SIPS: Unsupervised Succinct Interest Points

Titus Cieslewski, Davide Scaramuzza

Robotics and Perception Group, University of Zurich and ETH Zurich

Abstract. Detecting interest points is a key component of vision-based estimation algorithms, such as visual odometry or visual SLAM. Classically, interest point detection has been done with methods such as Harris, FAST, or DoG. Recently, better detectors have been proposed based on Neural Networks. Traditionally, interest point detectors have been designed to maximize repeatability or matching score. Instead, we pursue another metric, which we call succinctness. This metric captures the minimum amount of interest points that need to be extracted in order to achieve accurate relative pose estimation. Extracting a minimum amount of interest points is attractive for many applications, because it reduces computational load, memory, and, potentially, data transmission. We propose a novel reinforcement- and ranking-based training framework, which uses a full relative pose estimation pipeline during training. It can be trained in an unsupervised manner, without pose or 3D point ground truth. Using this training framework, we present a detector which outperforms previous interest point detectors in terms of succinctness on a variety of publicly available datasets.

Fig. 1. State-of-the art interest point detectors are designed to yield a large amount of repeatable interest points. Instead, we propose to train a network to detect a minimum amount of interest points that are sufficient to obtain accurate relative pose estimates. Previous methods typically extract several hundred interest points for pose estimation. In contrast, our detector extracts less than one hundred interest points. The per-pixel score produced by our detector is shown on the right. Circles indicate extracted interest points (inliers in green, outliers in red). Green lines indicate correct matches.
1 Introduction

Visual odometry (VO) and visual simultaneous localization and mapping (VSLAM) are essential ingredients for any robotic system that aims to visually navigate through an unstructured environment without external help. Examples of state-of-the-art VO/VSLAM systems are [12,34] or systems that additionally make use of an inertial measurement unit (IMU) such as [5,6,7,8,9]. What most of them have in common is that they extract, as a first step, a sparse set of interest points from the images that they process. These interest points are then used for pose estimation using algorithms such as the eight-point [10], five-point [11], or P3P [12] algorithm. As a consequence, the interest points must be repeatable: if an interest point is detected in one image, it should also be detected in any other image observing the same scene. Popular metrics to measure this property under different conditions are repeatability and the matching score [13,14]. They are prominently used in state of the art feature detectors and descriptors [15,16,17,18].

In this work, we pursue an alternative metric, which we call succinctness. Repeatability tends to depend on the amount of points that are extracted: the more points are extracted, the more likely it is to have correct correspondences between interest points detected in different images, and thus the higher repeatability tends to become [16,17]. With succinctness, we specifically explore how few points can be extracted in order to still obtain accurate pose estimates. While repeatability still matters, it is not important that it applies to as many points as possible; instead, it is sufficient that only a small amount of interest points is repeatable, as long as they have consistently the highest rank in different images.

Having as few interest points as possible has many advantages in VO and VSLAM. Firstly, it reduces the computational cost of many algorithms: triangulation, bundle adjustment, relative pose estimation after loop closure, etc., all run faster with less extracted interest points. Secondly, it similarly reduces the memory footprint of representing large maps. Finally, it reduces the bandwidth requirements of multi-robot VSLAM systems [19,20,21]. In all of these applications, it is furthermore important that the interest points are sufficiently distributed in the image, otherwise they result in a degenerate relative pose estimate.

To achieve succinctness, we present an unsupervised training method which reinforces succinct interest points selected by a convolutional neural network (CNN). Firstly, it reinforces interest points that are correctly matched in structure-from-motion (SfM). Secondly, it uses a rank loss as proposed in [17] to ensure that correct matches are preserved even if the number of extracted points is reduced (see Fig. 3). While previous work would obtain matching patches from ground truth labels or synthetic warping, we instead use SfM in the training loop to determine correct matches from consistency only. This allows training from unlabeled image sequences. It also allows us to inherently capture SfM-specific interest point requirements in the detector training, including requirements of the used descriptor. The resulting CNN outperforms previous interest point detectors with respect to succinctness on real world datasets, even though
succinctness is not expressed in the loss function. The code to extract SIPS and reproduce our results will be made publicly available.

1.1 Contributions

To sum up, the contributions of this paper are:

1. The introduction of the succinctness metric which allows to quantify how few interest points can be extracted with a given interest point detector while achieving an accurate pose estimate based on these interest points.
2. An unsupervised interest point detector training framework and a resulting detector that achieves high succinctness. In contrast to previous training methods, SfM is part of the training loop. This allows the training to account for SfM needs, while training on real data. Furthermore, the loss is formulated in a way that requires neither pose nor 3D point ground truth.

2 Related Work

Classically, interest points have been selected from distinctive image locations, that is, image locations which significantly differ from neighboring image locations. Whether an image location is distinctive can be determined explicitly [22] or using a first-order approximation such as the Harris [23] or Shi-Tomasi [24] detector. Alternatively, distinctive interest points can be detected by convolving the image with a suitable kernel, such as the Laplacian of Gaussian (LoG), or its faster approximation, the Difference of Gaussians (DoG) kernel, prominently used in [25]. Other filter-based detectors have been proposed by [26,27]. Finally, another method for detecting interest points is to explicitly identify image regions that resemble corners. Iteratively, gradually more efficient ways to do this have been proposed by [28,29,30] and have found widespread popularity with [31] due to its highly efficient implementation. An interesting challenge in interest point detection is ensuring that the detection is independent of scale and affine transformations. Scale invariance can be achieved using multi-scale detection [32], while affine invariance, in particular invariance to rotation is already given for most aforementioned detectors.

