BAFT: BINARY AFFINE FEATURE TRANSFORM

Jonas T. Arnfred, Viet Dung Nguyen, and Stefan Winkler

Advanced Digital Sciences Center (ADSC), University of Illinois at Urbana-Champaign, Singapore

ABSTRACT

We introduce BAFT, a fast binary and quasi affine invariant local image feature. It combines the affine invariance of Harris Affine feature descriptors with the speed of binary descriptors such as BRISK and ORB. BAFT derives its speed and precision from sampling local image patches in a pattern that depends on the second moment matrix of the same image patch. This approach results in a fast but discriminative descriptor, especially for image pairs with large perspective changes.

Our evaluation on 40 different image pairs shows that BAFT increases the area under the precision/recall curve (AUC) compared to traditional descriptors for the majority of image pairs. In addition we show that this improvement comes with a very low performance penalty compared to the similar ORB descriptor. The BAFT source code is available for download.

1. INTRODUCTION

The usefulness of local image features has been demonstrated by their diverse application in various computer vision domains ranging from object recognition [1] over scene alignment [2] to 3D structure from motion [3]. The features derive their usefulness through a balance of discriminative description of local image regions coupled with partial invariance to affine and photometric image transformations [4].

For applications that require real-time matching or run on mobile devices with limited resources, the computational efficiency of finding local keypoints and computing corresponding descriptors is crucial [5]. However, many existing fast descriptors trade off invariance with speed. We therefore propose the Binary Affine Feature Transform (BAFT), a fast binary descriptor robust to perspective changes like affine transformations.

In the design of BAFT we were inspired by earlier local image features. The closest kin of BAFT is the ORB keypoint detector and descriptor [6]. ORB introduced a simple but fast scale- and rotation-invariant local feature by combining the FAST keypoint detector [7] with the binary descriptor of BRIEF [8]. To obtain rotation- and scale-invariance, ORB applies the FAST keypoint detector to an image pyramid and estimates the orientation using intensity centroids after ranking and filtering the keypoints by the Harris Corner Measure [9], which we describe later in Section 2. The key to ORB's speed however is the simplistic descriptor sampling taken from BRIEF.

Another influence on BAFT are the Harris Affine and Hessian Affine detectors [10], which build on the theoretical foundations for affine invariant descriptors [11, 12]. The Harris Affine detector provides affine invariance by normalising the image region around each keypoint with respect to its second moment matrix in an iterative process, which we also describe in Section 2.

BAFT combines the two approaches by adapting the sampling of the binary descriptor to the normalisation matrix computed from the second moment matrix. Instead of iteratively refining this matrix, we compute it once and normalise the sampling pattern of points accordingly. The speed of BAFT is partially derived from the fact that the second moment matrix is already computed in order to weight the FAST keypoints, and as such little extra computation is necessary to create a skew- and stretch-invariant binary feature descriptor.

A few other works deserve mention before we meet them again in later comparisons. Lowe's SIFT descriptor uses a Difference-of-Gaussians on an image pyramid to find keypoints and a gradient histogram of the surrounding region to describe them. AKAZE [13] is an example of a binary descriptor showing impressive performance by using a non linear scale space for keypoint detection and description. Finally, ASIFT [14] is a hybrid of a feature descriptor and a matching algorithm. The image pairs are transformed to cover a set of affine transformations and for each transformation a set of SIFT descriptors is calculated and matched using RANSAC [15] to weed out false correspondences. While ASIFT is not a traditional feature descriptor, it is an interesting approach to matching images with large affine variations.

This study was supported by the research grant for the Human-Centered Cyber-physical Systems Programme at the Advanced Digital Sciences Center (ADSC) from Singapore's Agency for Science, Technology and Research (A*STAR).

Send correspondence to stefan.winkler@adsc.com.sg.

