UNIVERSITÄT BONN

Project Report
for Internship at
Rheinische Friedrich-Wilhelms-Universität Bonn
Institute of Geodesy and Geoinformation

On Evaluation of Interest Point Detectors and Descriptors

by

Abhinav Tripathi

from
Lucknow, India

Bonn 2019

Supervisor:
Prof. Dr. Cyrill Stachniss, University of Bonn, Germany

Second Supervisor:
Andres Milioto, University of Bonn, Germany

Abstract

Feature detectors and descriptors are widely used in a number of computer vision applications. There have been many comparisions of descriptors, but comprehensive comparisions of detectors are limited. In this work, we compare the performance of interest point detectors and descriptors. We evaluate the detectors based on their repeatability and coverage. We also introduce matching ratios as a metric for measuring the performance of detectors, providing a comprehensive study. Descriptors are compared based on their ability to correctly estimate homography. The distintiveness and accuracy of descriptors is qualitatively assessed using the reprojection error of the nearest neighbor matches.

Our approach involves using HPatches Sequences dataset for comparing several features including both - hand-crafted and learnt features. We show that higher repeatability of keypoints does not directly improve the matching performance of features. We also show that the traditional features are still very competitive with the deep learning based features for viewpoint changes, however learnt features outperform traditional features in the case of illumination changes.

Contents

Chapter 1

Introduction

An interest point or a keypoint detector is an algotrithm to detect certain salient points like corners or blobs in the images. A descriptor is a vector that somehow describes the feature points mathematically. A detector and descriptor together form a local feature. The feature detection and description algorithms are expected to return the same keypoints and feature vectors even if the scene in the image undergoes transformations like changes in viewpoints and illumination. There are a lot of applications of interest point detectors and descriptors - object recognition, robot navigation, feature tracking, image alignment, structure from motion and image retrieval to name a few. In this work, we compare the performance of several interest point detectors and descriptors.

1.1 Motivation

Many feature evaluation pipelines focused on various aspects of features have been proposed in the past. Most of the detector evaluations are based on the methods described by Mikolajczyk *et al.*in [8]. These methods are designed for region detectors and cannot be directly extended to detectors which only give keypoints which do not provide any scale information. Similarly, for evaluating descriptors, most of the approaches ([1], [7]) take into consideration a fixed size patch to compute and evaluate descriptors. The reason is that many of the feature detection algorithms can take an image patch as input and output a feature vector. However, this is not true for some recently developed deep learning based feature descriptors like [2]. These algorithms operate on complete images rather than just a patch to compute the keypoints and feature vectors. Hence the same evaluation pipeline cannot be used for comparing them with other hand-crafted and learnt features. This work aims to bridge this gap and provide an evaluation pipeline which is universal and can be applied to any detector and descriptor.

Figure 1.1: Image shows 40 best matches using SuperPoint features. Experiments show that the SuperPoint features work extremely well in case of sequences with illumination changes.

1.2 Contribution

The main contribution of this work is to provide a universally applicable pipeline for comprehensive comparision of various feature detection and description algorithms. We do so by providing an evaluation protocol (Chapter 4 and Chapter 5) on a recent feature benchmark [1] for image sequences representative of various difficult imaging scenarios, such as illumination and viewpoint changes.

Based on our experiments, we make following claims: (i) higher repeatability of keypoints does not directly improve the matching performance of features, (ii) traditional features are still very competitive with the deep learning based features in performance on sequences with viewpoint changes, (iii) however learnt features outperform traditional features in the case of illumination changes.

These claims are backed up by the paper and our experimental evaluation.

1.3 Report Overview

In Chapter 2, we discuss the features we selected for comparision. Chapter 3 describes the dataset on which the evaluations were made. Chapter 4 and Chapter 5 describe the evaluation metrics and their results, which is followed by Chapter 6 the conclusion.

Chapter 2

Detectors and Descriptors

We compare a few classical interest point detectors and descriptors with the latest deep learning based ones. A brief discussion of different features compared is provided below. Table 2.1 shows the average number of keypoints detected by each detector over the dataset HPatches. For this work, we compare a set of 6 detectors and 5 descriptors enumerated below.