The problem with relying on distinctiveness as interest point selection criterion is that it does not necessarily result in high repeatability, unless all distinctive points are selected. A more refined interest point selection criterion is needed if one wants to preserve repeatability when extracting less points. Some of the aforementioned works have attempted to derive such a criterion based on models or heuristics. An alternative and more promising method to deriving this criterion is data-driven, using machine learning. An early method based on a neural network has been proposed in [33], though it was limited to three layers due to the computational constraints of the time, and also only applied to the edge regions of an image. Subsequently, [34,35,36] used other types of regression functions to learn detectors.
Recently, [15] proposed a fully trained deep neural network pipeline for feature detection and description. Their pipeline comprises a network for interest point detection, a network for estimating the orientation of the feature, and a network for the description of that feature. Their training relies on obtaining image patches as input, which are obtained from a prior detector. While their detector network does not explicitly imitate the prior detector, it is not clear whether this pre-selection does not limit the points that can be detected with the learned detector. Furthermore, no mechanism is in place to ensure repeatability also when few points are extracted. An interest point detector which is trained using a covariance constraint, a loss function that ensures that the detector response is covariant with image transformations, has been proposed by [16]. In contrast to [15], this should impose less limits on what points can be trained to be detected, but also here there is no mechanism to account for repeatability at low interest point counts.

More recently, [17] discovered the very useful properties of rank loss, which we adopt and describe in detail in Section 5. In short, they showed that if scores for patches of corresponding points are consistently ranked between two images, useful interest points automatically emerge from the highest- and lowest-ranked points. From this, they derived a rank loss with which they trained networks to provide such a score. The networks have been trained either with data containing ground truth correspondences or with synthetically warped data. While we believe that rank loss is a key to achieving succinctness, the authors have not evaluated their method at as low interest point counts as we target. Besides providing an evaluation for succinctness at very low interest point counts, we also extend the rank loss methodology of [17] to be trained with structure from motion (SfM) in the loop. This not only takes effects of SfM into account, but also allows the network to be trained without ground truth or synthetic warping. Most recently, [18] have proposed a three-step training methodology where the last step allows joint training of detector and descriptor. Also here, there is no mechanism to ensure repeatability at low interest point counts and also here, SfM is not part of the training loop.

All previously mentioned works as well as our work tackle interest point detection. In particular, our goal is to determine succinct interest points at detection time. An alternative approach would be to first extract and describe a large set of interest points, and then reduce them to a minimal set sufficient for localization in post-processing. Several approaches exist where this reduction is based on whether specific features are repeatably detected in multiple images observing the same scene [37,38,39,40,41]. Interestingly, learning to predict from single images whether already-extracted features are likely to be matched has been already proposed [42,43], but in both cases as a filtering step of already-extracted features. Instead, we directly identify (and meaningfully score) succinct interest points in the detector.
Fig. 2. We train a neural network (NN) to output a per-pixel score $S$ given an image $I$. The score $S$ is passed to non-maximum suppression (NMS) to extract $n$ interest points $P$. The interest points extracted from two images are passed to the pose estimation module (PE), which also contains interest point description and matching. The PE module produces the relative pose estimate $T^0_1$ and the inlier and outlier sets $I$ and $O$. During training, the loss $L_i$ of either image $I_i$ is obtained from $I, O$ and $S_i$, allowing for unsupervised learning (diagram omitted for $I_0$). The pose ground truth $T^0_1\ast$ is only used to evaluate rotation and translation errors $e_R$ and $e_t$.

3 System overview

The training of our detector is coupled to a Structure-from-Motion (SfM) pipeline. See Fig. 2 for an overview of this pipeline. Its goal is to establish the relative pose $T^0_1$ of two cameras $C_0, C_1$ given their images (of the same scene) $I_0, I_1$. As is done classically, we achieve this by extracting interest points $P$ from both images $I$ by calculating a per-pixel score $S$ based on which $n$ interest points are selected using non-maximum suppression (NMS). These points $P$ are matched based on descriptors that can be provided by any descriptor method. Then, a geometric algorithm with random sample consensus (RANSAC) [12] is used to find $T^0_1$. More specifically, we consider the case where the depth of the interest points $P$ for one of the two images is already known and use P3P RANSAC. This is most representative of the use cases of relative pose estimation in real visual odometry and SLAM systems, where 3D landmark positions need to be established to guarantee a consistent scale.

Given this pipeline, our goal is to minimize the amount $n$ of points $P$ that needs to be extracted from both images while still achieving a good relative pose estimate. In our pipeline, the part that can be trained is obtaining $S$ from $I$, which is achieved using a convolutional neural network (CNN). Our core contribution is the development of a methodology for training this CNN in such a way that the full system results in succinct interest points. The CNN architecture is detailed in Section 6.2.

4 Succinctness

Our goal is to achieve a good relative pose estimate $T^0_1$ with as small a number of extracted interest points as possible. This section is dedicated to the succinctness
metric which we use to quantify the degree to which an interest point detector is able to achieve this goal.

