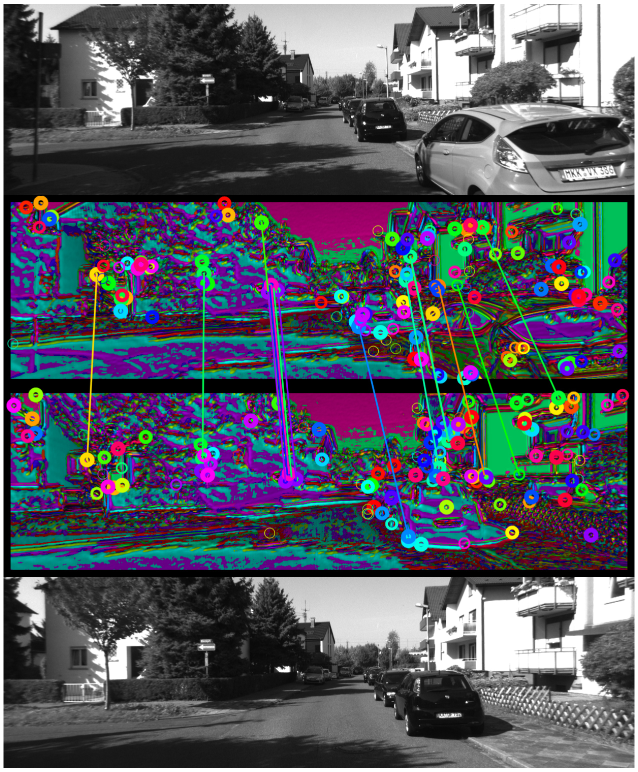

Figure 1. We propose a CNN interest point detector which provides implicitly matched interest points — descriptors are not needed for matching. This image illustrates the output of the final layer, which determines the interest points. Hue indicates which channel has the strongest response for a given pixel, and brightness indicates that response. Circles indicate the 128 interest points, which are the global maxima of each channel, circle thicknesses indicate confidence in a point. Lines indicate inlier matches after P3P localization.

1. Introduction

Many applications of computer vision, such as structure from motion and visual localization, rely on the establishment of point correspondences between images. Correspondences can be found densely [1–3], 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 handcrafted methods [4–6], but has more recently been solved using methods involving convolutional neural networks (CNNs) [7–9]. In this paper, we propose a novel approach to sparse feature matching that exploits 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. Essentially, 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). Then, 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 still 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. While we hypothesize the channel to specialize on visual features which are neither too common nor too rare, we observe the overall matching score to be slightly decreased as not every channel can result in a successful detection. Nevertheless, this is accounted for by providing a score for every interest point which indicates how likely it can be matched in an another frame.

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

2. Related Work

The components of a modern feature matching system can be best understood by considering the sub-problems, also in a historical context. From a computational perspective, it used to be prohibitive to calculate dense correspondences for images, so a first problem was to find a sparse set of points in the image that could be used for correspondence calculation. One could pick points at random, but if a point would land on a textureless region, there is no information surrounding it with which correspondence to an other image can be established. Thus, in a first interest point detector, [10] has identified points which are explicitly distinct from the points that surround them. Subsequently, faster approximations for distinctiveness have been found, whether using first-order approximations [11,12], or convolutional filters such as the Laplacian of Gaussian or Difference of Gaussian [4,5]. Alternatively, a detector can explicitly target a subset of distinctive points, such as corners and dots [13]. All of these methods calculate a "featureness" 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 has been extracted in the images, they need to be matched between each other to establish one-to-one correspondences. The simplest approach would be to match points whose surrounding image patches are most similar to each other, but this approach is very fragile to slight changes in illumination and viewpoint. To provide remedy to this variance, descriptors have been introduced. Descriptors are functions of patches, whose output is typically lower-dimensional, but invariant to slight illumination and viewpoint changes, 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) [4]. Another example are binary descriptors, which are particularly efficient to calculate [14,15].

Most descriptors, however, are still sensitive to strong changes in scale and orientation, or, more generally, affine transformations. A possible approach is to normalize image patches with respect to them, for example with absolute orientation and scale detection. Scale can be detected at the level of the interest point detector, by using explicit multiscale detection [15,16], while orientation can be determined as the direction with the strongest gradient [4] or with the notion of moments [6]. A wide variety of traditional feature matching systems comprising these components exist, see the survey in [17].

The recent success of convolutional neural networks (CNNs), however, has led the community to revisit the traditional methods used for feature matching. CNN-based methods can be used to replace every aforementioned component.

For interest point detection, the cornerness response traditionally calculated for the full image can be calculated using 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 [18], to present consistent ranking in the images in which they are extracted [19], to provide particularly sharp and thus unambiguous responses [20], or even to predict the probability of a certain pixel to result in an inlier [21]. A majority of these methods is compared in the recent survey [22].

