SIFT-AID: BOOSTING SIFT WITH AN AFFINE INVARIANT DESCRIPTOR BASED ON CONVOLUTIONAL NEURAL NETWORKS

Mariano Rodríguez, Gabriele Facciolo, Rafael Grompone von Gioi, Pablo Musé, Jean-Michel Morel, Julie Delon

SIFT-AID: BOOSTING SIFT WITH AN AFFINE INVARIANT DESCRIPTOR BASED ON CONVOLUTIONAL NEURAL NETWORKS

M. Rodríguez,† *G. Facciolo,*† *R. Grompone von Gioi,*† *P. Musé,*§ *J.-M. Morel,*† *and J. Delon*‡

† CMLA, ENS Paris-Saclay, CNRS, Université Paris-Saclay, 94235 Cachan, France
§ IIE, Universidad de la República, Uruguay
‡ MAP5, Université Paris Descartes, France

ABSTRACT

The classic approach to image matching consists in the detection, description and matching of keypoints. The descriptor encodes the local information around the keypoint. An advantage of local approaches is that viewpoint deformations are well approximated by affine maps. This motivated the quest for affine invariant local descriptors. Despite numerous efforts, such descriptors remained elusive, ultimately resulting in the compromise of using viewpoint simulations to attain affine invariance. In this work we propose a CNN-based patch descriptor which captures affine invariance without the need for viewpoint simulations. This is achieved by training a neural network to associate similar vectorial representations to patches related by affine transformations. During matching, these vectors are compared very efficiently. The invariance to translation, rotation and scale is still obtained by the first stages of SIFT, which produce the keypoints. The proposed descriptor outperforms the state-of-the-art in retaining affine invariant properties.

***Index Terms*—** image comparison, affine invariance, IMAS, SIFT, RootSIFT, convolutional neural networks.

1. INTRODUCTION

The classic approach to image matching consists in three steps: detection, description and matching [1]. First, keypoints are detected in both images to be compared. Second, regions around these points are described by local descriptors. Finally, all these descriptors are compared and possibly matched. Both the detection and description steps are usually designed to ensure some invariance to various geometric or radiometric changes. A benefit of local descriptors is that viewpoint deformations are well approximated by affine maps. Indeed, for any smooth deformation, its first order Taylor approximation is an affine map. This observation has motivated the development of comparison methods based on local descriptors that are as affine invariant as possible.

The Titan V used for this research was donated by the NVIDIA Corporation. Programme ECOS Sud UdelaR - Paris Descartes U17E04. We thank Pierre Perrault for fruitful discussions.

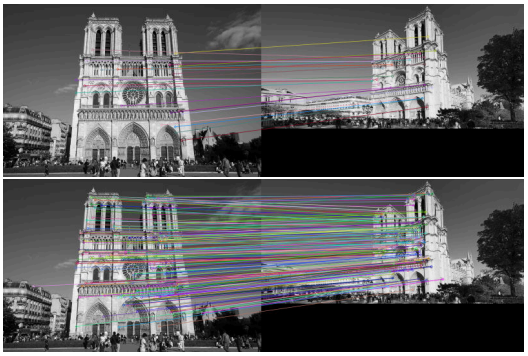

Fig. 1: Top: matches by Affine-RootSIFT (48). Bottom: matches by the proposed SIFT-AID method (295).

To ensure invariance to affine transforms, some authors have proposed moment-based region detectors [2, 3] including the Harris-Affine and Hessian-Affine region detectors [4, 5]. Locally affine invariant region detectors can also be based on edges [6, 7], intensity [8, 7], or entropy [9]. Finally, the detectors MSER (Maximally Stable Extremal Region) [10] and LLD (Level Line Descriptor) [11, 12, 13] both rely on image level lines. Yet the affine invariance of these desciptors in images acquired with real cameras is limited by the fact that optical blur and affine transforms do not commute, as shown in [14]. Thus, none of the previously mentioned descriptors can be considered fully affine invariant. In [15], RootSIFT [16] was reported to be the robustest descriptor to affine viewpoint changes (up to $60°$). To overcome this limitation, several simulation-based solutions have been proposed: ASIFT [17], FAIR-SURF [18], MODS [19], Affine-AC-W [20]. Some optimal versions have been proposed in [21], including Optimal Affine-RootSIFT, which was proven to be the best choice.