The goal statement encompasses three metrics: the error of $T_1^0$ with respect to ground truth, which we decompose into rotation error $e_R$ and translation error $e_t$, and the number of extracted points $n$. Different choices of $n$ result in different inlier sets $I$, which directly affect the estimate $T_1^0$. With P3P, the minimal inlier count $|I|$ for a unique solution is four, but requiring more inliers can lead to more accurate relative pose estimation.

We consider the required inlier count $k$ as a parameter and consider the following $k$-dependent performance metrics for an image pair: $n_k$, the minimum amount of points that need to be extracted from both images to obtain at least $k$ inliers, and $e_{R,k}$ and $e_{t,k}$, the rotational and translational errors achieved with $n_k$ interest points. Formally:

$$n_k = \min_{|I| \geq k} n, \quad e_{R,k} = e_R |n = n_k, \quad e_{t,k} = e_t |n = n_k.$$  \hspace{1cm} (1)

Since each metric will differ for different image pairs, we consider their distribution over a large set of image pairs as overall performance metric of the method. Hence, we randomly sample a predefined amount $l$ of image pairs $(I_0, I_1)$ from the datasets used for evaluation. For each image pair $(I_0, I_1)_i, i = 1..l$, we run the full pipeline with iteratively adapted interest point count until we find $n_k$. This gives us the values of $(1)$ for image pair $i$: $(n_{k,i}, e_{R,k,i}, e_{t,k,i})$. Then, we draw what fraction of samples has values $n_{k,i} < n$, as a function of $n$. We call this the succinctness curve:

$$s(n) = \frac{1}{l} \sum_{i \in [1,l]} \mathbb{1}(n \Rightarrow |I_i| \geq k)$$  \hspace{1cm} (2)

where $\mathbb{1}(P)$ is the indicator function, assuming value 1 if predicate $P$ is true and 0 otherwise. See Fig. 5 for examples of such curves. The performance criterion given by this curve is simple: the closer it approaches the top left corner, the better. To characterize this curve with a single number, we calculate the area under this curve (AUC) up to a maximum value $n_{\text{max}}$, and normalize by $n_{\text{max}}$. We refer to this number as AUC-$n_{\text{max}}$:

$$\text{AUC-}n_{\text{max}} = \frac{1}{n_{\text{max}}} \int_{n=0}^{n_{\text{max}}} s(n)dn = \frac{1}{n_{\text{max}}} \sum_{i \in [1,l]} \frac{1}{l}(n_{\text{max}} - n_{k,i})$$  \hspace{1cm} (3)

This function yields values between 0 and 1: 0 indicates that all sampled image pairs require more than $n_{\text{max}}$ interest points to achieve $|I| \geq k$, whereas 1 indicates the (impossible) ideal case, where all samples achieve $|I| \geq k$ by extracting $n = 0$ interest points. The errors $e_{R,i}$ and $e_{t,i}$ are summarized similarly, with the corresponding curves instead indicating the fraction of the samples with errors below a given $e_{R/t}$, and corresponding AUC metrics.

5 Training methodology

To describe our training methodology, we first describe our loss function assuming that the inlier and outlier sets $I, O$ are already given in Sections 5.1 and 5.2.
We then discuss the details of how inlier and outlier sets are determined, given an image pair, during training, in Section 5.3. Finally, we discuss how meaningful image pairs are selected from image sequences without relying on pose ground truth in Section 5.4.

5.1 Preliminaries

Let $p_{i,j} \in P_i$ be the $j$-th extracted interest point in image $I_i$. Then, the structure-from-motion (SfM) pipeline matches these points from the two images to obtain matches

$$\mathcal{M} = \{m = (p_{0,j}, p_{1,j})\}.$$  \hspace{1cm} (4)

Not all points in $P_i$ will be matched, and out of all matches, some will be matched correctly and others incorrectly. The structure from motion algorithm attempts to recognize correct and incorrect matches based on consistency and accordingly assigns matches to inlier and outlier sets $I \subset \mathcal{M}$ and $O \subset \mathcal{M}$, $I \cap O = \emptyset$. We henceforth use the following shorthands:

$$p_{i,j} \in I \Leftrightarrow \exists m \in I : p_{i,j} \in m$$ \hspace{1cm} (5)

indicates that an interest point belongs to an inlier match. We use the analogous shorthand for outliers. And:

$$m(p_{i,j}) = p_{i',j} : (p_{i,j}, p_{i',j}) \in \mathcal{M}$$ \hspace{1cm} (6)

indicates the point in the other image matched to $p_{i,j}$ and is only defined if such a matching exists.

5.2 Loss Function

Our loss is a function of the inlier and outlier sets $I, O$ and the scores $S$, see the red part in Fig. 2. It is defined and back-propagated for each of the two images separately, even though $I, O$ are defined for the image pair. The overall loss $L_i$ of image $I_i$ is composed of two components, the reinforcement loss $L_{e,i}(p)$ and the rank loss $L_{a,i}(p)$ which are in turn defined for each pixel $p \in P_i$.

The reinforcement loss reinforces interest points that result in inliers and punishes interest points that result in outliers. Formally, its value for pixel $p$ in image $I_i$ is:

$$L_{e,i}(p) = \begin{cases} 1 - S_i(p), & \text{if } p \in I \\ S_i(p), & \text{if } p \in O \\ 0, & \text{otherwise.} \end{cases}$$ \hspace{1cm} (7)

The intuition here is straightforward: we want to have, among our minimal, highest ranking interest point set, interest points which are most likely going to contribute to inliers.

