Using Multi-Level Convolutional Information for Scale- and Viewpoint-Robust Local Features

Philipp Lindenberger [1] **Romeo Valentin** [1] **Mark Frey** [1] **Robin Wiethüchter** [2]

Abstract

Fast and robust keypoint detection and feature matching is an important task for many problems in computer vision, for example, Image Retrieval and Augmented Reality. Two key issues arise: keypoint detectors in real-time applications have to be both fast to compute and robust to change in illumination, viewpoint, and scale. In this work, we aim to increase the robustness of feature descriptors against scale and viewpoint changes while maintaining similar performance otherwise. To this end, we expand on previous work, using a pre-trained ResNet backbone and add an attention layer for keypoint selection, which we train directly on the quality of the keypoints. Critical to our goal, we forward multi-level convolutional activations directly to the final attention layer, bypassing further transformations and thus combining local with global information in the descriptor generation.

Figure 1: *Example of correspondences obtained with our model. Notably, our model is able to handle changes in scale, illumination, and viewpoint.*

1. Introduction

Local feature extraction is an essential task for many problems in computer vision, like Image Retrieval, Autonomous Navigation, and Augmented Reality. The general goal is to encode an image patch into a feature "descriptor" which represents this patch. Applying this encoding to representative patches on multiple images allows the computer to reason about the relation between two images. Then, matching up similar descriptors allows, for example, to compute a geometric viewpoint transformation or find an image in a database of images.

The demand for fast and precise feature descriptors grows rapidly, driven for example by the growing application of mobile image processing devices (e.g. smartphones and autonomous mobile robots), as well as the fast increase in labeled and unlabeled image data of complex scenes. Historically, feature descriptor algorithms were hand-designed, e.g. extracting corners or strong color gradients. The main challenge for these algorithms was producing similar descriptors under changing conditions like (i) illumination, (ii) viewpoint, and (iii) scale. In the past years, Deep Learning based feature descriptors have been shown to overcome these difficulties, learning both the proposal of distinctive patches as well as the encoding in an end-to-end manner (Fischer et al., 2014; Zagoruyko & Komodakis, 2015; Balntas et al., 2016).

In this work, we introduce a novel architecture and discuss its capabilities to obtain repeatable features which are robust against viewpoint and illumination changes. To this end, we propose combining low-, medium- and high-level features from a convolutional network together with an attention mechanism to generate robust descriptors which are precisely located. The proposed network can be seen as a

[*]Equal contribution [1]Department of Mathematics, ETH Zürich, Switzerland [2]Department of Computer Science, ETH Zürich, Switzerland. Correspondence to: <philipp.lindenberger@math.ethz.ch>, <rvalentin@student.ethz.ch>, <freymark@student.ethz.ch>, <robinw@student.ethz.ch>.

refinement layer with little overhead, where the backbone model (ResNet-50 pre-trained by DELG (Cao et al., 2020)) remains unchanged. We discuss the use of different loss functions and how they can be combined to overcome different challenges. An example of image correspondences obtained with our model can be seen in Fig. 1.

2. Related Work

One of the most famous local feature detection algorithms is SIFT (Lowe, 2004b) which uses gradients to find keypoints and calculate descriptors. The use of descriptors for keypoints in SIFT and related approaches (Bay et al., 2006; Mikolajczyk & Schmid, 2005; Harris et al., 1988; Lowe, 2004a; Mikolajczyk & Schmid, 2004; Mikolajczyk et al., 2005) provides robustness against viewpoint changes. Recently, learned feature extractors and descriptors through CNNs have been proposed, such as SuperPoint (DeTone et al., 2018), R2D2 (Revaud et al., 2019), D2-Net (Dusmanu et al., 2019), S2DNet (Germain et al., 2020), DELF (Noh et al., 2017), and DELG (Cao et al., 2020), replacing more classical approaches, like SURF (Bay et al., 2006), by leveraging automatic feature generation inherent in convolutional layers. All methods start with a standard convolutional backbone like ResNet (He et al., 2016), combined with a frontend and problem-specific loss functions, and preprocessing tricks to improve on robustness. Critically, D2-Net showcases how robust training for keypoints can be done directly on image correspondences by using the triplet margin ranking loss (Balntas et al., 2016; Mishchuk et al., 2017), which minimizes the distance between corresponding descriptors while maximizing it between unrelated ones. DELF and DELG introduce an attention layer at the final stage in order to select the best keypoints and improve viewpoint invariance. In DELG, they additionally aim to jointly train global and local descriptors for hierarchical retrieval. R2D2 tackles reliability and repeatability for keypoint matching with several losses, one of them encouraging peaks in the attention maps.