On the other hand, local descriptors, which once were manually-designed, are currently being learned from data, with the promise of a better performance. Mimicking the classic process of image matching, they learn a similarity measure between image patches. In [22], three similarity score architectures were introduced (CNN + a decision net-

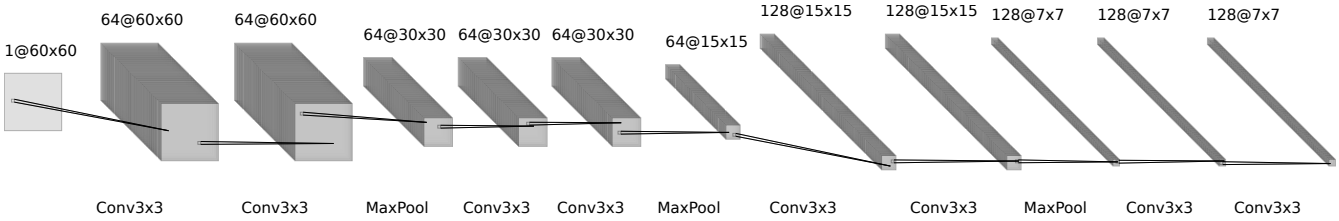

Fig. 2: The proposed descritor is computed using a CNN that produces a feature vector of dimension 6272.

work). For stereo matching, two architectures based on CNNs were proposed in [23], one of them computing the similarity score with the cosine proximity operator.

CNN-based geometric matching between images has also been tested for the case of affine and homography transformations [24, 25]. In [24], the POOL4 layer of the VGG-16 network [26] was used for acquiring features from images and correlation maps fed to a regression network that outputs the best affine transform fitting the query into the target image. In a direct approach, the authors of [25] trained a network to estimate the homography relating the query to the target image. Both [24, 25] were trained on synthetically generated images, however neither of them took into account the blur caused by camera zoom-out or tilt.

In this paper we combine manually-designed and learned methods in order to obtain a fast affine invariant image matching algorithm, capable of capturing strong viewpoint changes. The proposed method is based on the first stages of SIFT [1, 27], which ensure invariance to similarity transformations (translations, rotations and zooms) up to small perturbations (see [28] for a mathematical proof). At this point the SIFT descriptor is replaced by a neural network (Figure 2) that takes a 60×60 patch as input and produces a 6272-element vector descriptor. The network is trained on a dataset containing pairs of patches related by affine transformations, aiming at producing similar descriptor vectors for affine pairs and dissimilar vectors otherwise [23].

A simple way of measuring similarity between vector descriptors is through the cosine proximity operator, i.e. $\cos(\mathbf{x}, \mathbf{y}) := \frac{\langle \mathbf{x}, \mathbf{y} \rangle}{\|\mathbf{x}\| \|\mathbf{y}\|}$. Therefore, we train the network to cluster similar descriptors with respect to angle. Finally, only the sign of each vector component is kept, leading to a binary descriptor. This allows to save memory and accelerate the matching process, while keeping the same level of performance and discriminative power. Figure 1 presents an example of the proposed method compared to the Affine-RootSIFT method.

2. AFFINE VIEWPOINT SIMULATION

Let us now focus on how to properly model affine viewpoints, which is needed for generating synthetic data to train our descriptor. Let u denote an image, $\mathbf{A}$ the set of affine maps and define $Au(\mathbf{x}) = u(A\mathbf{x})$ for $A \in \mathbf{A}$. We define $\mathbf{A}^+ = \{A \in \mathbf{A} | \det(A) > 0\}$. We call $\mathcal{S}$ the set of similarity transformations, which are any combination of translations, rotations and zooms. Finally we define the set $\mathbf{A}^+_* = \mathbf{A}^+ \setminus \mathcal{S}$, where we exclude pure similarities. It was proven in [14] that every $A \in \mathbf{A}^+_*$ is uniquely decomposed as

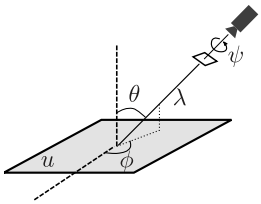

Fig. 3: Geometric interpretation of equation (1).

$$A = \lambda R_1(\psi) T_t R_2(\phi), \tag{1}$$

where R_1, R_2 are rotations and $T_t = \begin{bmatrix} t & 0 \\ 0 & 1 \end{bmatrix}$ with $t > 1$, $\lambda > 0$, $\phi \in [0, \pi)$ and $\psi \in [0, 2\pi)$. Furthermore, the above decomposition comes with a geometric interpretation (see Figure 3) where the longitude ϕ and latitude $\theta = \arccos \frac{1}{t}$ characterize the camera's viewpoint angles (or tilt), ψ parameterizes the camera roll and λ corresponds to the camera zoom.