The rank loss, which we adapt from [17], encourages that the interest points are ranked similarly with respect to the score $S$, between the images of the image
Motivation of the rank loss: Suppose the $k_1$ highest-ranked interest points from images 0 and 1 are all inliers that are matched among themselves. Applying only the reinforcement loss $L_{i,e}$ could result in the situation illustrated on the left, where matching interest points are not consistently ranked among the two images. If we now only extract the $k_2 < k_1$ highest-ranked points as interest points (above red line), we can end up in a situation where there is no correct correspondence between the images. The rank loss $L_{i,a}$ is designed to nudge the CNN towards the scenario depicted on the right, where matching points are consistently ranked. In the ideal case, correct correspondences would be obtained with an arbitrarily low $k_2$.

Formally, we denote the rank of an interest point selected in image $i$ as the mapping $r_i : P_i \rightarrow [1, n] \subset \mathbb{N}$:

$$r_i(p) < r_i(q) \Leftrightarrow S_i(p) > S_i(q) \quad \forall p, q \in P_i.$$ (8)

Accordingly, $r_i^{-1}(r)$ maps to the interest point in image $i$ with the $r$-th highest score. With this loss we encourage that corresponding inlier points are consistently extracted also when fewer points are extracted than during training. Why enforcing ranking should result in this is illustrated in Fig. 3. To encourage consistent ranking, thus, we add the following per-pixel loss:

$$L_{a,i}(p) = \begin{cases} (S_i(p) - S_i(r_i^{-1}(r_j(m(p)))))^2, & \text{if } p \in \mathcal{I} \\ 0, & \text{otherwise,} \end{cases}$$ (9)

with $r_j$ referring to the ranking in the other image, $(i, j) \in \{(0, 1), (1, 0)\}$. Put in words, the rank loss pulls the score of an inlier point $p$ towards the score of another point $q \in P_i$ in the same image, where $q$ is the point whose rank is the same as the rank of the match of $p$ in the other image.

With this, the overall loss, per image of a training pair, is:

$$L_i = \frac{1}{|\mathcal{I}| + |\mathcal{O}|} \sum_{\mathcal{I} \cup \mathcal{O}} L_{e,i}(p) + \frac{1}{|\mathcal{I}|} \sum_{\mathcal{I}} L_{a,i}(p).$$ (10)

This loss is backpropagated, for each image separately, using the Adam optimizer [44] at a step size of $10^{-5}$.

In order to achieve good relative pose estimation, interest points need to be well distributed in the image. This means that groups of interest points should not be concentrated around local maxima. While this is taken care of by non-maximum suppression (NMS) during deployment, it would be desirable that the
scores $S$ themselves do not exhibit large contiguous regions with similar, high values. We have found that we do not need to formulate an explicit loss for this, as it is implicitly taken care of by the fact that NMS is part of the pipeline that ultimately generates the inlier and outlier sets used by the loss. This observation is quantitatively supported by our results (Figs. 4) and qualitatively supported by examples of the network output, such as the ones in Fig. 1 or in Section 9.4 of the supplementary material.

5.3 Obtaining the inlier and outlier sets

In order to obtain the inlier and outlier sets $I, O$, a forward pass of the full pose estimation pipeline is performed, with the current state of the neural network in the loop. Experimental evidence shows that this is able to bootstrap from random weights, provided the extracted interest point count $n$ is high enough. We have found 500 to be a good value for $n$ during training. As for the NMS radius, we have set it to 10 or 15 pixels depending on the dataset – generally, as large as possible without suppressing too many high-scored pixels. The interest points are described using SURF descriptors. We choose SURF because we have found it to provide a good trade-off between computing speed and matching accuracy. However, any other descriptor could be used here – potentially including one that is learned together in the detector as in [15,18]. In this work, however, we focus on interest point detection only. Based on the descriptors, interest points are matched using the OpenCV BFMatcher. Given $M$, we use either of two methods to obtain $I$ and $O$:

The P3P method The preferred method – because it most resembles the envisioned use of the system – is to obtain $I$ and $O$ from P3P RANSAC. This is achieved using the standard implementation in OpenCV. The drawback of this method is that it requires the depths of points $P$ in one of the images. We obtain these using stereo matching for stereo datasets.

The KLT method Without any means to obtain the depths of $P$, one should theoretically be able to obtain $I, O$ from five-point or eight-point RANSAC using the point-to-epipolar line distance as inlier criterion. Unfortunately, we experienced difficulties with this, obtaining either only a subset of the actual inliers or having to use a RANSAC implementation that took too much time to estimate $I, O$ to be practical. In order to still be able to train our method from monocular image sequences, we propose an alternative method to obtaining valid $I$ and $O$, based on Lucas-Kanade tracking [45] (KLT). In this method, both $P_0$ and $P_1$ are tracked across the images between $I_0$ and $I_1$ to obtain $P_1^* = \{T(p_{0,j})\}$ and vice versa, where $T(p_{0,j})$ is $p_{0,j} \in P_0$ tracked from $I_0$ to $I_1$. 
using KLT. \( M \) then undergoes the following assignment to \( I \) and \( O \):

\[
m = (p_{0,j}, p_{1,j}) \begin{cases} 
\in I & \text{if } \exists T(p_{i,j}) \text{ and } \|T(p_{i,j}) - p_{i',j}\| < 3 \text{ pixels} \\
\in I \cup O & \forall (i, i') \in \{(0, 1), (1, 0)\} \\
\not\in I, O & \text{if either track lost at image border} \\
\in O & \text{otherwise.}
\end{cases}
\]

(11)

Besides enabling training on monocular image sequences, this approach has the advantages that it also works with uncalibrated cameras and that it does not wrongfully punish unmatched interest points which have left the field of view between images. Conversely, it has the disadvantage that it is now not trained with P3P RANSAC in the loop any more. Note however that non-maximum suppression, interest point description and matching are still part of the training.