- Detectors: ORB, SIFT, SFOP, LIFT, SuperPoint, D2Net
- Descriptors: ORB, SIFT, LIFT, SuperPoint, D2Net

Table 2.1: Average feature computation time and number of keypoints detected

Feature	Run time [s]	CPU/GPU	Number of Keypoints
ORB	0.07	CPU	498.99
SIFT	0.41	CPU	4664.98
SFOP (no descriptor)	7.01	CPU	1901.42
LIFT	39.49	CPU	933.06
SuperPoint	0.89	GPU	1758.92
D2Net	0.51	GPU	6288.80

2.1 ORB

ORB [9] stands for *Oriented FAST (oFAST) and Rotation Aware BRIEF (rBRIEF).* It was proposed as a real time alternative to SIFT and SURF with similar performance. The keypoints are detected using FAST points, where intensities of a set of contiguous pixels on a circle are tested against the intensity of centre pixel. ORB uses FAST-9 (arc of 9 contiguous pixels) with a circle of radius 3. If the difference in intensities of the centre pixel and pixels on circular ring are greater

than a threshold, then the centre is considered as a candidate for keypoint. To obtain N keypoints, the threshold is set low enough to get more than N keypoints. Then the keypoints are ordered according to the Harris Corner Measure and the first N points are selected. This step removes the keypoints along the edges and favours the ones at the corners. Since FAST operates on a single scale, a multiscale pyramid of five scales is used with a scaling factor of $\sqrt{2}$ with Harris filtering at each scale. The orientation is decided using a vector pointing to the intensity centroid in the neighbourhood of the centre pixel.

BRIEF is a binary descriptor constructed by comparing the intensity values (of 256 pairs by default) of points around any given keypoint. These pairs of points are steered along the orientation angle from oFAST. This makes the pairs of points highly correlated which is not good for the descriptor. In a final step, a greedy algorithm is used to learn the pairs with reduced correlation. This is known as rBRIEF.

The property of ORB which makes it stand out among other classical features is that it runs in real time.

2.2 SIFT

SIFT or *Scale Invariant Feature Transform* [6] has four main stages to generate a set of features from an image. The first two steps involve extracting keypoints across various scales. This is followed by orientation assignment and feature description. In the first step, a scale space is created and the image is blurred progressively in different scales. A difference of these blurred images is calculated as an approximation of Laplacian of Gaussian operated on the original image. In the second step, the putative keypoints are calculated as the local extremas in the images. The keypoints with low contrast and along the edges are removed. In the third step, an orientation of the detected feature is computed by binning the orientation of pixels around the keypoint in a histogram of 36 bins. The peak decides the orientation of the keypoint. If there is another peak with 80% of the maximum, it is considered as a different keypoint with same position but different orientation.

Finally, a feature vector is created by taking a region of 16×16 sq. pixels around the keypoint. This region is broken into 16 windows of 4×4 sq. pixels. Each of these 16 windows can hold one of 8 quantized orientation values. This results into a 128 dimensional feature vector.

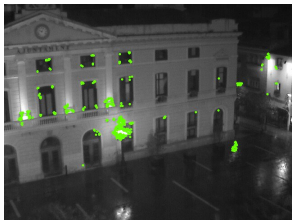
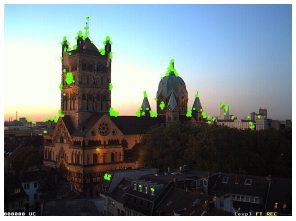
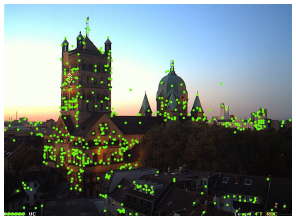
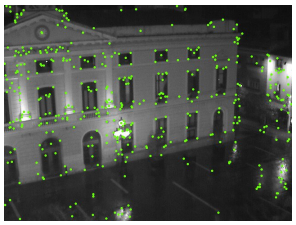
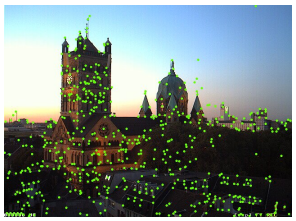
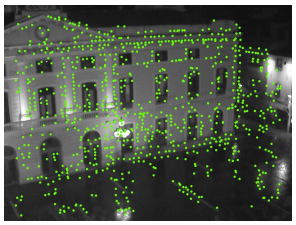
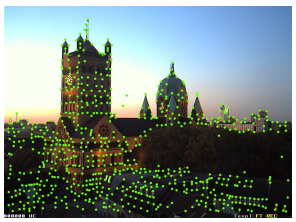
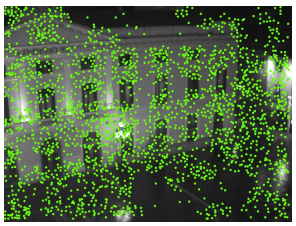
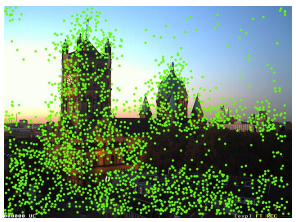
Detector	Ajuntament	Castle	Boat
ORB			
SIFT			
SFOP			
LIFT			
SuperPoint			
D2Net			