A digital image $\mathbf{u}$ obtained by any camera at infinity can be written as $\mathbf{u} = \mathbf{S}_1 \mathbb{G}_1 A \mathcal{T} u_0$ where $\mathbf{S}_1$ is the image sampling operator (on a unitary grid), A a linear map, $\mathcal{T}$ a planar translation, u_0 a continuous image and $\mathbb{G}_\delta$ denotes the convolution by a Gaussian kernel broad enough to ensure no aliasing by δ-sampling. Unfortunately, $\mathbb{G}_1$ and A do not commute when A involves a tilt or a zoom. As a consequence, a simple warping $A(\mathbf{u}_0)$ of the frontal image $\mathbf{u}_0 := \mathbf{S}_1 \mathbb{G}_1 u_0$ is not a correct optical affine simulation of $\mathbf{u}$. As stated in [14, 15], the correct way of simulating a tilt t in the x-direction is:

$$\mathbf{u} \to \mathbf{S}_1 T^x_t \mathbb{G}^x_{\sqrt{t^2-1}} I \mathbf{u}, \tag{2}$$

where I is the Shannon-Whittaker interpolator and the superscript x indicates the operator takes place only in the x-direction. We denote $\mathbb{T}^x_t := T^x_t \mathbb{G}^x_{\sqrt{t^2-1}} I$. Similarly for y.

It is clear that there is loss of information due to the blur; indeed, the operator $\mathbb{T}^x_t$ is not invertible. Which means that, depending on the image $\mathbf{u}$, there might not be any optical transformation $\mathbb{A}$ satisfying $\mathbb{A}(\mathbf{u}_1) = \mathbf{u}_2$ or $\mathbf{u}_1 = \mathbb{A}(\mathbf{u}_2)$. Consider, for example, $\mathbf{u}_1 = \mathbb{T}^x_t \mathbf{u}$ and $\mathbf{u}_2 = \mathbb{T}^y_t \mathbf{u}$.

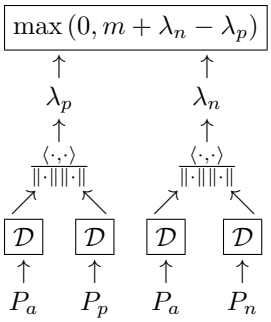

Fig. 4: Diagram of the siamese network for training $\mathcal{D}$.

With that in mind, we design a data generation scheme that, given an image $\mathbf{u}$ and a pair of random affine transformations $\mathbb{A}_1$ and $\mathbb{A}_2$, simulates affine views $\mathbf{u}_1 = \mathbb{A}_1(\mathbf{u})$ and $\mathbf{u}_2 = \mathbb{A}_2(\mathbf{u})$. Both $\mathbb{A}_1, \mathbb{A}_2$ with maximal viewpoint angles up to $75°$ with respect to $\mathbf{u}$. Instances of $\mathbf{u}$ are provided accordingly from three independent MS-COCO [29] datasets for training, validation and test. Patch pairs seeing the same scene from $\mathbf{u}_1$ and $\mathbf{u}_2$ are said to belong to the same *class* and will be used to train the descriptor network.

3. DESCRIPTORS AND MATCHING CRITERIA

Inspired on [23], our descriptor network $\mathcal{D}$ is trained to produce similar descriptor vectors for patch pairs of the same class, and dissimilar vectors for patch pairs of different class. The network architecture is adapted from [25], see Figure 2. It consists of 4 blocks of two convolutional layers each followed by batch normalization and ReLU activations. Between each block a max-pooling layer is introduced. A 2D Spatial Dropout with a probability 0.5 is applied after the last convolutional layer.

Here, dropout is not used to avoid over-fitting but to encourage the descriptor network to use all the dimensions of the feature vector. In addition, it does facilitate the learning process: the validation loss has proved to be much more stable than without dropout.

The affine approximation holds locally, which suggests the use of small patch sizes; on the other hand, small patches entail less information, leading to insufficient descriptions. As a compromise, we set the patch size to 60×60, which provides a good balance between locality and enough viewpoint information.