2. AFFINE INVARIANT FEATURE

Like ORB, BAFT[1] detects features using the FAST keypoint detector on an image pyramid. For each FAST keypoint we compute the second moment matrix M of a local image region defined as follows:

$$\mathbf{M}(\mathbf{x_r}) = \sum_{p,q} w(p,q) \begin{bmatrix} I_x^2(\mathbf{x_r}) & I_x I_y(\mathbf{x_r}) \\ I_x I_y(\mathbf{x_r}) & I_y^2(\mathbf{x_r}) \end{bmatrix}. \quad (1)$$

Here $w(p,q)$ is the weight of the pixel position (p,q) in the image region $\mathbf{x_r}$. I_x and I_y are the image derivatives in x and y directions. The weighting function $w(p,q)$ is often Gaussian, but for BAFT we use a uniform weight over a square region. The eigenvalues of $\mathbf{M}$ are a good indicator of the 'cornerness' of the image patch. $\mathbf{M}$ will have two large positive eigenvalues when the gradients I_x^2 and I_y^2 are both large, which is typically the case for a corner. Harris and Stephens [9] suggest using $m_c = \det(\mathbf{M}) - \alpha \operatorname{trace}^2(\mathbf{M})$ as a measure for evaluating the 'cornerness' of a keypoint. Using this measure we order the keypoints returned by FAST and pick out the N best. We calculate the second moment matrix using a square region of 12×12 pixels, but use only the central 2×2 pixel region for the purpose of evaluating the cornerness. The second moment matrix computed from the larger region is then stored in memory with the keypoint to use when we build the descriptor.

We build the descriptor by sampling the image around each keypoint based on a set of points $\mathbf{P}$ relative to the keypoint position. For our purposes $\mathbf{P}$ can be seen as a $2 \times k$ matrix, where k is the number of points we are sampling. If two image regions $\mathbf{R_l}$ and $\mathbf{R_r}$ are related by an affine transformation then there exists a matrix $\mathbf{A}$ such that $\mathbf{R_r}$ sampled by $\mathbf{P}' = \mathbf{AP}$ is equal to $\mathbf{R_l}$ sampled by $\mathbf{P}$. We can decompose the affine transformation into a skew matrix $\mathbf{S}$ and a rotation matrix $\mathbf{R}$: $\mathbf{A = SRS}^{-1}$; as suggested by Lindeberg [11], we compute the skew matrix as the square root of the second moment matrix $\mathbf{S} = \mathbf{M}^{\frac{1}{2}}$. For the rotation matrix we use the normalised eigenvector $v_\lambda = (v_0\ v_1)^T$ corresponding to the largest eigenvalue as the direction of our feature point and let $\mathbf{R} = \left(\begin{smallmatrix} -v_1 & v_0 \\ v_0 & v_1 \end{smallmatrix}\right)$. To normalize the sample points based on an image region we use $\mathbf{P}_{\text{norm}} = s\mathbf{R}\mathbf{M}^{\frac{1}{2}}\mathbf{P}$ to sample the region where s is the scale of region. This process is illustrated in Figure 1.

Because of our choice of rotation matrix, BAFT is not fully rotation-invariant. Two examples illustrate this fact: For the first example consider an image patch with a uniform gradient of $\mathbf{v} = [q\ \ -p]$. This patch would have a second moment matrix of $\mathbf{M} = \alpha\left(\begin{smallmatrix} q^2 & -qp \\ -qp & (-p)^2 \end{smallmatrix}\right)$ where α is a constant. This matrix would be identical to that of an image patch with the inverse gradient of $\mathbf{v} = [-q\ p]$. As a result, BAFT is ill suited to handle large rotations; for smaller rotations, using the eigenvector adds stability. For the second example consider an image patch with the two dominant gradients of $\mathbf{v}_1 = [q\ -p]$ and $\mathbf{v}_2 = [q\ p]$. The somewhat simplistic second moment matrix constituted of only those two gradients can be calculated as $\mathbf{M} = \alpha\left(\begin{smallmatrix} 4q^2 & 0 \\ 0 & 4p^2 \end{smallmatrix}\right)$. $\mathbf{M}$ yields a stable eigenvector with a direction between the two gradients, instead of deciding which gradient is dominant.