CNNs are furthermore well suited for descriptors. Descriptors are functions defined on image patches, and CNNs are proven function approximators for just that. The output channels of a CNN can be interpreted as the coefficients of a descriptor. Not only has this been successfully used to replace traditional descriptors [23–29], but it has also proven very effective in other applications involving visual descriptors, such as place recognition [30]. A comparison of traditional and learned descriptors is provided in [31].

Finally, CNNs have also been shown to be useful for spatial normalization [32] or affine region detection [33].

While the results of the aforementioned surveys comparing CNN-based methods to traditional methods do not yet suggest absolute superiority of CNN-based methods over traditional methods, the clear advantage of CNN-based methods is that they are malleable: on the one hand, 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. On the other hand, 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 [9] and SuperPoint [8]. LF-Net [9] builds on top of a previous method by the same authors, LIFT [7], the first such system, in which the method was trained on a set of pre-extracted patches. [9], instead, is trained in an unsupervised and more unconstrained manner, only requiring an image sequence with ground truth depths and poses. Like [7], [9] uses separate CNNs for multi-scale interest point detection, feature orientation estimation and feature description. In contrast, SuperPoint [8] 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. What both of these approaches have in common is that they 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 [34], the authors replace descriptors with word identifiers of the corresponding visual word in a Bag-of-Words visual vocabulary [35]. 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 [36], 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. From each channel i, we extract the argmax as i-th interest point with coordinates $\boldsymbol{c}_i$. 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 $\boldsymbol{c}_i$ from image I is matched with point $\boldsymbol{c}'_i$ 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 $(\boldsymbol{c}_i, \boldsymbol{c}'_i)$ and determines which of them are inliers. This relies on ground truth correspondences $\Psi(\boldsymbol{c}_i)$. Interest points, correspondences and inlier labels shape mini-batches that contribute to the loss for a given training step as described in Section 4. Note that the system is not back-propagated in an end-to-end fashion. It resembles the systems in [9, 21], which use similar training methodologies that are not fully differentiable, for similar purposes. During evaluation and deployment, the inlider 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 [37] localization using RANSAC [38], which produces a relative pose estimate. We compare this pose estimate to the ground truth relative pose to assess the viability of our method for feature matching in a visual localization setting.

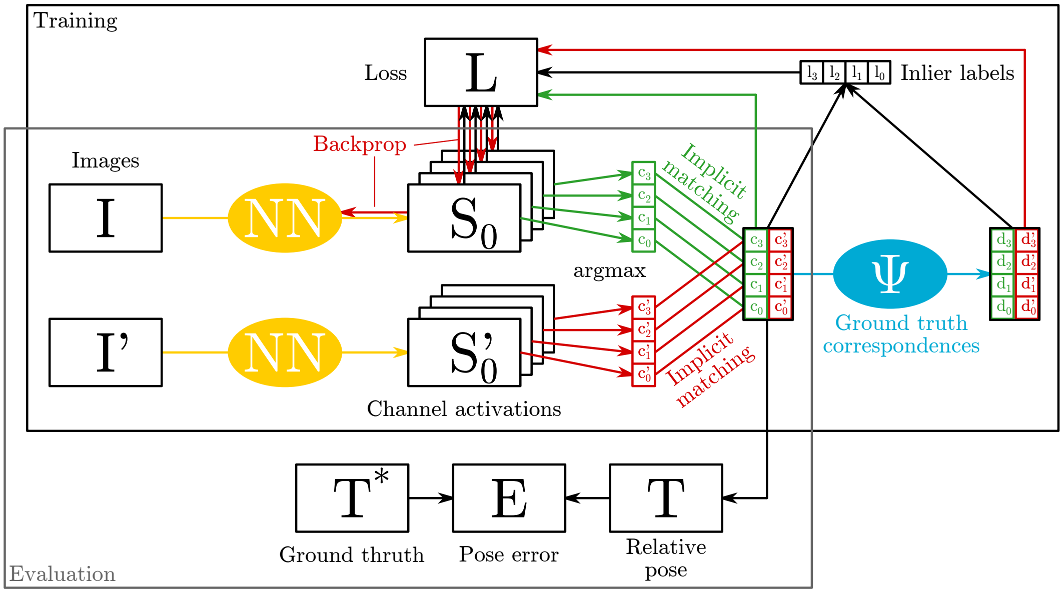

Figure 2. Overview of the proposed implicit interest point matching approach. Given an input image a convolutional neural network computes n activations which are then transformed to interest point coordinates by means of a spatial argmax. The interest points from two different images are then matched merely by their channel id. During training, these correspondences are labeled as *inliers*, *outliers*, or *unassigned* by comparing against ground truth correspondences. The formulated loss then promotes inliers, penalizes outliers, and suppresses redundancy. Finally, at test time, the same correspondences can be used to compute a relative pose between two image frames which can be compared against ground truth for performance evaluation.