Training with hinge loss. During training, the descriptor network is immersed into a siamese network, represented in Figure 4. The siamese network consists of two identical subnetworks joined at the top by a virtual layer that computes the hinge loss between their two outputs:

$\lambda_p = \cos(\mathcal{D}(P_a), \mathcal{D}(P_p))$, $\lambda_n = \cos(\mathcal{D}(P_a), \mathcal{D}(P_n))$, where patches P_a, P_p belong to the same class whereas P_n does not. While training, we simulate random contrast changes on all input patches. The hinge loss, i.e. $L(\lambda_p, \lambda_n) := \max(0, m + \lambda_n - \lambda_p)$, is used with parameter m set to 0.2 in our experiments.

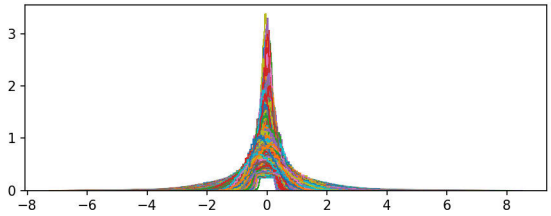

Fig. 5: Density plots from each BigAID dimension (6272), computed over $5 \cdot 10^4$ BigAID descriptions of random patches from the test dataset.

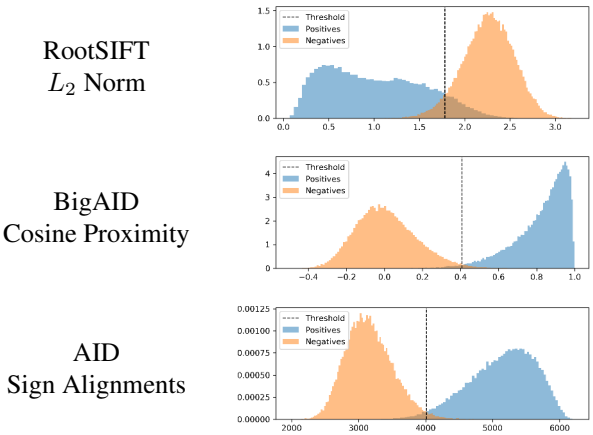

Fig. 6: Positive and negative density estimation on measurements. For that, $6 \cdot 10^5$ random intra and extra class pairs were used. The vertical line depicts the threshold minimizing both error probabilities: false negatives and false positives.

Binary descriptor and matching. When training is complete, the descriptor network is plugged out from the siamese network and expected to produce descriptors that capture affine invariant properties from input patches. We call this description *BigAID* (6272 floats). Figure 5 shows density estimations on each BigAID dimension. Notice the involvement of all the dimensions in the description and the symmetry of all densities around zero. With this in mind, we propose a new affine invariant descriptor, that we call *AID* (6272 bits), which only keeps the sign information from the BigAID. Two AID descriptors $\mathbf{x}$ and $\mathbf{y}$ are consequently matched via the sign alignment measure, i.e. $\sum_i \mathbb{1}_{\text{sign}(x_i)=\text{sign}(y_i)}$. Intra- and extra-class measure density estimations are shown in Figure 6 for RootSIFT (128 floats = 4096 bits) and our descriptors, suggesting that for the BigAID and AID descriptors, a simple thresholding of their respective measures is sufficient to single out classes.

	Test I: Using SIFT keypoints					Test II: Using Affine-RootSIFT keypoints				
	# keypoints per image		Without viewpoint simulations			# keypoints per image		With viewpoint simulations	Without viewpoint simulations	
	query	target	RS	BigAID	AID	query	target	A-RS	BigAID*	AID*
coke	5443	5670	115	1316	**1409**	28609	31965	1395	5298	**5346**
notredame	2285	1235	14	282	**295**	11739	6444	48	590	**731**
arc	1384	1387	40	**445**	420	5719	4759	244	579	**600**
graffiti	1661	3117	0	**182**	172	14290	15225	**613**	502	516
adam	269	192	30	67	**69**	3647	2364	484	496	**520**

Table 1: Viewpoint performance test. RS, A-RS, BigAID and AID denote Homography consistent Matches found by ORSA for RootSIFT, Affine-RootSIFT, BigAID and AID. The Second-Nearest-Neighbor ratio in RootSIFT and Affine-RootSIFT was set to 0.8. The thresholds for BigAID and AID were 0.4 and 4000, respectively. The star (*) indicates on oracle keypoints.