3. Method

In this section, we present our architecture and training details, as well as the considered loss functions for training the descriptor computation and selection end-to-end.

3.1. Architecture

The architecture consists of the following components: (i) a pre-trained ResNet-50 backbone, extracting pixel-wise image features, (ii) three autoencoders + upsampling, forwarding compressed embeddings of intermediate ResNet activations, (iii) an attention mechanism selecting representative keypoints, and (iv) the final descriptor extraction, combining the features from different scales into a set of repeatable features. Fig. 2 presents an overview of our model.

ResNet-50 backbone. In order to leverage previous successes in deep image processing, a ResNet-50 architecture is used as the "backbone" of the architecture, extracting image features on different scales. Pre-trained weights are loaded from the DELG architecture, where a ResNet-50 has been trained on a similar task. The weights of our "backbone" are then frozen in our training pipeline, which allows the combined evaluation of state-of-the-art image retrieval descriptors from DELG with our feature extraction in one forward-pass. Critically, we assume that the channels at a pixel location at any part within the ResNet can be interpreted as a sort of descriptor since they encode information about the pixel and its neighborhood. Thus, the backbone provides pixel-wise (dense) descriptors, and we aim to extract a sparse set of distinctive and repeatable descriptors. Furthermore, the key insight to our approach is that in order to overcome the challenge of scale invariance, low- and high-level features need to be combined. Research has shown that typically, low-level features appear in earlier layers of convolutional architectures and become more high-level the deeper the network goes. We extract activations at three different points of the backbone and combine them to form our final descriptors. Specifically, we extract the outputs of the second, third, and fourth block of the ResNet-50 architecture. Since we are not interested in classification, the rest of the ResNet architecture is truncated.

Autoencoders. Convolutional architectures like ResNet generally increase the number of channels and decrease the resolution as the network depth increases. Thus, in order to combine channels from different depths in the architecture, both the resolution and the number of channels have to be matched up. In our approach, we first extract the activations at three different levels ("early", "middle", and "deep") and upsample the latter to the resolution of the early level feature map. Then, for each level, an autoencoder is trained, reducing the number of channels to a common number $N_{enc.}$. For each input image, we obtain three feature maps, each with the same resolution and number of channels. We propose training the autoencoder on a joint reconstruction and correspondence loss to incorporate both the discriminativeness of the DELG features and learn viewpoint robustness from pixel-wise correspondences.

Attention Mechanism. Besides obtaining robust feature descriptors, the goal of the algorithm is extracting meaningful feature locations – i.e. image patches where feature descriptors can be reconstructed from different angles or lighting conditions. For example, in hand-crafted algorithms corners are typically a good choice for repeatable features, since they strongly differ from other features and can be reliably and repeatably obtained under varying viewpoint and