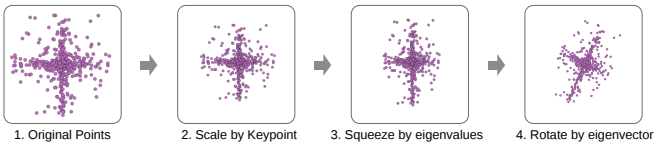

Fig. 1: Affine adjusted sampling of the descriptor. For each keypoint we sample the surrounding image based on the eigenvectors and values of the Harris response. 1. The original distribution of points. 2. Points scaled according to keypoint scale information. 3. Points squeezed based on the eigenvalues of the Harris response. 4. Points aligned with the eigenvector of the Harris response corresponding to the largest eigenvalue.

Given the sampled image values we use the output of *winner-take-all* hashing (WTA) as our descriptor. We group the samples in sets of four and find the highest and lowest values for each group. The resulting hash is made by concatenating the binary representation of the index of the highest and lowest values of a given set. Depending on how many points we sample, we can create descriptors of different length. The 16 byte version of BAFT is made from sampling 128 points in 32 groups with four sampling points each (each group contributes 2×2 bits). Similarly the 32 byte version samples 256 points in 64 groups, etc.

3. EXPERIMENTAL RESULTS

We evaluate all descriptors on the ASIFT dataset [14], which focuses on the challenge of perspective change and contains five sets of 10 images. We use the standard implementation of the OpenCV library version 3.0.0 for SIFT, ORB, and AKAZE. For Harris Affine and Hessian Affine we make use of the implementation provided by Mikolajczyk et al. [16].[2] For ASIFT we use the implementation by Morel et al. [14].[3]

3.1. Evaluating Correspondences

We compare descriptors by matching a set of image pairs using nearest neighbor ratio match. Each descriptor is matched with its nearest neighbor and ranked by the ratio r between the best and second best match. The ratio serves as an imperfect

[1] The C++ source code for BAFT is available for download at https://github.com/arnfred/BAFT.

[2] http://www.robots.ox.ac.uk/~vgg/research/affine/descriptors.html

[3] http://www.ipol.im/pub/art/2011/my-asift/

measure of how likely a given match is correct. For applications where high accuracy is needed we can select a subset of matches for which the ratio is less than a given threshold τ.

We calculate recall and precision for the subset of matches with $r \leq \tau$. The area under the curve (AUC) over τ for a set of matches can be calculated as the integral over the recall/precision curve. In practice we estimate this number using Monte Carlo integration.

All transformations in the two datasets are planar. The points in each image pair (I_1, I_2) are related by a homography $\mathbf{H}$. We calculate the projection error as:

$$e_P = |\mathbf{H}\mathbf{p_1} - \mathbf{p_2}| + \left|\mathbf{H}^{-1}\mathbf{p_2} - \mathbf{p_1}\right|. \quad (2)$$

A match between two points $(\mathbf{p_1}, \mathbf{p_2})$ is deemed as a correct correspondence if the projection error is less than $e_{\max}$; for our experiments we use $e_{\max} = 5$. For an image pair (I_1, I_2), we find the positions of all keypoints $(\mathcal{P}_1, \mathcal{P}_2)$ and calculate the total number of possible correspondences by counting all the pairs of keypoints for which $e_P \leq e_{\max}$.

3.2. Speed

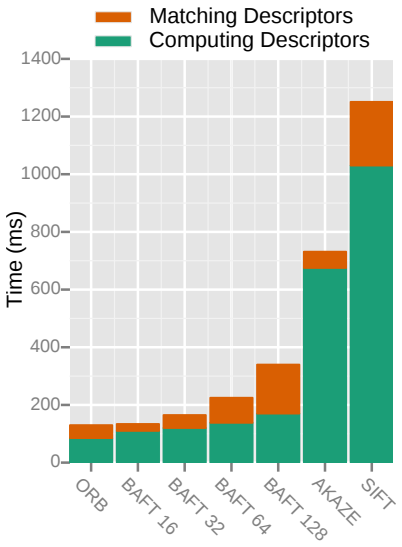

Fig. 2: Comparison of feature computation and matching speed for different descriptors.