### 5.4 Selecting good training image pairs without pose ground truth

In this section, we describe how we select the image pairs \((I_0, I_1)\) for training without relying on pose ground truth. This is also achieved with KLT. For each image sequence used as training dataset, we track a large amount of corners \(n_{\text{sel}}\) across the full sequence and use the FAST detector to constantly detect new corners, such that \(|P_{\text{sel}}|\) has a constant value. Each point in \(P_{\text{sel}}\) is either newly detected or tracked from a previous frame.

To generate training image pairs, \(I_0\) is first randomly selected from the full dataset. \(I_1\) is then randomly selected from the images that come after \(I_0\) and which have at least \(o \cdot |P_{\text{sel},0}|\) points in \(P_{\text{sel},1}\) that have been tracked from \(P_{\text{sel},0}\). If the minimum overlap \(o\) is chosen too big, the images in the pairs will be too similar for the network to train viewpoint invariance, whereas if \(o\) is too small, this could reduce the chance of successful relative pose estimations and as a consequence wrongfully punish good interest points in the reinforcement loss.

We have empirically found a good value for \(o\) to be 0.5.

### 6 Experiments

We train and evaluate our methodology on the KITTI [46], Robotcar [47], Euroc [48] and TUM mono [4] datasets. On these, we compare SIFT [25], ORB [49], Predicting Matchability [42](PredMatch) and LIFT [15], each treated as a black box which provides interest points, their scores (responses), and their descriptors. For [42,15] we use the publicly available code, for SIFT and ORB we use the OpenCV 3.4 implementations. Furthermore, we compare Harris [23], FAST [31] and DDET [16], where we take our pipeline as depicted in Fig. 2 and replace the Neural Network with the respective scoring function. Finally, we also compare several instances of our trained network, where the instances differ based on the datasets they were trained with. Networks are designated with \(kt\), \(rc\) and \(tm\) respectively for networks trained with the KITTI, Robotcar and TUM mono datasets. Finally, there is \(all\), which stands for a network trained on all training sequences.
6.1 Dataset description

Both the KITTI and the Robotcar datasets are outdoor datasets captured by a car, while the Euroc dataset is captured indoors, by a drone. The TUM mono dataset contains both indoor and outdoor datasets and has been captured with a hand-held camera. Pose ground truth is only available for KITTI, Robotcar and Euroc, so we only use these datasets for evaluation. The TUM mono dataset only contains monocular sequences, so it has only been trained using the KLT method. For the other datasets, both training methods are evaluated. In all datasets, training, validation and testing have been separated strictly and geographically. See Section 9.1 in the supplementary material for details on this separation.

For evaluation in both the KITTI and Robotcar datasets, image pairs are randomly sampled such that the two images are captured within a relative distance $\delta_t$ of 5 m and a heading difference $\delta_R$ of less than 30 degrees. For evaluation in Euroc, the distance $\delta_t$ is reduced to 0.5 m due to it being captured in a smaller space. Image pairs are selected pseudo-randomly, but with a fixed seed, such that all methods are evaluated with the same image pairs. Several example image pairs are shown in the supplementary material in Section 9.4.

6.2 Network Architecture

We use a fully convolutional network [50] with $d + 1$ layers. The first $\frac{d}{2}$ layers are unpadded $3 \times 3$ 2D convolution layers with an output of 64 channels each. The second $\frac{d}{2}$ layers are correspondingly $3 \times 3$ deconvolution layers with an output of 128 channels each. Each of the first $d$ layers is followed by a ReLU activation. Finally, the last layer is a padded $3 \times 3$ convolution that outputs a single channel, which is the network output. It is followed by a sigmoid activation to force $S$ into the range $[0, 1]$. We choose the network depth $d$ as the only hyperparameter of our network and report the trade-off between computation time and performance that results from varying it (see Fig. 6). Based on this, we set the default depth of the network to $d = 16$.

7 Results

Fig. 4 shows the trade-off between succinctness and pose estimate accuracy as we vary $k$, the required inlier count. Succinctness is summarized with AUC-20, while both rotation and position errors are summarized with AUC-1. We do not show results of ORB, Harris and FAST because all of them perform equally to or worse than SIFT. From our networks, we only show the results for the network trained on all training datasets with the P3P method (only image pairs sampled from TUM mono are treated with the KLT method). This network has the best overall performance. See Section 9.2 in the supplementary material for a detailed analysis of the different networks.