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 matches as *inliers* if the correspondence of an interest point $\Psi(\boldsymbol{c}_i)$ is within $3px$ of the matched interest point in the other image $\boldsymbol{c}'_i$ and vice versa. Otherwise, the match is either labeled an *outlier* if both correspondences lie somewhere else within the respective image, or *unassigned* if one of the correspondences is outside the image frame. In some datasets, a method to compute ground truth correspondence is provided. If such a method is not directly provided, correspondence can alternatively be calculated from ground truth depth and pose [19]. In case these are not provided either, the correspondence could be estimated using an SfM algorithm [9] or, alternatively, by using Lucas-Kanade Tracking (KLT) [39], which is arguably the most general way to obtain correspondences and even applies to uncalibrated image sequences. In the present paper, we use KLT and obtain correspondences in pairs extracted from image sequences by tracking points from one image to the other, using strict bi-directional checking to obtain only high-quality correspondences. In KLT, only image regions with sufficient texture are tracked. However, image regions without texture are not useful interest point candidates, so this works in our favor.

4. Training Methodology

We use standard iterative training using Adam training updates [40]. In each iteration, a training sample, which consists of two images of the same scene is forward passed through the system to obtain a set of matches $\{(\boldsymbol{c}_i, \boldsymbol{c}'_i), i \in \{0, \dots, n-1\}\}$ and an associated set of true correspondences $\{(\boldsymbol{d}_i, \boldsymbol{d}'_i) = (\Psi(\boldsymbol{c}_i), \Psi^{-1}(\boldsymbol{c}'_i))\}$ 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* $\boldsymbol{P}(\boldsymbol{c}_i)$ centered around $\boldsymbol{c}_i$ and shaped according to the receptive field, $r \times r$, of a single pixel at the output. The other from stacking *correspondence patches* $\boldsymbol{P}(\boldsymbol{d}'_i)$ centered around the correspondences $\boldsymbol{d}'_i$. 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 the case $n = 3$. Note that the diagonal in the first tensor contains the responses of patch $\boldsymbol{P}(\boldsymbol{c}_i)$ at channel i, which per definition is the maximum value in channel i and the value that caused $\boldsymbol{c}_i$ to be selected as interest point. Similarly, the diagonal in the second tensor contains the responses that *should* be the maximum in the

chn.	status	$\boldsymbol{P}(\boldsymbol{c}_0)$	$\boldsymbol{P}(\boldsymbol{c}_1)$	$\boldsymbol{P}(\boldsymbol{c}_2)$	$\boldsymbol{P}(\boldsymbol{d}_0')$	$\boldsymbol{P}(\boldsymbol{d}_1')$	$\boldsymbol{P}(\boldsymbol{d}_2'))$
0	inlier	↑					n/a
1	outlier	↓	↓			↑	n/a
2	unass.	↓					n/a

Table 1. An overview of the mini-batch and how losses are applied to it. Depending on whether a channel is defined as *inlier*, *outlier*, or *unassigned*, different losses are applied to shape the activations of the patches. For inliers, the activation of the maxima is strengthened while at the same time suppressing the activation in the other channels. For outliers, the activation of the maxima is weakened while promoting the loss of the warped correspondence in order to facilitate the emergence of matches during training.

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'.

4.1. Inlier Reinforcement

For inlier reinforcement and outlier suppression, we use the probabilistic loss put forward in [21]. In short, if $l_i \in \{true, false\}$ is the random variable associated with a specific channel i and image coordinate $\boldsymbol{c}_i$ being an inlier, then we approximate its probability distribution to be only conditioned on the content of the image patch centered at $\boldsymbol{c}_i$: $p_i(l_i|\boldsymbol{P}(\boldsymbol{c}_i))$, which we choose to be the output of our convolutional neural network. Given that the process of extracting our interest points (argmax), is also dependent on the network output, we arguably introduce a circular dependency. This, however, only means that there is no objective "ground truth" for p_i; instead, the system converges to a self-consistent state. See [21] for a thorough discussion of this and the mathematical derivation of the loss. The loss is derived from a steady-state consideration, where at equilibrium the effect of inliers and outliers on the predicted loss should negate one another. This leads to the binary cross-entropy loss:

$$L_{inl}(p_i, l_i) = \begin{cases} -\log(p_i), & l_i = true, \\ -\log(1-p_i), & l_i = false. \end{cases} \tag{1}$$

In Table 1 the inlier reinforcement effect is observed for channel 0 on $\boldsymbol{P}(\boldsymbol{c}_0)$ and the outlier suppression effect is present for channel 1 on $\boldsymbol{P}(\boldsymbol{c}_1)$.

4.2. Redundancy Suppression

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. Thus, given that we already have an inlier interest point and that we require that two channels can not have interest points at the same location, we treat all other channels as outliers by applying the same suppression loss as above:

$$L_{red}(p_i) = -\log(1-p_i). \tag{2}$$

In Table 1 the redundancy suppression is present on channels 1 & 2 on $\boldsymbol{P}(\boldsymbol{c}_0)$. Empirically, we found that without redundancy suppression, all channels tend to converge to a single feature, all selecting the same interest point in every image.