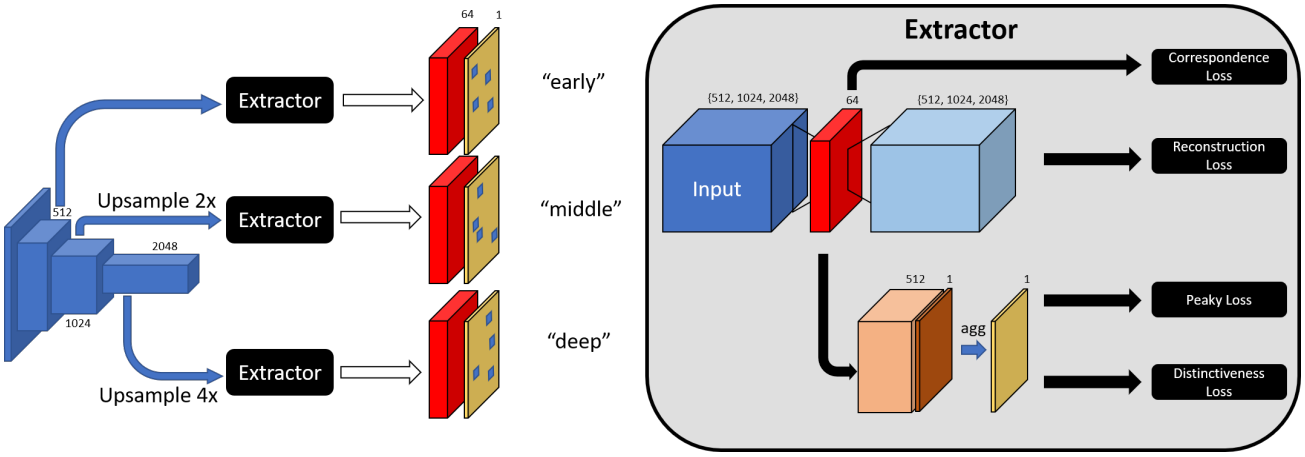

Figure 2: Proposed architecture. *First the input image is forwarded through the ResNet50-backbone (He et al., 2016), using the weights of DELG (Cao et al., 2020) (which are not refined in our training). We extract the output of the last three ResNet-blocks and upsample the last two to obtain featuremaps with equal resolution. The descriptors are forwarded to an autoencoder, which is jointly trained on a correspondence loss and a reconstruction loss. The embedded 64-dimensional* ***dense descriptors*** *are used to train an attention model to learn repeatable* ***keypoint scores****, where agg is the Softmax. Both attention model and autoencoder are jointly learned.*

illumination conditions. The attention mechanism should therefore be trained to choose specifically robust locations. It assigns each pixel in the output an attention score in $[0, 1]$, which can be interpreted as a robustness score that can be thresholded. We train the attention model on top of our dense descriptors. Similar to DELG, we apply an additional 1x1 convolutional layer to 512 channels, followed by batch normalization and another 1x1 convolution down to one channel. To obtain outputs in $[0, 1]$, we use softmax similar to R2D2 (Revaud et al., 2019). This architecture allows us to introduce a reasonable amount of weights while maintaining local information from our dense descriptors.

Feature Extraction. To extract our features we follow the approach used in the DELF and DELG paper. There they used information of the convolution (stride, effective padding, and receptive field) to calculate adjusted keypoint locations. Though, their keypoint extractor relies on a regular grid with constant approximation between grid cells, which leads to less accurate keypoint localization. In their implementation, they overcome this problem by using an image pyramid that detects keypoints at different scales. Our method circumvents this requirement with a slightly adapted version of the feature extraction found in the D2Net implementation, utilizing the encoded scale information from different convolutional layers.

We aim to detect keypoints independently in the three proposed attention layers by thresholding. In the D2Net extraction, the keypoint locations are refined with an approach similar to the method proposed by the SIFT paper, using local gradients around the original detections. After refinement, the descriptors are interpolated at the newly calculated positions. Descriptors are then extracted from the encoded dense features at the detected spatial location, in the respective convolutional layer. Finally, we apply L2 normalization on the descriptors. We experimented with combining descriptors (adding, averaging) and weighting them with the attention scores but could not observe notable improvements.

3.2. Loss functions

Desirable properties of a good feature are repeatability, distinctiveness, and reliability. Here, we briefly present some of the loss functions we used, aimed at improving these properties. Commonly, descriptor losses are evaluated (i) just on a single descriptor or image patch (e.g. for obtaining a local distinctiveness maximum), (ii) on a triplet of image patches (query, true and false correspondence), or (iii) on a query as well as all possible correspondences in the other image, e.g. for employing a ranking loss. In our model, we use the encoded features in our autoencoder with 64 dimensions as descriptors and use the attention scores as a measure for keypoint quality.

Correspondence loss. The correspondence loss, as introduced in the D2Net paper, aims for both *distinctiveness* and *repeatability* at the same time, using a triplet margin ranking loss. Given two images $\mathcal{I}_1$ and $\mathcal{I}_2$ as well as a set of one-to-one correspondences, the loss (Balntas et al., 2016; Mishchuk et al., 2017) is based on triplets of image patches: A query descriptor d_q from $\mathcal{I}_1$, the corresponding descriptor d_c from $\mathcal{I}_2$ and a false descriptor d_f from $\mathcal{I}_2$. The correspondence loss minimizes the distance between the correct correspondences while maximizing the distance between

the query and its closest false patch, up to some margin M[1]:

$$m(q,c) = \max(0, M + ||d_q - d_c||_2^2 - ||d_q - d_f(q)||_2^2) \quad (1)$$

Intuitively, this loss aims to increase distinctiveness of the descriptors. Compared to D2Net (Dusmanu et al., 2019), we omit the repeatability score since we aim to learn repeatability with an attention mechanism. The final correspondence loss is then defined as the sum over all pixel-wise correspondences $(q,c) \in \mathcal{C}$:

$$\mathcal{L}_{corr}(\mathcal{I}_1, \mathcal{I}_2) = \sum_{(q,c)\in\mathcal{C}} m(q,c) \quad (2)$$