Table 2.2: Keypoints are shown in green dots. The figure shows the output of different keypoint detectors on same image. Notice that ORB keypoints are mostly clustered in some regions. SuperPoint keypoints are the most intuitive to understand as they occur at corners of geometric shapes. Most of the LIFT keypoints occur along the edges of the structures.

2.3 SFOP

SFOP detector [5] is based on the principle of completeness and complementarity. Whereas certain detectors are based on extracting either junctions or blobs, SFOP focuses on both. However, SFOP is only a detector. Unlike the rest of the algorithms discussed, it does not have a paired descriptor. The underlyng mathematics of SFOP is complex and the reader is referred to [5] for a detailed understanding.

2.4 LIFT

LIFT or *Learned Invariant Feature Transform* [10] also has three main stages during test time. The network first detects keypoints, which is followed by orientation estimation and descriptor vector computation.

LIFT has a siamese training architecture with four heads. Each branch takes as input a patch which passes through the detector, orientation estimator and the descriptor blocks. The first two patches have different views of the same physical point, which are used as positive examples to train the descriptor, the third and fourth heads have a negative example as they contain patches of a different physical point and no feature point respectively. LIFT is trained on Piccadilly and Roman-Forum sequences of Photo-Tourism dataset. Patches are extracted using the keypoints from these images which survive a SIFT based VisualSFM reconstruction pipeline.

2.5 SuperPoint

SuperPoint [2] is fully convolutional network which operates on full size images and outputs interest points and descriptors in a single forward pass. The model architecture consists of a shared encoder that splits into two decoder heads: one for detection and other for description. The decoder branches are trained with specific losses for the purpose of detection and description.

The training of the network involves three major steps. First, the interest point detector is trained on a dataset of synthetic images of simple geometric shapes with no ambiguity in the location of interest points. Second, a procedure of *homographic adaptation* is used to extract keypoints by viewing the input image from different viewpoints and scales. An aggregation of interest points detected over the samples helps boost the performance of the detector. Finally, the generated labels from previous step are used to jointly train detector and descriptor networks.

2.6 D2Net

D2Net [3] is similar to SuperPoint during test time. It is also a fully convolutional network that operates on full size images and outputs interest points and descriptors. However, in this method all the parameters for detection and description are shared. The network is simultaneously optimized for both the tasks during training. Since both detector and descriptor share the same feature maps representation, the network is referred to as D2Net.

Chapter 3

Dataset

For evaluation of interest point detectors and descriptors we use the dataset *HPatches Sequences*. HPatches dataset of full image sequences was used to evaluate keypoint detectors and descriptors. It is a set of 116 sequences with 59 sequences with viewpoint changes and 57 sequences with illumination changes. Each sequence is a set of six images with a reference image and five target images. Homography matrix between reference image and each target image is provided as ground truth. Hence, each sequence has six images and five homography matrices. The dataset is an aggregation of a number of new sequences with some sequences taken from older datasets. The experiments show that number of keypoints detected does not play a major role in the performance of features. Table 3.1 summarizes the basic statistics of the dataset. Figure 3.1 shows the distribution of image sizes in the dataset. Figure 3.2 and Figure 3.3 show example sequences from the dataset.