4.3. Correspondence Reinforcement

Finally, we have found that our network does not converge with the above losses alone. This observation is corroborated by [9], where the authors have found that ground truth correspondence has been needed to converge during training. To this end, all responses at corresponding patches that correspond to outliers are reinforced with the same loss as in inlier reinforcement:

$$L_{cor}(p_i) = -\log(p_i). \tag{3}$$

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

Note that in contrast to [21], the addition of the suppression and correspondence losses potentially introduce an imbalance between reinforcement and suppression that might not result in correct inlierness prediction. Indeed, we show in the results in Figure 10 that the network response is not equal to the inlierness probability of a given point. Still, the relation between them is almost perfectly linear.

4.4. Training Pair Selection

In datasets such as HPatches [41], pairs are pre-selected. For image sequences, we adopt the training pair selection from [21]: Given one image, points are densely sampled and subjected to KLT tracking for as far into the subsequent images as possible. A pair can now be formed between this initial image and any 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 $o = 0.5$. The second image is randomly selected from within the admissible range.

5. Experiments

Like [21], we subject our method to evaluation and comparison with state-of-the-art on three datasets: The viewpoint-variant part of the wide-baseline HPatches benchmark [22,41], sequence 00 of the outdoor autonomous driving dataset KITTI [42] and sequence V1_01 of the indoor drone dataset EuRoC [43]. For the HPatche 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 [44], where we use the rectified images of sequences `01`, `02`, `03` (indoors) and `48`, `49`, `50` (outdoors) as uncalibrated image sequences. For all datasets, we evaluate matching score. For the two sequences (KITTI and EuRoC), 100 image pairs are randomly selected using the method in Section 4.4. Ground truth pose is provided in these sequences, so we also evaluate pose estimation accuracy. To that end, the true correspondence-based inlier determination is replaced with inlier determination resulting from applying P3P [37] and RANSAC [38]. As a consequence, the matching score here reflects the fraction of RANSAC inliers.

5.1. Baselines

We compare our method to SIFT [4], SURF [5], LF-Net [9] and SuperPoint [8]. For SIFT and SURF, 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 equal interest point count as IMIPs, $n = 128$, and at a more native interest point count of $n = 500$.

5.2. Pose Estimation Accuracy

We evaluate how a localization system, or SLAM system during loop closure, would fare with our feature matching method by assessing its effect on P3P localization [37] with RANSAC [38]. To simulate this in the most straightforward way, we have evaluated our method on image pairs selected from stereo datasets. For one of the images, we find the depths of the IMIPs interest points using epipolar stereo matching. We then localize the other image with respect to the resulting, implicitly matched sparse point cloud.

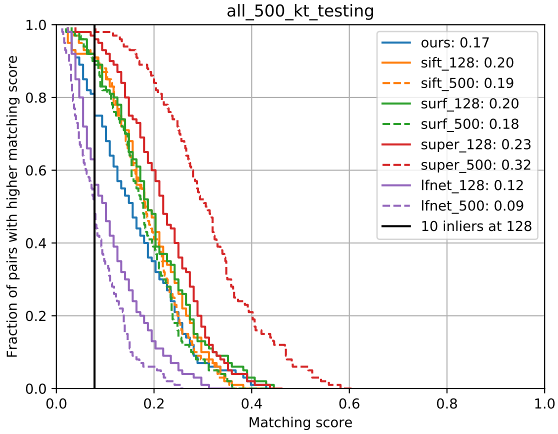

Figure 3. Matching score distribution on KITTI (higher is better). The average matching score is reported in the legend.

We compare the resulting estimate to the ground truth and report rotational and translational error using the geodesic (angle in angle-axis) and euclidean distances. Since different image pairs will have a different associated inlier count, we plot these errors with respect to that inlier count. This should give an intuition of what inlier count threshold to choose to reject poor estimates after RANSAC.

6. Results

Figure 3 shows the matching score distribution for the image pairs evaluated on KITTI 00. As we can see, several baselines exhibit higher matching scores than our method. Consider, however, that these baselines each represent features with descriptors with hundreds of floating point coefficients. Our method, in contrast, establishes matches without any descriptors. This, we believe, justifies a somewhat worse performance in terms of matching score, particularly in applications where memory and bandwidth are relevant.

Figure 4 shows rotation and translation errors as a function of the inlier count. We can see that good relative poses are established beyond 10 inliers. The quality of these poses is very similar to poses estimated using SIFT features, see Figures 5 and 6. We transfer the insight that good relative poses can be established with 10 inliers, back to the matching score distribution in Figure 3 by adding a vertical line at the matching score corresponding to 10 inliers. From the intercept with our curve, we can see that 80% of image pairs, randomly sampled with an image overlap between 0.5 and 1.0, result in good relative pose estimates. A few of these relative poses might still exhibit errors of tens of centimeters, but modern localization and SLAM systems typically integrate the relative pose estimates of several frames to smooth out the error between them.