As one would expect, increasing $k$ results in better pose accuracy but worse succinctness: More required inliers allow a more accurate pose estimation, but
also require more interest points to be extracted in the first place. For pose accuracy, however, a plateau is reached almost universally at \( k = 10 \). An exception is the rotation accuracy in Euroc, which continues to improve as \( k \) increases beyond 10. This could be due to the fact that the images in Euroc have a lower resolution than the images in the other datasets. We can see that our best network outperforms the best baseline, LIFT, on all datasets. Surprising results are the poor succinctness of PredMatch, the excellent relative pose quality of DDET on Euroc and the poor relative pose quality of SIFT on Euroc. We believe that the poor succinctness of PredMatch is simply due to the fact that it has been originally trained and evaluated with much higher interest point counts than 200, which is our upper bound. With the absence of a mechanism that would ensure consistent ranking of scores, good performance at high interest point counts would not necessarily transfer to good performance at low interest point counts (Fig. 3). What redeems PredMatch is its pose estimation accuracy which is typically higher than the one of the SIFT detector (its original baseline), especially in the Euroc dataset. The performance gap between SIFT and DDET on Euroc can be explained with the high amount of repetitive features, such as checker-
boards, heaters and patterns on the floor, that are present in the dataset. We could verify that SIFT would detect a large amount of such repetitive features, while DDET seems to be exceptionally good at avoiding them.

Succinctness curves for $k = 10$ are shown in Fig. 5. As we can see, our network can achieve essentially the highest success rate for getting 10 inliers extracting only 100 interest points, while only marginally worse success rates can already be achieved on KITTI and Euroc with only 50 extracted interest points. These values of course depend on the maximum distances $\delta_t$ and $\delta_R$ which we impose on the image pairs selected for evaluation. Stronger viewpoint changes would likely require more points to be extracted.

Fig. 6 shows the trade-off between execution time and succinctness as the depth of the network is increased, evaluated on the validation data of the KITTI dataset. Here, the forward pass time only of the neural network, that is, obtaining $S_i$ from $I_i$, is measured. Surprisingly good results can be achieved with a depth as low as 3 ($d = 2$), succinctness comparable to LIFT is achieved at $d = 8$ and a plateau is reached at $d = 16$, which we thus choose as default depth for our network. Unsurprisingly, the time of a forward pass grows linearly with the network depth. Compared to detection times of traditional feature detectors, the forward pass time of our network at a depth of say $d = 16$ may seem to be high for practical purposes, but we believe that, with recent developments...
in hardware-accelerated inference engines \cite{51,52,53}, this will become virtually a non-issue in the near future. Compare this to LIFT, where the same forward pass (sum over multiple scales) takes 650 milliseconds in total.

8 Conclusion

In this paper, we have presented a new, unsupervised training method for an interest point detector. Rather than just aiming at repeatability at high interest point counts, we explicitly target extracting as few interest points as necessary while achieving good relative pose estimates. We quantify this using a novel metric we call succinctness and simultaneously quantify the quality of the relative pose estimate. With respect to these metrics, we show that our approach outperforms a variety of previous methods on datasets which represent realistic pose estimation scenarios for autonomous cars and drones.

Succinctness is achieved by using a mixture of inlier reinforcement and enforcing consistent ranking. The training method uses structure-from-motion in the loop, including non-maximum-suppression, interest point description and matching, and inlier and outlier distinction using either P3P RANSAC or Lucas-Kanade tracking. Due to that, the network can adapt to the particularities and needs of these components. Since the loss only depends on the inliers and outliers estimated in the forward pass, no ground truth information is required from training image sequences. Finally, the full code of the method will be made publicly available.

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. We would like to thank T. Sattler for very fruitful discussions and feedback and A. Loquercio for very helpful code reviews.
References