Table 3.1: HPatches Sequences Dataset

Type	#Sequences	#Imgs	#ImgPairs
Viewpoint	57	342	285
Illumination	59	354	295
Total	**116**	**696**	**580**

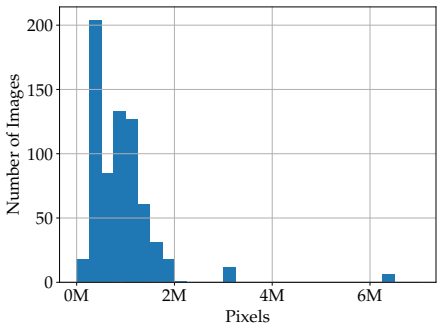
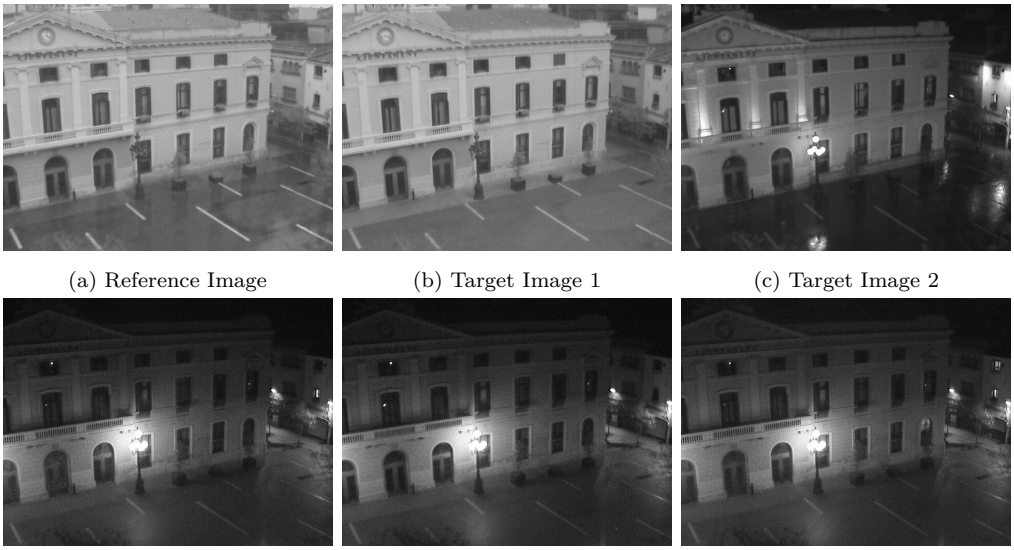


Figure 3.1: Distribution of image size



(a) Reference Image (b) Target Image 1 (c) Target Image 2

(d) Target Image 3 (e) Target Image 4 (f) Target Image 5

Figure 3.2: The figure shows the six images from the sequence `i_ajuntament` with a reference image and five target images with **illumination changes**.

(a) Reference Image (b) Target Image 1 (c) Target Image 2

(d) Target Image 3 (e) Target Image 4 (f) Target Image 5

Figure 3.3: The figure shows the six images from the sequence `v_graffiti` with a reference image and five target images with **viewpoint changes**. The homography matrix between reference image and each target image is given.

Chapter 4

Detector Evaluation

The main focus of this chapter is to provide a framework for evaluating keypoint detectors. The keypoint detectors are evaluated over the following metrics: *coverage, repeatability, unique matching ratio, multiple matching ratio* and *spurious keypoint ratio*. The details of the metrics and the evaluation results are discussed in the sections below.

4.1 Coverage

Coverage represents the spatial distribution of the detected keypoints. It is important to have keypoints well distributed over the images. Many computer vision applications like tracking require the keypoints to be evenly spread across the image for accurate results. Sparsely clustered keypoints are not well suited for the task of visual odometry. For evaluating how the keypoints are distributed over an image, a strategy inspired from [4] is used. The requirement is to have the keypoints spread as evenly as possible. Let the euclidean distance between two keypoints $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$ in the same image be $d_{ij} = ||\boldsymbol{x_i} - \boldsymbol{x_j}||$. First an average distance of all other keypoints is calculated w.r.t. a given keypoint as a harmonic mean.

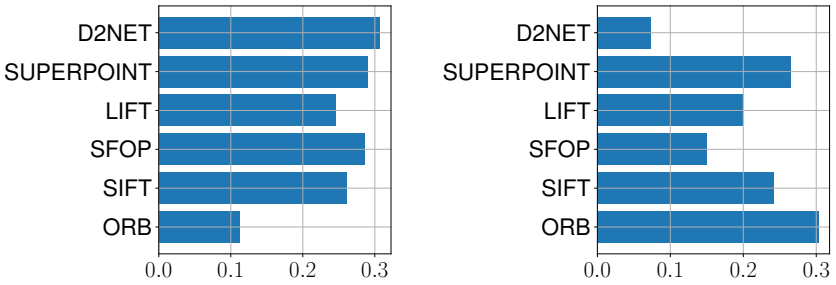
$$D_i = \frac{N-1}{\Sigma_{i \neq j}(1/d_{ij})} \tag{4.1}$$