In Figure 7 and Figure 8 we evaluated matching score

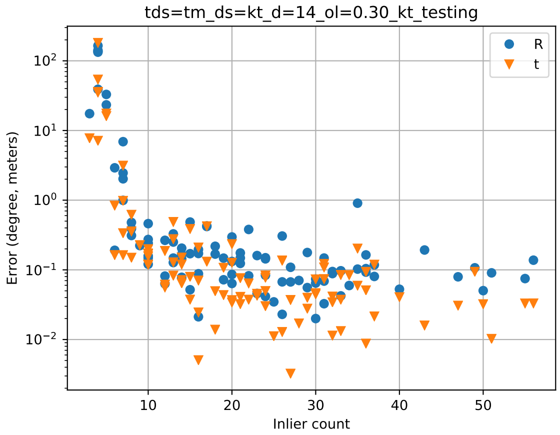

Figure 4. Pose error distribution as a function of inlier count of our method on KITTI.

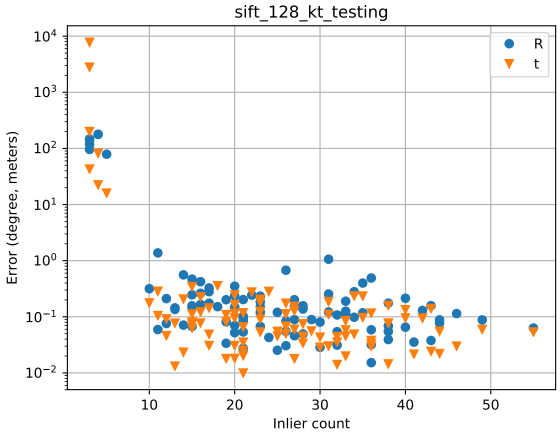

Figure 5. Pose error distribution of SIFT using the 128 top scored points on KITTI.

and pose estimation on EuRoC. Here our approach seems to compete better with the baseline and only Superpoint seems to remain clearly ahead in terms of matching score. The pose error comes with a similar character again, although slightly more inliers are required to obtain good performance.

As we can see in Figure 9, the wide baselines and scale changes of HPatches pose a challenge to our currently single-scale method. Future work to address the challenges of wide baselines could include a stronger focus on considerations such as scale and rotation invariance, whether by explicit modeling inside the neural network architecture [9] or by implicit means, such as careful data augmentation and curricular learning [45]. In the meantime, our method provides a representation of locations for metric localization at narrow baselines that has unprecedented compactness.

Finally, we provide some statistics to better understand

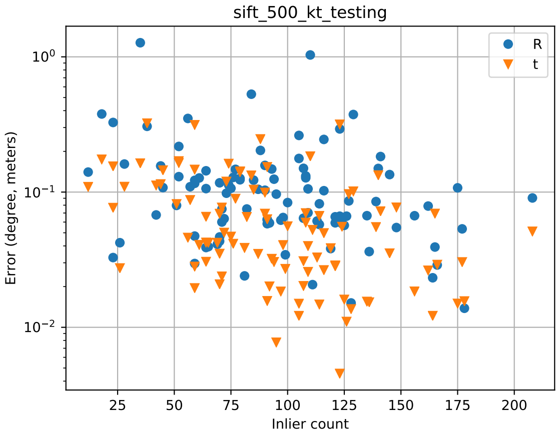

Figure 6. Pose error distribution of SIFT using the 500 top scored points on KITTI.

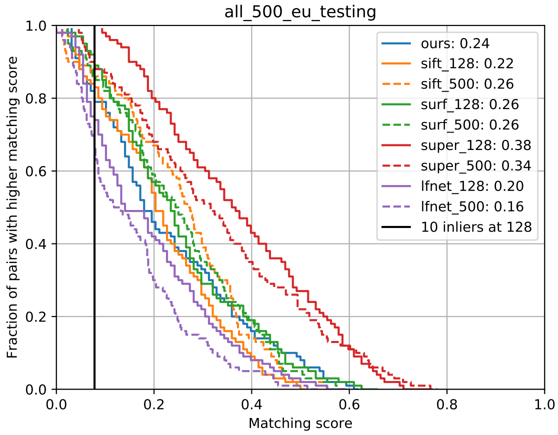

Figure 7. Matching score distribution on EuRoC.

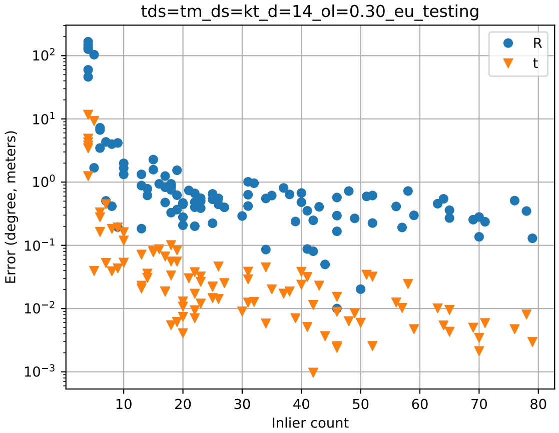

Figure 8. Pose error distribution of our method on EuRoC.