1. Mur-Artal, R., Montiel, J.M.M., Tardós, J.D.: ORB-SLAM: a versatile and accurate monocular SLAM system. IEEE Trans. Robot. 31(5) (2015) 1147–1163
2. Forster, C., Zhang, Z., Gassner, M., Werlberger, M., Scaramuzza, D.: SVO: Semidirect visual odometry for monocular and multicamera systems. IEEE Trans. Robot. 33(2) (2017) 249–265
3. Engel, J., Schöps, J., Cremers, D.: LSD-SLAM: Large-scale direct monocular SLAM. In: Eur. Conf. Comput. Vis. (ECCV). (2014)
4. Engel, J., Koltun, V., Cremers, D.: Direct Sparse Odometry. IEEE Trans. Pattern Anal. Machine Intell. PP(99) (2017) 1–1
5. Bloesch, M., Omari, S., Hutter, M., Siegwart, R.: Robust visual inertial odometry using a direct EKF-based approach. In: IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS). (2015)
6. Leutenegger, S., Lynen, S., Bosse, M., Siegwart, R., Furgale, P.: Keyframe-based visual-inertial SLAM using nonlinear optimization. Int. J. Robot. Research (2015)
7. Qin, T., Li, P., Shen, S.: VINS-Mono: A robust and versatile monocular visual-inertial state estimator. arXiv e-prints (August 2017)
8. Forster, C., Carlone, L., Dellaert, F., Scaramuzza, D.: On-manifold preintegration for real-time visual-inertial odometry. IEEE Trans. Robot. 33(1) (2017) 1–21
9. Delmerico, J., Scaramuzza, D.: A benchmark comparison of monocular visual-inertial odometry algorithms for flying robots. IEEE Int. Conf. Robot. Autom. (ICRA) (May 2018)
10. Longuet-Higgins, H.C.: A computer algorithm for reconstructing a scene from two projections. Nature (293) (1981) 133–135
11. Nistér, D.: An efficient solution to the five-point relative pose problem. IEEE Trans. Pattern Anal. Machine Intell. 26(6) (2004) 756–777
12. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM 24(6) (1981) 381–395
13. Schmid, C., Mohr, R., Bauckhage, C.: Evaluation of interest point detectors. Int. J. Comput. Vis. 37(2) (June 2000) 151–172
14. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. IEEE Trans. Pattern Anal. Machine Intell. 27(10) (2005) 1615–1630
15. Yi, K.M., Trulls, E., Lepetit, V., Fua, P.: LIFT: Learned invariant feature transform. In: Eur. Conf. Comput. Vis. (ECCV). (2016) 467–483
16. Lenc, K., Vedaldi, A.: Learning covariant feature detectors. In: Eur. Conf. Comput. Vis. Workshops (ECCVW). (2016) 100–117
17. Savinov, N., Seki, A., Ladicky, L., Sattler, T., Pollefeys, M.: Quad-networks: Unsupervised learning to rank for interest point detection. In: IEEE Int. Conf. Comput. Vis. Pattern Recog. (CVPR). (2017)
18. DeTone, D., Malisiewicz, T., Rabinovich, A.: SuperPoint: Self-supervised interest point detection and description. arXiv e-prints (December 2017)
19. Forster, C., Lynen, S., Kneip, L., Scaramuzza, D.: Collaborative monocular SLAM with multiple micro aerial vehicles. In: IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS). (2013) 3962–3970
20. Cieslewski, T., Scaramuzza, D.: Efficient decentralized visual place recognition using a distributed inverted index. IEEE Robot. Autom. Lett. 2(2) (April 2017) 640–647
21. Cieslewski, T., Choudhary, S., Scaramuzza, D.: Data-efficient decentralized visual SLAM. IEEE Int. Conf. Robot. Autom. (ICRA) (May 2018)
22. Moravec, H.P.: Obstacle Avoidance and Navigation in the Real World by Seeing Robot Rover. PhD thesis, Carnegie-Mellon University, Pittsburgh, Pennsylvania (September 1980)
23. Harris, C., Stephens, M.: A combined corner and edge detector. In: Proc. Fourth Alvey Vision Conf. Volume 15., Manchester, UK (1988) 147–151
24. Shi, J., Tomasi, C.: Good features to track. In: IEEE Int. Conf. Comput. Vis. Pattern Recog. (CVPR). (June 1994) 593–600
25. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. 60(2) (November 2004) 91–110
26. Mainali, P., Lafruit, G., Yang, Q., Geelen, B., Gool, L.V., Lauwereins, R.: SIFER: scale-invariant feature detector with error resilience. Int. J. Comput. Vis. 104(2) (2013) 172–197
27. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: SURF: Speeded up robust features. Comput. Vis. Image. Und. 110(3) (2008) 346–359
28. Guiducci, A.: Corner characterization by differential geometry techniques. Pattern Recognition Letters 8(5) (1988) 311–318
29. Smith, S.M., Brady, J.M.: SUSAN – a new approach to low level image processing. Int. J. Comput. Vis. 23(1) (1997) 45–78
30. Trajković, M., Hedley, M.: Fast corner detection. Image and vision computing 16(2) (1998) 75–87
31. Rosten, E., Drummond, T.: Machine learning for high-speed corner detection. In: Eur. Conf. Comput. Vis. (ECCV). (2006) 430–443
32. Mikolajczyk, K., Schmid, C.: Scale & affine invariant interest point detectors. Int. J. Comput. Vis. 60 (2004) 63–86
33. Dias, P., Kassim, A.A., Srinivasan, V.: A neural network based corner detection method. In: IEEE Int. Conf. Neural Netw. Volume 4. (1995) 2116–2120
34. Richardson, A., Olson, E.: Learning convolutional filters for interest point detection. In: IEEE Int. Conf. Robot. Autom. (ICRA). (May 2013) 631–637
35. Trujillo, L., Olague, G.: Using evolution to learn how to perform interest point detection. In: IEEE Int. Conf. Pattern Recog. (ICPR). (2006) 211–214
36. Verdie, Y., Yi, K.M., Fua, P., Lepetit, V.: TILDE: A temporally invariant learned detector. In: IEEE Int. Conf. Comput. Vis. Pattern Recog. (CVPR). (2015) 5279–5288
37. Turcot, P., Lowe, D.G.: Better matching with fewer features: The selection of useful features in large database recognition problems. In: Int. Conf. Comput. Vis. Workshops (ICCVW). (2009) 2109–2116
38. Li, Y., Snavely, N., Huttenlocher, D.P.: Location recognition using prioritized feature matching. In: Eur. Conf. Comput. Vis. (ECCV). (2010) 791–804
39. Park, H.S., Wang, Y., Nurvitadhi, E., Hoe, J.C., Sheikh, Y., Chen, M.: 3d point cloud reduction using mixed-integer quadratic programming. In: IEEE Int. Conf. Comput. Vis. Pattern Recog. Workshops (CVPRW). (2013) 229–236
40. Cao, S., Snavely, N.: Minimal scene descriptions from structure from motion models. In: IEEE Int. Conf. Comput. Vis. Pattern Recog. (CVPR). (June 2014) 461–468
41. Dymczyk, M., Lynen, S., Cieslewski, T., Bosse, M., Siegwart, R., Furgale, P.: The gist of maps - summarizing experience for lifelong localization. In: IEEE Int. Conf. Robot. Autom. (ICRA). (2015)
42. Hartmann, W., Havlena, M., Schindler, K.: Predicting matchability. In: IEEE Int. Conf. Comput. Vis. Pattern Recog. (CVPR). (2014) 9–16
43. Dymczyk, M., Stumm, E., Nieto, J., Siegwart, R., Gilitschenski, I.: Will it last? learning stable features for long-term visual localization. In: 3D Vision (3DV). (2016) 572–581
44. Kingma, D.P., Ba, J.L.: Adam: A method for stochastic optimization. (2015)
45. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Int. Joint Conf. Artificial Intell. (IJCAI). (1981) 674–679
46. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the KITTI vision benchmark suite. In: IEEE Int. Conf. Comput. Vis. Pattern Recog. (CVPR). (2012)
47. Maddern, W., Pascoe, G., Linegar, C., Newman, P.: 1 year, 1000 km: The Oxford RobotCar dataset. Int. J. Robot. Research 36(1) (2017) 3–15
48. Burri, M., Nikolic, J., Gohl, P., Schneider, T., Rehder, J., Omari, S., Achtelik, M.W., Siegwart, R.: The EuRoC micro aerial vehicle datasets. Int. J. Robot. Research 35 (2015) 1157–1163
49. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: ORB: An efficient alternative to SIFT or SURF. In: Int. Conf. Comput. Vis. (ICCV). (2011)
50. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: IEEE Int. Conf. Comput. Vis. Pattern Recog. (CVPR). (June 2015) 3431–3440
51. Han, S., Liu, X., Mao, H., Pu, J., Pedram, A., Horowitz, M.A., Dally, W.J.: EIE: efficient inference engine on compressed deep neural network. In: ACM/IEEE Int. Symp. Comp. Arch. (ISCA). (2016) 243–254
52. Andri, R., Cavigelli, L., Rossi, D., Benini, L.: Yodann: An ultra-low power convolutional neural network accelerator based on binary weights. In: IEEE Comp. Soc. Symp. VLSI (ISVLSI). (2016) 236–241
53. Suleiman, A., Chen, Y.H., Emer, J., Sze, V.: Towards closing the energy gap between HOG and CNN features for embedded vision. In: IEEE Int. Symp. Circuits Syst. (ISCAS). (2017) 1–4
9 Supplementary material