	A-RootSIFT L_2 norm		SIFT-BigAID Cos. Prox.		SIFT-AID Sign Align.	
	ET-D	ET-M	ET-D*	ET-M	ET-D*	ET-M
coke	4.500	26.730	9.876	116.767	9.838	4.402
notredame	1.930	1.930	3.272	10.829	3.177	0.540
arc	1.520	0.670	2.581	7.372	2.465	0.389
graf	2.790	6.180	4.441	20.027	4.369	0.895
adam	1.210	0.230	0.601	0.241	0.525	0.048

Table 2: Time performance for Affine-RootSIFT, SIFT-BigAID and SIFT-AID. Elapsed time (in seconds) in building descriptors (ET-D) and matching them (ET-M); The star (*) denotes GPU time.

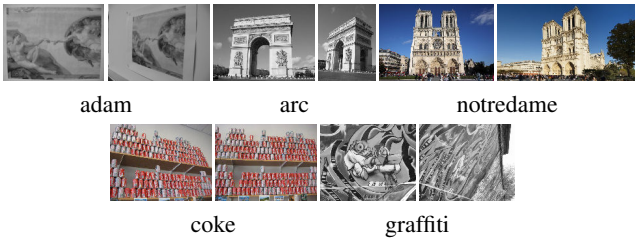

Fig. 7: Viewpoint challenge dataset.

4. EXPERIMENTS

Up until now, the descriptor network $\mathcal{D}$ has only seen optically simulated input patches. Figure 7 provides a realistic viewpoint challenge dataset in the form of 5 pairs of images. Given a fixed set of SIFT keypoints from these images, the proposed methods are compared against RootSIFT in the section Test I of Table 1. The number of homography-consistent matches found by ORSA [30] (an a-contrario validated RANSAC) shows the superiority of the AID descriptors with respect to RootSIFT. AID is more compact and has a similar performance to BigAID. For these reasons, we prefer the AID descriptor and we call *SIFT-AID* the matching method resulting from its combination with SIFT keypoints.

The A-RS column (Test II) in Table 1 shows the number of homography consistent matches for Affine-RootSIFT. Notice how SIFT-AID has comparable performances without using viewpoint simulations. But in some cases, it yields less matches, as for the *adam* pair. Why? As stated in [15], Affine-RootSIFT has about 7 times more keypoints than SIFT. Some of those keypoints come exclusively from simulated versions of the input images, i.e., they do not belong to the Gaussian pyramid of the original input images. To further test AID descriptors, we define an oracle yielding precise keypoints in the original Gaussian pyramid best approximating each keypoint from the first stages of Affine-RootSIFT. Keypoints provided by this oracle are the best possible choices that could have been found by the first stages of SIFT. Table 1 (Test II) also shows the number of homography consistent matches for oracle + AID descriptors. This experiment reveals that both AID and BigAID would have been sufficient to identify almost all Affine-RootSIFT matches, provided that proper keypoints had been correctly spotted by the first stages of SIFT. In the case of the *graffiti* pair, most of the missing matches for AID descriptors involve viewpoint angles close to $75°$, the maximal viewpoint angle present in the training dataset.

Finally, Table 2 shows the time consumed by SIFT-AID and Affine-RootSIFT in building descriptors and matching them[1] (non optimized codes). Overall, the SIFT-AID method can achieve results in less time than Affine-RootSIFT.

5. CONCLUSION

We proposed a CNN image patch descriptor capturing affine invariance. Our experiments show that the SIFT-AID method attains a performance comparable to Affine-RootSIFT without the necessity of using viewpoint simulations. Most of the missing matches are due to SIFT's keypoint detection step failures; more work is needed to improve this step. The viewpoint robustness of the proposed method could be further extended by affine simulations techniques similar to those in [14, 15]. This extension will be the focus of future work. Finally, the descriptor network architecture could be optimized to improve the performance.

Reproducibility: The source code of SIFT-AID is available at `https://rdguez-mariano.github.io/pages/sift-aid`

[1]Hardware settings: (CPU) Intel(R) Core(TM) i7-6700HQ 2.60GHz; (GPU) NVIDIA Corporation GM204GLM [Quadro M5000M].

6. REFERENCES

[1] D. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.

[2] T. Lindeberg and J. Garding, "Shape-adapted smoothing in estimation of 3-D depth cues from affine distortions of local 2-D brightness structure," *ECCV*, pp. 389–400, 1994.

[3] A. Baumberg, "Reliable feature matching across widely separated views," *CVPR*, vol. 1, pp. 774–781, 2000.