Since the choice of reference point can affect the distances, the value is calculated with each keypoint as reference and finally averaged.

$$C_0 = \frac{N}{\Sigma_{i=1}^{N}(1/D_i)} \tag{4.2}$$

Harmonic mean is used in order to penalize small distances. Finally, we define coverage as:

$$C = \frac{C_0}{\sqrt{H \times W}} \tag{4.3}$$



(a) Coverage (Higher is better)

(b) Repeatability (Higher is better)

Figure 4.1: ORB has significantly less coverage as the keypoints are clustered around the region which has high texture and abrubt intetnsity changes, however it has high repeatability. SuperPoint and SIFT have both high coverage and repeatability.

In multi-scale systems, keypoints of different scale may be very close. Hence, the distances d_{ij} less than 1 px are removed from computation of coverage.

Coverage results: Experiments show that ORB has lowest coverage among all interest point detectors. This can be explained by the fact that ORB has the least number of keypoints, further the ORB keypoints are chosen using the Harris corner measure and only the points with rich texture are picked. The results are shown in Figure 4.1a.

4.2 Repeatability

Repeatability is calculated for a pair of images. It measures the ability of a detector to identify same features across different images despite changes in viewpoint and illumination. We consider only the region common in both the images for a fair comparision. The higher the repeatability of a detector, the higher is the possibility of finding a match.

The classical repeatability benchmark proposed in [8] considers the overlap area of the region detectors. Since we are also comparing deep learning based detectors which detect keypoints (and do not detect regions), the same strategy cannot be used. We follow the repeatability rate described in [2] as it can be applied universally to all the feature detectors.

Let the number of keypoints in the common region of the pair of images be N_1 and N_2. For every keypoint in the first image, the presence or absence of keypoint in the second image is checked. The repeatability rate is symmetrically calculated as:

$$\texttt{Rep} = \frac{1}{N_1 + N_2} \left(\texttt{Corr}_1 + \texttt{Corr}_2\right) \tag{4.4}$$

where $\texttt{Corr}_1$ and $\texttt{Corr}_2$ are defined below.

Let $\boldsymbol{x_i}$ be a keypoint detected in the first image and $\boldsymbol{x_j}$ be a keypoint detected in the second image. Using ground truth homography $\mathcal{H}$, we can project the keypoint $\boldsymbol{x_i}$ on to the second image to get $\hat{\boldsymbol{x}}_{\boldsymbol{i}}$. [1]

$$\hat{\boldsymbol{x}}_{\boldsymbol{i}} = \mathcal{H} \circ \boldsymbol{x_i} \tag{4.5}$$

`Corr` measures the number of keypoints that are repeated. This can be easily calculated by reprojecting the keypoints to the other image using ground truth homography.

$$\texttt{Corr}_1 = \sum_{i=1}^{N_1} \left(\min_{j \in 1,2,\ldots,N_2} ||\hat{\boldsymbol{x}}_{\boldsymbol{i}} - \boldsymbol{x_j}|| < \varepsilon \right) \tag{4.6}$$

$\hat{\boldsymbol{x}}_{\boldsymbol{i}}$ represents a keypoint detected in the first image reprojected on to the second image. It is considered repeated only if there exists a keypoint $\boldsymbol{x_j}$ in second image within ε pixels of $\hat{\boldsymbol{x}}_{\boldsymbol{i}}$. Similarly we have:

$$\texttt{Corr}_2 = \sum_{j=1}^{N_2} \left(\min_{i \in 1,2,\ldots,N_1} ||\hat{\boldsymbol{x}}_{\boldsymbol{j}} - \boldsymbol{x_i}|| < \varepsilon \right) \tag{4.7}$$

Repeatability results: The higher the repeatability rate, the higher is the number of points with the potential of good matching. Experimental results (see Figure 4.1b) show that ORB has the highest repeatability among all the detectors while D2Net has the least. Further experiments show that SuperPoint performs as good as ORB in case of illumination changes. We believe this is because of the training on the synthetic images with added noise.