9.1 Detailed Dataset Description

KITTI In the KITTI dataset, we use sequences 06, 08, 09 and 10 for training, sequence 05 for validation and sequence 00 for testing.

Robotcar Robotcar is an extensive dataset, with many sequences recorded over the same trajectory across different seasons and times of day. In this paper, however, we do not tackle the aspect of these appearance changes, but use the dataset because it provides ground truth poses for every camera frame, which we need for evaluation. In the Robotcar dataset, we are faced with the problem that all datasets are taken either on the main route or on a very small alternative route. Thus, in order to provide a rich training set while still allowing testing on the main route, we use a geographical subdivision of the main route for the training, validation and testing split. See Fig. 7 for a visualization of the splitting. Additionally to the training data on the main route, we use the alternative route as a source of training data. From the main route, we use sequence 2014-07-14-14-49-50 and from the alternative route the sequences 2014-05-19-12-51-39 and 2014-06-26-08-53-56. Finally, we crop the bottom 160 pixels of the images because they mainly contain the hood of the car.

Euroc Euroc is used for testing only, and we use sequence V1_01.

TUM mono We use the indoor sequences 01, 02, 03 and the outdoor sequences 48, 49 and 50, for training with the KLT method only.

9.2 Performance of different networks

We can see in Fig. 8 that while networks trained on similar data perform better than networks trained on completely different data, the network trained on
Fig. 8. AUC-200 of the required interest point count $n$ (solid), AUC-1° of the rotation error $e_R$ (dashed) and AUC-1m of the translation error $e_t$ (dotted), for different values on $k$, using $l = 100$ samples on the testing data. Generally, higher values are better. See definitions in equations (1), (3).

All training datasets performs best overall, usually performing similarly to the networks trained on similar data only. For the Euroc dataset, where no similar data was available, several networks perform similarly, but more importantly, the network trained with all data has a slight advantage over the other networks. Another thing that is clear from the results is that training with the P3P method is preferable to training with the KLT method.

9.3 Error curves for $k = 10$

See Fig. 9

9.4 Sample testing image pairs

See Figs. 10-12
Fig. 9. Error curves, as defined in equation (2), for \( k = 10 \). Note the different x range in (c) and (f) due to the different scale of the Euroc dataset.
Fig. 10. Sample image pairs from the KITTI dataset and corresponding scores and interest points. Circles indicate extracted interest points, outliers are red, inliers are green, green lines indicate correct matches. Metrics according to [1] are reported in the caption.
Fig. 11. Sample image pairs from the Robotcar dataset and corresponding scores and interest points. Circles indicate extracted interest points, outliers are red, inliers are green, green lines indicate correct matches. Metrics according to (1) are reported in the caption.
Fig. 12. Sample image pairs from the Euroc dataset and corresponding scores and interest points. Circles indicate extracted interest points, outliers are red, inliers are green, green lines indicate correct matches. Metrics according to [1] are reported in the caption.