Autoencoder loss. The final loss used in the autoencoder is a weighted sum of the correspondence loss and a reconstruction loss, which aims to forward the discriminative descriptor information from the DeLG ResNet backbone. Given the dense input features $d_{in} \in \mathbb{R}^{HxWxC}$ from the ResNet-50 layers, the encoded 64-dimensional features d_e and the decoded features d_r, we introduce the following loss to train our autoencoder, summing the reconstruction loss over both images $i \in (1,2)$:

$$\mathcal{L}_{auto}(\mathcal{I}_1, \mathcal{I}_2) = \mathcal{L}_{corr}(\mathcal{I}_1, \mathcal{I}_2) + \sum_{i\in(1,2)} \frac{||d_{in,i} - d_{rec.,i}||_2^2}{H \cdot W \cdot C} \quad (3)$$

Distinctiveness loss. The distinctiveness loss, introduced in the D2D architecture (Wiles et al., 2020), enforces the attention score $a_{ij} \in [0,1]$ to be close to one if the correspondence is unique (distinctive), else close to zero the more similar areas can be found. Given a correspondence location p in one image and a set of false correspondences $\mathcal{F}(p)$ in the other image $\mathcal{I}$, the loss counts the number of false descriptors with distance within the margin:

$$m_x(p) = \sum_{d_f\in\mathcal{F}} \mathbb{1}(||d_p - d_f||_2^2 < M) \quad (4)$$

Similar to D2D, we use $M = 1$. The loss for a_{ij} can then be written as a distance loss to a smoothed label of the count:

$$L_{dist}(p, \mathcal{I}) = \left| a(p) - \frac{1}{(1 + m_x(p))^{0.25}} \right| \quad (5)$$

The final loss is then the sum over all correspondences divided by the number of positive correspondences L_c:

$$\mathcal{L}_{dist}(\mathcal{I}_1, \mathcal{I}_2) = \frac{1}{L_c} \sum_{(q,c)\in\mathcal{C}} L_{dist}(q, \mathcal{I}_2) + L_{dist}(c, \mathcal{I}_1) \quad (6)$$

[1]For the sake of clarity, the notation is slightly simplified. The full equations can be found in the paper on the D2 architecture.

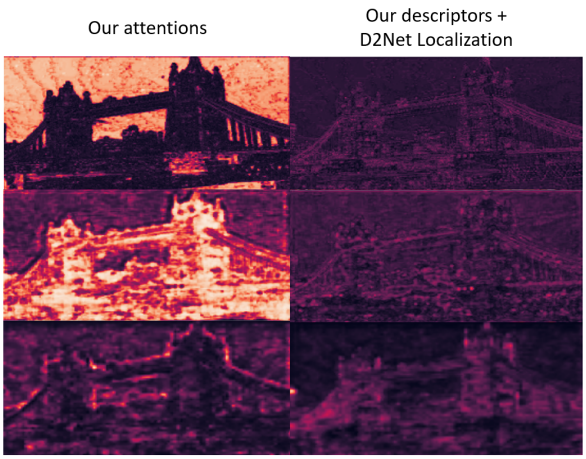

Figure 3: Attentions. *Our attention layer (left) clearly highlights structural information of different scale, i.e. early, middle and deep activations. On the right we visualized the hard detections from D2Net (Dusmanu et al., 2019) on our descriptors, which are less prominent but contain finer structural information.*

Peaky loss. The soft detection scores can additionally be trained using the peaky loss, introduced in the R2D2 paper. Let $\mathcal{P}$ be the set of overlapping image patches of size NxN, then the peaky loss

$$\mathcal{L}_{peaky}(\mathcal{I}) = 1 - \frac{1}{|\mathcal{P}|} \sum_{p\in\mathcal{P}} (\max_{(i,j)\in p} a_{ij} - \operatorname{mean}_{(i,j)\in p} a_{ij}) \quad (7)$$

makes the attention scores a_{ij} "peaky" for each NxN image patch, i.e. there is a single pixel that evaluates much higher than the mean.