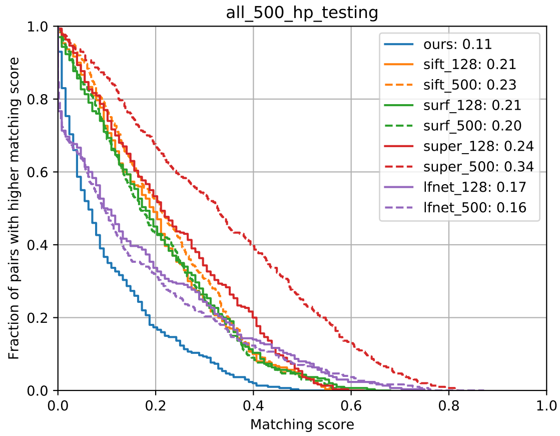

Figure 9. Matching score distribution on HPatches.

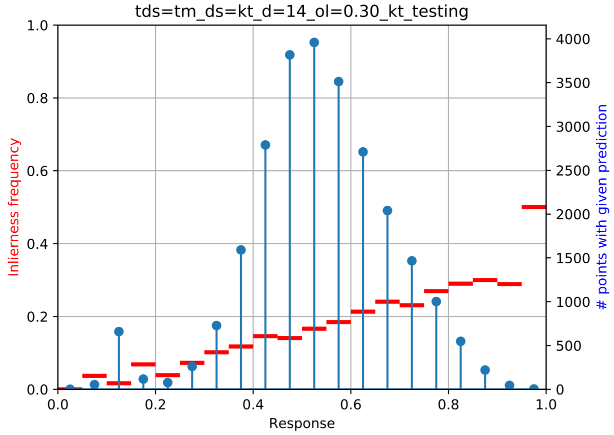

Figure 10. Inlierness frequency versus response, and histogram of interest point responses across the testing dataset. Inlierness frequency is calculated per histogram bin by dividing the amount of inliers within a response range by the total amount of interest points within that response range.

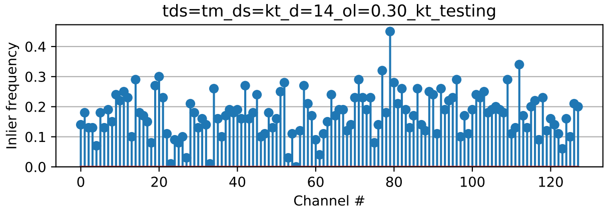

Figure 11. Distribution of inliers among channels. Inlier detection is distributed relatively evenly, with few exceptions at the higher and lower ends.

our network in Figures 10 and 11. As we can see in Figure 10, the response of an interest point is monotonic with respect to its probability of being an inlier. The response does not predict the probability as nicely as in [21], presumably because the introduction of suppression and in particular correspondence loss disturbs the balance between inlier and outlier loss. Still, it could potentially be used to accelerate RANSAC by preferentially sampling interest points with a higher response. Figure 11 confirms that suppression and correspondence loss work well in terms of well distributing good features among the channels of the neural network. There is only a very small number of channels that contribute almost no inliers. This might simply reflect a different distribution of features between the training and testing datasets.

7. Conclusion

In this paper, we have introduced a descriptor-free approach for detecting sparse visual features and matching them between images. We rely on a convolutional neural network to predict multiple activation layers and we define the location of the globally 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, which enables implicit matching. 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. Our method thus also unlocks the potential to significantly boost scalability in applications such as large-scale localization and multi-agent SLAM.

We have devised a methodology to train the introduced architecture that circumvents non-differentiable operations by formulating a reinforcement loss on the activation layers directly. While this relies on the availability of ground truth correspondences, these correspondences can be obtained from unlabeled and even uncalibrated image sequences using traditional direct tracking. Our training is based on a probabilistic methodology which has the effect that interest points can be ranked by their response, which will also rank them by the probability for them to yield inliers. In various experiments on multiple datasets, we have shown that we can successfully train our network to detect interest points. 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.

Acknowledgements

This research was funded by the National Center of Competence in Research (NCCR) Robotics through the Swiss National Science Foundation and the SNSF-ERC Starting Grant.