4.3 Matching Ratio

For this metric, we try to match two points from different images without descriptors. Like repeatability, only the keypoints within the overlapping regions of the images are considered. We project the keypoints in a query image on to the target image using the given homography as ground truth and calculate reprojection error matrix $\mathbf{D}_{N_1 \times N_2}$. The entries of this matrix are

$$\mathbf{D} = [d_{ij}]_{N_1 \times N_2} = ||\hat{\boldsymbol{x}}_{\boldsymbol{i}} - \boldsymbol{x_j}|| \tag{4.8}$$

We consider a pair (i, j) as a match when the corresponding reprojection error is less than some threshold $d_{ij} < d_0$. Adjacency matrix can be calculated as

$$\mathbf{A} = [a_{ij}]_{N_1 \times N_2} = \begin{cases} 1, \, d_{ij} < d_0 \\ 0, \, \text{otherwise} \end{cases} \tag{4.9}$$

There are three possible cases:

[1] For clarity, we slightly abuse the notation by using $\mathcal{H}$ for both - (a) 3×3 homography matrix and (b) a function that acts on non-homogenous coordinates as in Equation (4.5).

- A matching (i, j) can be unique if $a_{ij} = \sum_i a_{ij} = \sum_j a_{ij} = 1$. Let the total number of unique matches be N_u, then define *unique match ratio* as

$$\rho_u = \frac{N_u}{\min(N_1, N_2)} \tag{4.10}$$

- A matching (i, j) can be non-unique if $a_{ij} = 1$ but either $\sum_i a_{ij} > 1$ or $\sum_j a_{ij} > 1$. Let the total number of multiple matches be N_m. Define *multiple match ratio* as

$$\rho_m = \frac{N_m}{(N_1 + N_2)} \tag{4.11}$$

- A point i or j has no match if the corresponding row or column is 0. These keypoints are known as spurious keypoints. Define *spurious keypoint ratio* as

$$\rho_s = \frac{\rho_{\texttt{ref}} + \rho_{\texttt{trg}}}{2} \tag{4.12}$$

  where $\rho_{\texttt{ref}}$ and $\rho_{\texttt{trg}}$ are the ratios of number of unmatched keypoints N_s over the number of keypoints in the shared region.

$$\rho_{\texttt{ref}} = \frac{N_{s_{\texttt{qry}}}}{N_1} \text{ and } \rho_{\texttt{trg}} = \frac{N_{s_{\texttt{trg}}}}{N_2} \tag{4.13}$$

These ratios incorporate two struggling criteria for detectors: we want the number of unique matches to be maximized while the overall number of detections to be minimized. Hence, a good detector must have high unique matching ratio and low multiple matching ratio and spurious keypoint ratio.

Matching ratios results: These ratios tell us about the matching quality of detected keypoints. We want the number of unique matches to be maximized while the overall number of detections to be minimized. A comparision is shown in Table 4.1. Results show that the quality of detections is the best in case of SuperPoint.

- *Unique Match Ratio*: Higher is better for this metric. This measures the ability of a keypoint detector to find the same interest point *uniquely* under changes in viewpoint and illumination. We see that SuperPoint has the highest number of unique matches.

- *Multiple Match Ratio*: Lower is better. It measures whether there are multiple matches (using adjacency matrix matching) within a distance of 1 pixel. It can go up in two cases: (a) multiple keypoints are detected very close to each other with high repeatability, and (b) same repeatable keypoint is selected across different scales. SuperPoint performs the best in this case.

Table 4.1: Matching Ratios

Detector	ρ_u	ρ_m	ρ_s
ORB	0.0794	0.2348	**0.6845**
SIFT	0.1635	0.1028	0.7336
SFOP	0.1852	0.0026	0.8351
LIFT	0.2122	0.0036	0.7912
SuperPoint	**0.3179**	**0.0001**	0.7136
D2Net	0.0885	0.0003	0.9198

- *Spurious Keypoint Ratio*: Lower is better. It tells us the percentage of detected keypoints that are not useful for matching. ORB keypoints have the lowest number of spurious keypoints.

Chapter 5

Descriptor Evaluation

The interest point descriptors are evaluated on the basis of their ability to estimate homography. This is of course biased by the detector of the feature descriptor. Hence, we can say that it is a measure of how good the feature is in general rather than just the descriptor.