Attention loss. To train our attention layers, we combine the distinctiveness loss and the peaky loss. The final loss is then:

$$\mathcal{L}_{att}(\mathcal{I}_1, \mathcal{I}_2) = \mathcal{L}_{dist}(\mathcal{I}_1, \mathcal{I}_2) + \lambda \sum_{i\in(1,2)} \mathcal{L}_{peaky}(\mathcal{I}_i) \quad (8)$$

where we use $\lambda = 0.1$ to reduce the weight of the peaky loss.

3.3. Training Setup

We train the different parts of our architecture individually. First of all, the weights of the backbone are loaded from pre-trained weights and subsequently frozen, in order to decrease computational demands. Then, the autoencoders are trained individually on a reconstruction loss, stopping the gradient flow at the extraction of the ResNet activations. Finally, the attention mechanism is trained, using the fixed-weight backbone and autoencoders.

MegaDepth Dataset. For training the model, the MegaDepth dataset (Li & Snavely, 2018) is used. The

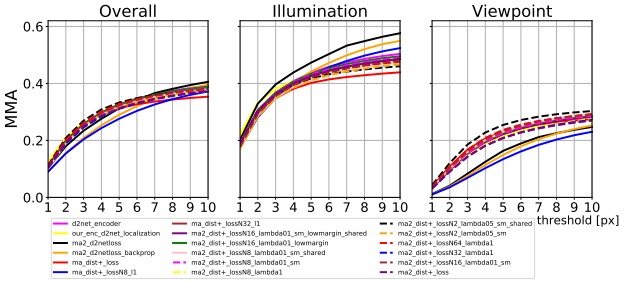

Figure 4: *HPatches benchmark results of* ***our proposed losses and models*** *in mean matching accuracy (MMA). Note that the y-axis is rescaled in comparison to Fig. 5.*

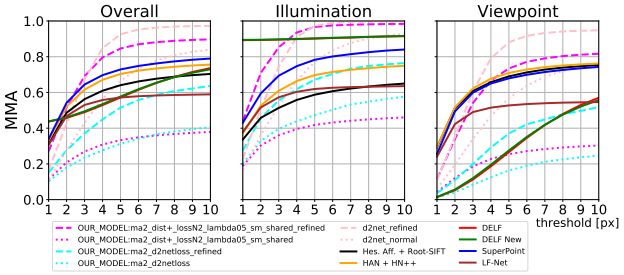

Figure 5: *HPatches benchmark results of* ***our model in comparison to other models*** *measured in mean matching accuracy (MMA). Note that the models labeled with* refined *at the end use a refinement method in the matching process based on RANSAC. We also reevaluated D2-Net with this method.*

dataset contains a variety of images taken under different conditions of 196 landmarks, typically buildings. All images contain RGB-D data, and following the pipeline of D2Net (Dusmanu et al., 2019) this dataset can be used to obtain pixel-wise correspondences through Structure-from-Motion. We use 100 correspondences per landmark, randomly cropped to 256x256. Similar to D2Net, we split the dataset in training (118) and validation (78).

Training parameters. The autoencoders are constructed with $N_{enc.} = 64$ and the full model is trained with the ADAM optimizer (Kingma & Ba, 2014), a learning rate of $5\text{e}{-4}$ and a batch size of 64. We found that using $N_{enc.} = 32$ or $N_{enc.} = 16$ significantly worsened results, while larger descriptors are slightly more accurate, but also significantly more expensive during matching.

4. Evaluation and Discussion

HPatches Benchmark. As shown in Figure 4 and 5, we used the HPatches dataset to benchmark our models and compare them to other state-of-the-art approaches like SIFT, DELF, D2-Net, or SuperPoint. The HPatches dataset was created to evaluate local image descriptors. The dataset contains 108 sequences that each consist of 6 images of the same location but differ in either illumination or viewpoint. For each sequence, extracted keypoints are matched with the other 5 images. Therefore, the corresponding descriptors and a nearest neighbor search is used. The matched keypoints are evaluated by projecting each keypoint of the first image onto the other images with a homography and measuring the distance of the projected and the corresponding matched keypoint in the target image. A match is considered correct if the distance is below a threshold which is summarized for all keypoints as the mean matching accuracy (**MMA**).