[4] K. Mikolajczyk and C. Schmid, "An affine invariant interest point detector," *ECCV*, vol. 1, pp. 128–142, 2002.

[5] K. Mikolajczyk and C. Schmid, "Scale and Affine Invariant Interest Point Detectors," *IJCV*, vol. 60, no. 1, pp. 63–86, 2004.

[6] T. Tuytelaars, L. Van Gool, and Others, "Content-based image retrieval based on local affinely invariant regions," *Int. Conf. on Visual Information Systems*, pp. 493–500, 1999.

[7] T. Tuytelaars and L. Van Gool, "Matching Widely Separated Views Based on Affine Invariant Regions," *IJCV*, vol. 59, no. 1, pp. 61–85, 2004.

[8] T. Tuytelaars and L. Van Gool, "Wide baseline stereo matching based on local, affinely invariant regions," *BMVC*, pp. 412–425, 2000.

[9] T. Kadir, A. Zisserman, and M. Brady, "An Affine Invariant Salient Region Detector," in *ECCV*, 2004, pp. 228–241.

[10] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *IVC*, vol. 22, no. 10, pp. 761–767, 2004.

[11] P. Musé, F. Sur, F. Cao, and Y. Gousseau, "Unsupervised thresholds for shape matching," *ICIP*, 2003.

[12] P. Musé, F. Sur, F. Cao, Y. Gousseau, and J. M. Morel, "An A Contrario Decision Method for Shape Element Recognition," *IJCV*, vol. 69, no. 3, pp. 295–315, 2006.

[13] F. Cao, J.-L. Lisani, J.-M. Morel, P. Musé, and F. Sur, *A Theory of Shape Identification*, Springer Verlag, 2008.

[14] J. M. Morel and G. Yu, "ASIFT: A new framework for fully affine invariant image comparison," *SIIMS*, vol. 2, no. 2, pp. 438–469, 2009.

[15] M. Rodriguez, J. Delon, and J.-M. Morel, "Covering the space of tilts. application to affine invariant image comparison," *SIIMS*, vol. 11, no. 2, pp. 1230–1267, 2018.

[16] R. Arandjelovic and A. Zisserman, "Three things everyone should know to improve object retrieval," in *CVPR*, 2012, pp. 2911–2918.

[17] G. Yu and J.-M. Morel, "ASIFT: An Algorithm for Fully Affine Invariant Comparison," *IPOL*, vol. 1, pp. 1–28, 2011.

[18] Y. Pang, W. Li, Y. Yuan, and J. Pan, "Fully affine invariant SURF for image matching," *Neurocomputing*, vol. 85, pp. 6–10, 2012.

[19] D. Mishkin, J. Matas, and M. Perdoch, "MODS: Fast and robust method for two-view matching.," *CVIU*, vol. 141, pp. 81–93, 2015.

[20] M. Rodriguez and R. Grompone von Gioi, "Affine invariant image comparison under repetitive structures," in *ICIP*, Oct 2018, pp. 1203–1207.

[21] M. Rodriguez, J. Delon, and J.-M. Morel, "Fast affine invariant image matching," *IPOL*, vol. 8, pp. 251–281, 2018.

[22] S. Zagoruyko and N. Komodakis, "Learning to compare image patches via convolutional neural networks," in *CVPR*, 2015, pp. 4353–4361.

[23] J. Zbontar and Y. LeCun, "Stereo matching by training a convolutional neural network to compare image patches," *JMLR*, vol. 17, no. 1-32, pp. 2, 2016.

[24] I. Rocco, R. Arandjelovic, and J. Sivic, "Convolutional neural network architecture for geometric matching," *TPAMI*, 2018.

[25] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Deep image homography estimation," *arXiv preprint arXiv:1606.03798*, 2016.

[26] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[27] I. Rey-Otero and M. Delbracio, "Anatomy of the SIFT method," *IPOL*, vol. 4, pp. 370–396, 2014.

[28] J. M. Morel and G. Yu, "Is SIFT scale invariant?," *Inv. Problems and Imaging*, vol. 5, no. 1, pp. 115–136, 2011.

[29] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *ECCV*. Springer, 2014, pp. 740–755.

[30] L. Moisan, P. Moulon, and P. Monasse, "Automatic Homographic Registration of a Pair of Images, with A Contrario Elimination of Outliers," *IPOL*, vol. 2, pp. 56–73, 2012.