5.1 Homography Estimation

A homography matrix is a square matrix of 9 elements. Different elements of the matrix represent different geometric transformations. Since it is not straightforward to compare two 3×3 homography matrices, we test the ability to transform the four corners of one image on to the other image. Let the four corners in the first image be c_1, c_2, c_3, c_4.

$$c'_j = \mathcal{H}_{\texttt{GT}} \circ c_j, \quad j \in \{1, 2, 3, 4\} \tag{5.1}$$

$$\hat{c'_j} = \mathcal{H}_{\texttt{EST}} \circ c_j, \quad j \in \{1, 2, 3, 4\} \tag{5.2}$$

Let c'_j represent ground truth homography applied to c_js and $\hat{c'_j}$ represent the estimated homography applied to c_js. Then, correctness of homography estimation is defined as:

$$\texttt{CorrHomography} = \frac{1}{N} \sum_{N} \left(\frac{1}{4} \sum_{j=1}^{4} ||c'_j - \hat{c'_j}|| < \varepsilon \right) \tag{5.3}$$

This metric represents the percentage of times homography can be correctly estimated. We say that the estimated homography is correct when the error from reprojecting the corners is below a certain threshold ε. For estimating homography, we use the OpenCV implementation of `findHomography` with RANSAC value set at 5.0 and match the nearest neighbor descriptors.

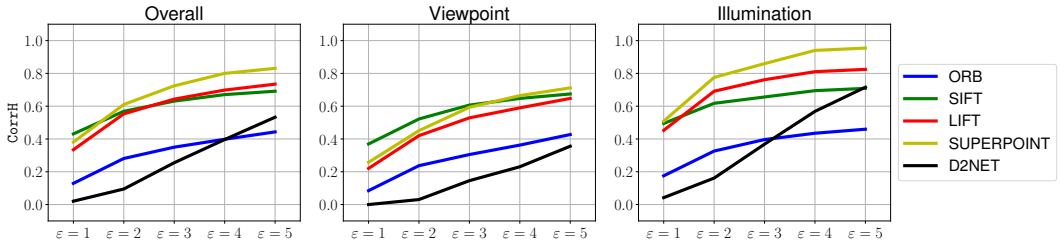


Figure 5.1: Homography Estimation: SuperPoint performs the best for sequences with illumination changes. SIFT descriptor has the highest accuracy for the sequences with changes in viewpoint as it performs the best at smaller value of ε. See Equation (5.3)

Homography estimation results: Figure 5.1 shows the number of times homography can be correctly estimated using nearest neighbors desctiptor matching strategy with the accuracy of ε pixels as mentioned in Equation (5.3). The higher the value of this metric, the better is the feature for the task of homography estimation. SIFT outperforms every other feature for an accuracy of 1 pixel. For more relaxed accuracy requirements, SuperPoint performs the best. For sequences with illumination changes, SuperPoint is the clear winner.

5.2 Reprojection Error

For each descriptor in the reference image, we search for the most similar descriptor in the target image. The nearest neighbors are considered as matches. For a good descriptor, it is necessary that the matches actually come from keypoints on the same object in the image. To check whether a match is correct or not, the ground truth homography can be used to calculate the reprojection error between the matching keypoints.

For a good descriptor, most of the matches must have a reprojection error close to zero.

Reprojection Errors of Nearest Neighbor: For a good feature, most of the nearest neighbor descriptor matches must have a low value of reprojection error. The qualitative results of the distribution can be seen in Figure 5.2. In an ideal case, where the ground truth homography perfectly describes the relation between two images and the nearest neighbor descriptor matches are all correct, the reprojection errors of matches will be zero. Thus, we get just one tall peak at zero for the feature that is perfect.

We see some strange distributions in case of ORB and SuperPoint with an initial peak in the case of former and several intermittent peaks in the case of latter. We are not quite sure what is the reason behind these unusual peaks.

We also see that the histogram for D2Net features is very wide. This means that the D2Net features performs worse than other features. This can be ex-