Final Model. Figure 5 shows our performance compared to various state-of-the-art approaches. With basic descriptor matching, the lack of descriptiveness in our features due to their lower dimension (64) and the still imprecise detections limit their quality at larger thresholds. Though, this can be overcome by matching features with RANSAC, yielding almost similar results to D2Net (also with RANSAC) but with significantly smaller features (64 vs. 512 in D2Net).

Attentions.. The proposed model extracts attentions that highlight structural dominant areas in the scene at different scales, see Fig. 3. We compared our attentions with the output of the D2Net detection strategy using their hard detection module, which yields finer structure but less prominent detections. Though, since this was applied to our encoded features, this shows that the proposed autoencoder does encode spatial information without significant losses in accuracy. A shortcoming of our attention method is the detection of "reliable" keypoints in the sky. This could potentially be improved by introducing a saliency metric to the model.

Ablation studies. We tested different attention model architectures. The first was similar to R2D2, where we used 1x1 convolutional layers to directly downsample to a 2D attention map, then performing softmax to obtain a 1D attention map. Though, the lack of parameters, in this case, worsened our results in the viewpoint benchmark (the map looked really similar to the score map of the dense descriptors, which are not always reliable indicators of good keypoints). Thus, we added an intermediate 1x1 convolutional layer to gain more weights, which improved our extractor in viewpoint robustness.

5. Conclusion

In this study, we have shown that forwarding intermediate activations from the convolutional network can be used to replace the need for an image-pyramid scheme, which is commonly used to improve scale-invariance. Furthermore, we were able to reduce the size of our descriptors to 64 dimensions while maintaining good detections after post-processing matches with RANSAC, whereas other state-of-the-art architectures like D2-Net require 512 dimensions.

References

Balntas, V., Riba, E., Ponsa, D., and Mikolajczyk, K. Learning local feature descriptors with triplets and shallow convolutional neural networks. In *Bmvc*, volume 1, pp. 3, 2016.

Bay, H., Tuytelaars, T., and Van Gool, L. Surf: Speeded up robust features. In *European conference on computer vision*, pp. 404–417. Springer, 2006.

Cao, B., Araujo, A., and Sim, J. Unifying deep local and global features for image search. *arXiv*, pp. arXiv–2001, 2020.

DeTone, D., Malisiewicz, T., and Rabinovich, A. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 224–236, 2018.

Dusmanu, M., Rocco, I., Pajdla, T., Pollefeys, M., Sivic, J., Torii, A., and Sattler, T. D2-net: A trainable cnn for joint description and detection of local features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8092–8101, 2019.

Fischer, P., Dosovitskiy, A., and Brox, T. Descriptor matching with convolutional neural networks: a comparison to sift. *arXiv preprint arXiv:1405.5769*, 2014.

Germain, H., Bourmaud, G., and Lepetit, V. S2dnet: Learning accurate correspondences for sparse-to-dense feature matching. *arXiv preprint arXiv:2004.01673*, 2020.

Harris, C. G., Stephens, M., et al. A combined corner and edge detector. In *Alvey vision conference*, volume 15, pp. 10–5244. Citeseer, 1988.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Li, Z. and Snavely, N. Megadepth: Learning single-view depth prediction from internet photos, 2018.

Lowe, D. G. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60 (2):91–110, 2004a.

Lowe, D. G. Method and apparatus for identifying scale invariant features in an image and use of same for locating an object in an image, March 23 2004b. US Patent 6,711,293.

Mikolajczyk, K. and Schmid, C. Scale & affine invariant interest point detectors. *International journal of computer vision*, 60(1):63–86, 2004.

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

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

Mishchuk, A., Mishkin, D., Radenovic, F., and Matas, J. Working hard to know your neighbor's margins: Local descriptor learning loss. In *Advances in Neural Information Processing Systems*, pp. 4826–4837, 2017.

Noh, H., Araujo, A., Sim, J., Weyand, T., and Han, B. Large-scale image retrieval with attentive deep local features. In *Proceedings of the IEEE international conference on computer vision*, pp. 3456–3465, 2017.

Revaud, J., Weinzaepfel, P., Souza, C. D., Pion, N., Csurka, G., Cabon, Y., and Humenberger, M. R2d2: Repeatable and reliable detector and descriptor, 2019.

Wiles, O., Ehrhardt, S., and Zisserman, A. D2d: Learning to find good correspondences for image matching and manipulation. *arXiv preprint arXiv:2007.08480*, 2020.

Zagoruyko, S. and Komodakis, N. Learning to compare image patches via convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4353–4361, 2015.