References

[1] B. K. Horn and B. G. Schunck, "Determining optical flow," *J. Artificial Intell.*, pp. 185 – 203, 1981. 1

[2] G. Farnebäck, "Two-frame motion estimation based on polynomial expansion," in *Image Analysis*, pp. 363–370, 2003. 1

[3] I. Rocco, M. Cimpoi, R. Arandjelovic, A. Torii, T. Pajdla, and J. Sivic, "Neighbourhood consensus networks," *arXiv e-prints*, Oct. 2018. 1

[4] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, pp. 91–110, Nov. 2004. 1, 2, 6

[5] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," *Comput. Vis. Image. Und.*, vol. 110, no. 3, pp. 346–359, 2008. 1, 2, 6

[6] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Int. Conf. Comput. Vis. (ICCV)*, 2011. 1, 2

[7] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua, "LIFT: Learned invariant feature transform," in *Eur. Conf. Comput. Vis. (ECCV)*, pp. 467–483, 2016. 1, 3

[8] D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperPoint: Self-supervised interest point detection and description," in *IEEE Conf. Comput. Vis. Pattern Recog. Workshops (CVPRW)*, June 2018. 1, 3, 6

[9] Y. Ono, E. Trulls, P. Fua, and K. M. Yi, "LF-Net: Learning local features from images," *arXiv e-prints*, May 2018. 1, 3, 4, 5, 6, 7

[10] H. P. Moravec, *Obstacle Avoidance and Navigation in the Real World by Seeing Robot Rover*. PhD thesis, Carnegie-Mellon University, Pittsburgh, Pennsylvania, Sept. 1980. 2

[11] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. Fourth Alvey Vision Conf.*, vol. 15, pp. 147–151, 1988. 2

[12] J. Shi and C. Tomasi, "Good features to track," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 593–600, June 1994. 2

[13] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Eur. Conf. Comput. Vis. (ECCV)*, pp. 430–443, 2006. 2

[14] 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. Machine Intell.*, vol. 34, no. 7, pp. 1281–1298, 2012. 2

[15] S. Leutenegger, M. Chli, and R. Siegwart, "BRISK: Binary robust invariant scalable keypoints," in *Int. Conf. Comput. Vis. (ICCV)*, pp. 2548–2555, Nov. 2011. 2

[16] K. Mikolajczyk and C. Schmid, "Scale & affine invariant interest point detectors," *Int. J. Comput. Vis.*, vol. 60, pp. 63–86, 2004. 2

[17] D. Mukherjee, Q. M. JonathanWu, and G. Wang, "A comparative experimental study of image feature detectors and descriptors," *Machine Vision and Applications*, vol. 26, pp. 443–466, May 2015. 2

[18] K. Lenc and A. Vedaldi, "Learning covariant feature detectors," in *Eur. Conf. Comput. Vis. Workshops (ECCVW)*, pp. 100–117, 2016. 3

[19] N. Savinov, A. Seki, L. Ladický, T. Sattler, and M. Pollefeys, "Quad-networks: Unsupervised learning to rank for interest point detection," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2017. 3, 4

[20] L. Zhang and S. Rusinkiewicz, "Learning to detect features in texture images," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018. 3

[21] T. Cieslewski, K. G. Derpanis, and D. Scaramuzza, "SIPs: Succinct interest points from unsupervised inlierness probability learning," *available on arXiv soon*. 3, 5, 6, 8

[22] K. Lenc and A. Vedaldi, "Large scale evaluation of local image feature detectors on homography datasets," in *British Machine Vis. Conf. (BMVC)*, 2018. 3, 6

[23] X. Han, T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg, "MatchNet: Unifying feature and metric learning for patch-based matching," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2015. 3

[24] S. Zagoruyko and N. Komodakis, "Learning to compare image patches via convolutional neural networks," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 4353,4361, 2015. 3

[25] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer, "Discriminative learning of deep convolutional feature point descriptors," in *Int. Conf. Comput. Vis. (ICCV)*, 2015. 3

[26] A. Mishchuk, D. Mishkin, F. Radenovic, and J. Matas, "Working hard to know your neighbors margins: Local descriptor learning loss," in *Conf. Neural Inf. Process. Syst. (NIPS)*, pp. 4826–4837, 2017. 3

[27] A. Loquercio, M. Dymczyk, B. Zeisl, S. Lynen, I. Gilitschenski, and R. Siegwart, "Efficient descriptor learning for large scale localization," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, pp. 3170–3177, 2017. 3

[28] X. Wei, Y. Zhang, Y. Gong, and N. Zheng, "Kernelized subspace pooling for deep local descriptors," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018. 3

[29] M. Keller, Z. Chen, F. Maffra, P. Schmuck, and M. Chli, "Learning deep descriptors with scale-aware triplet networks," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018. 3

[30] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN architecture for weakly supervised place recognition," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 5297–5307, June 2016. 3

[31] J. L. Schonberger, H. Hardmeier, T. Sattler, and M. Pollefeys, “Comparative evaluation of hand-crafted and learned local features,” in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 1482–1491, July 2017. 3

[32] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, “Spatial transformer networks,” in *Conf. Neural Inf. Process. Syst. (NIPS)*, pp. 2017–2025, 2015. 3

[33] D. Mishkin, F. Radenović, and J. Matas, “Repeatability is not enough: Learning affine regions via discriminability,” in *Eur. Conf. Comput. Vis. (ECCV)*, 2018. 3

[34] D. Tardioli, E. Montijano, and A. R. Mosteo, “Visual data association in narrow-bandwidth networks,” in *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, pp. 2572–2577, Sept. 2015. 3