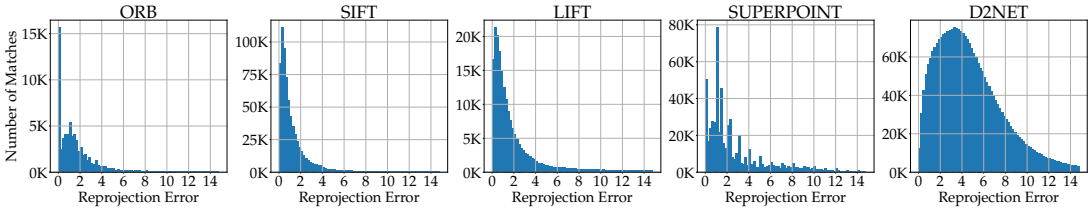


Figure 5.2: Histogram of number of nearest neighbor descriptor matches with reprojection error: Notice the unusual peaks in ORB and SuperPoint. SIFT, LIFT and D2Net show smoother distribution. However, the histogram for D2Net is very wide, which shows that D2Net is quite inaccurate. The histograms have been drawn over the whole dataset.

plained with the following reasoning: the network detects higher level features instead of hand-crafted corners or blobs. The low level local features like corners or blobs can be much more accurately localized than the higher level features. Hence, even though the descriptor may have a good description of the feature, the detector may not be accurate enough. Note that even though both D2Net and SuperPoint are both deep learning based, D2Net is trained to detect high level features, whereas SuperPoint is trained on synthetic dataset of geometric shapes to detect corners leading to better localization.

Chapter 6

Conclusion

Our approach involves evaluating several keypoint detectors and descriptors on the recent HPatches dataset. We see that the traditional features like SIFT are still very competitive (and in some cases even better) when compared with the modern deep learning based features. Based on the experiments, the following key insights can be drawn from evaluating detectors:

- The ORB keypoints do not cover the entire image, they are clustered in the regions where the measure of cornerness is high. SIFT, LIFT, SFOP, SuperPoint and D2Net have much better coverage when compared with ORB.

- ORB, SIFT and SuperPoint keypoints have high repeatability. These detectors can identify similar features over despite changes in viewpoints and illumination.

- SuperPoint detector has the highest number of unique matches and the lowest number of multiple matches, and thus the interest points are very distinctive. ORB detector has the least number of spurious keypoints, which means most of the detections are useful for matching.

Further, we draw the following conclusions from evaluating feature descriptors on the task of homography estimation:

- Even though ORB detector has the highest repeatability, the performance of ORB features is not as good during homography estimation.

- In case of sequences with illumination changes, SuperPoint descriptor performs the best. ORB and D2Net features fail the most in case of illumination changes.

- In case of viewpoint changes, SIFT gives the best performance. However, the performance of SuperPoint and LIFT is almost as good as SIFT.
- D2Net performs the worst when the accuracy threshold is low. Even though it is quite recent, the novelty in D2Net is not about giving the best results but in the ability to train detector and descriptor simultaneously in the same network.

Finally, we see that the nearest neighbor matches have relatively high reprojection error in case of D2Net. This is because the detector detects high level features which cannot be localized with the same accuracy as the low level features like corners or blobs.

6.1 Open source contributions

The code for extracting features and evaluating them is available at: https://gitlab.ipb.uni-bonn.de/amilioto/fancy-keypoints

Bibliography

[1] V. Balntas, K. Lenc, A. Vedaldi, and K. Mikolajczyk. HPatches: A Benchmark and Evaluation of Handcrafted and Learned Local Descriptors. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[2] D. DeTone, T. Malisiewicz, and A. Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 224–236, 2018.

[3] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, and T. Sattler. D2-net: A trainable cnn for joint description and detection of local features. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 8092–8101, 2019.

[4] S. Ehsan, N. Kanwal, A.F. Clark, and K.D. McDonald-Maier. Measuring the coverage of interest point detectors. In *International Conference Image Analysis and Recognition*, pages 253–261. Springer, 2011.

[5] W. Förstner, T. Dickscheid, and F. Schindler. Detecting interpretable and accurate scale-invariant keypoints. In *2009 IEEE 12th International Conference on Computer Vision*, pages 2256–2263. IEEE, 2009.

[6] D.G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *Intl. Journal of Computer Vision (IJCV)*, 60(2):91–110, 2004.

[7] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE transactions on pattern analysis and machine intelligence*, 27(10):1615–1630, 2005.

[8] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L.V. Gool. A comparison of affine region detectors. *International journal of computer vision*, 65(1-2):43–72, 2005.

[9] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: an efficient alternative to sift or surf. In *Proc. of the IEEE Intl. Conf. on Computer Vision (ICCV)*, 2011.

[10] K.M. Yi, E. Trulls, V. Lepetit, and P. Fua. Lift: Learned invariant feature transform. In *European Conference on Computer Vision*, pages 467–483. Springer, 2016.