[35] J. Sivic and A. Zisserman, “Video Google: a text retrieval approach to object matching in videos,” in *Int. Conf. Comput. Vis. (ICCV)*, 2003. 3

[36] S. Lynen, T. Sattler, M. Bosse, J. Hesch, M. Pollefeys, and R. Siegwart, “Get out of my lab: Large-scale, real-time visual-inertial localization,” in *Robotics: Science and Systems (RSS)*, July 2015. 3

[37] X.-S. Gao, X.-R. Hou, J. Tang, and H.-F. Cheng, “Complete solution classification for the perspective-three-point problem,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 25, pp. 930–943, Aug. 2003. 3, 6

[38] M. A. Fischler and R. C. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981. 3, 6

[39] B. D. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” in *Int. Joint Conf. Artificial Intell. (IJCAI)*, pp. 674–679, 1981. 4

[40] D. P. Kingma and J. L. Ba, “Adam: A method for stochastic optimization,” 2015. 4

[41] V. Balntas, K. Lenc, A. Vedaldi, and K. Mikolajczyk, “HPatches: A benchmark and evaluation of handcrafted and learned local descriptors,” in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2017. 5, 6

[42] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: The KITTI dataset,” *Int. J. Robot. Research*, vol. 32, no. 11, pp. 1231–1237, 2013. 6

[43] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, “The EuRoC micro aerial vehicle datasets,” *Int. J. Robot. Research*, vol. 35, pp. 1157–1163, 2015. 6

[44] J. Engel, V. Koltun, and D. Cremers, “Direct Sparse Odometry,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PP, no. 99, pp. 1–1, 2017. 6

[45] D. DeTone, T. Malisiewicz, and A. Rabinovich, “SuperPoint: Self-supervised interest point detection and description,” *arXiv e-prints*, Dec. 2017. 7

8. Supplementary Material

8.1. CSV files

As written in the paper, a detailed CSV file is attached for each of the two robotic testing dataset. The files can be found on the arXiv abstract page under “Ancillary files”. They contain the columns:

- `name`: sequence name and the indices of the two images that form the pair.
- `dR, dt`: ground truth difference in rotation and translation between the two image frames.
- `matching score`: The matching score of our method for the given pair.
- `eR, et`: rotation and translation *error* of the relative pose estimate.

Both `dR` and `eR` are measured with the geodesic distance, in degrees, which corresponds to the angle of the angle-axis representation of the relative rotation.

8.2. True pose difference plots

To provide an intuition for the robotic testing sets, we plot the distribution of true relative poses in Figure 12 Note the difference between the two datasets: KITTI exhibits larger distances, while EuRoC exhibits larger orientation differences. Recall that given the first image, the second image is randomly sampled among subsequent images with a scene overlap of at least 50%.

Besides showing the distribution, this plot also indicates which pairs result in 10 or more P3P RANSAC inliers. As shown in the paper, these result in a good relative pose estimate. While the pairs that fail to obtain a good relative pose estimate are typically the ones with larger pose differences, there is no clear boundary between pairs that succeed and pairs that fail.

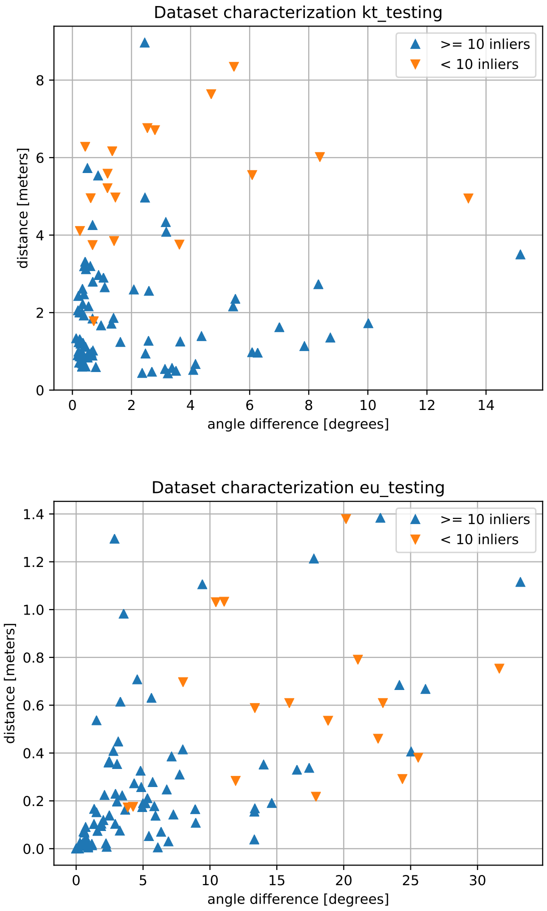

Figure 12. True pose difference plots for KITTI (top) and EuRoC (bottom). Markers indicate whether or note more than 10 P3P RANSAC inliers have been achieved for a given pair. As shown in the paper, this is a good inlier threshold to accept or reject relative pose estimates.