Figure 2 compares BAFT with SIFT, ORB, AKAZE in terms of speed. For each descriptor we evaluate how long it takes to compute and match a thousand features. All measurements are done on a Intel Core2 Duo CPU @ 2.26GHz. The results demonstrate that we do not need to compromise on performance to obtain high robustness to perspective change. To find and compute descriptors for a thousand feature points, the 16-byte version of BAFT achieves speeds within 5 percent of ORB, partly due to the higher speeds of matching a 16-byte descriptor. The 32-byte version of BAFT is only 27% slower than ORB, which features the same descriptor length. Even the longest (128-byte) version of BAFT is less than 100 milliseconds slower than ORB. BAFT-128 is four times as fast as AKAZE and six times faster than SIFT for the task of computing descriptors. The other descriptors (ASIFT, Harris-Affine, Hessian-Affine) are much (1-2 orders of magnitude) slower than BAFT and thus not shown in this plot.

3.3. Accuracy

Figure 3 shows precision/recall curves for a selection of image pairs (cf. Figure 4) from the ASIFT Dataset. Except for Abs_x4-7, even the 16 byte version of BAFT has higher precision than SIFT.

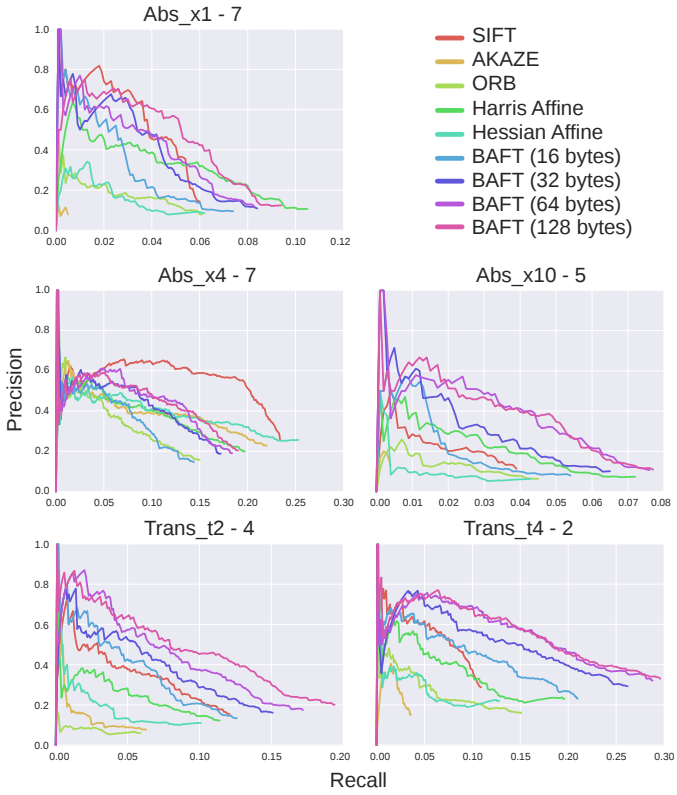

Fig. 3: BAFT compared with other feature descriptors on image pairs from the ASIFT dataset.

The complete AUC results for all image pairs are presented in Table 1. Cumulatively over all images in the ASIFT dataset, BAFT-16 outperforms the only faster tested descriptor ORB by more than 40%. BAFT-32 is superior to its closest contenders SIFT and AKAZE; BAFT-128 achieves about 80% higher AUC than SIFT.

Invariance invariably incurs a penalty for cases with little or no variation along the vector of invariability. A scale invariant detector for example is at a disadvantage when matching image pairs with no scale change. BAFT makes a tradeoff between a higher degree of skew invariance and a lower

Fig. 4: The image pairs used in Figure 3. From left to right the image sets with particular images noted in parenthesis are 'abs_x1' (1-7), 'abs_x4' (1-7), 'abs_x10' (1-5), 'trans_t2' (1-4), 'trans_t4' (1-2).

invariance to rotation in order to offset this penalty to some degree.

Figure 5 quantifies the rotation invariance of BAFT with respect to SIFT, ORB, and AKAZE. For each descriptor we compute the matching score for all images when matching with the same image rotated by a given angle (from -180 to $+180$ degrees). The results shown are averaged over all images as well as positive and negative angles. While BAFT cannot compete with SIFT or AKAZE for larger rotations, all but the 16-byte versions of BAFT fare similar to ORB up to about 90 degrees of rotation; for rotations of less than 45 degrees, they even approach SIFT and AKAZE.

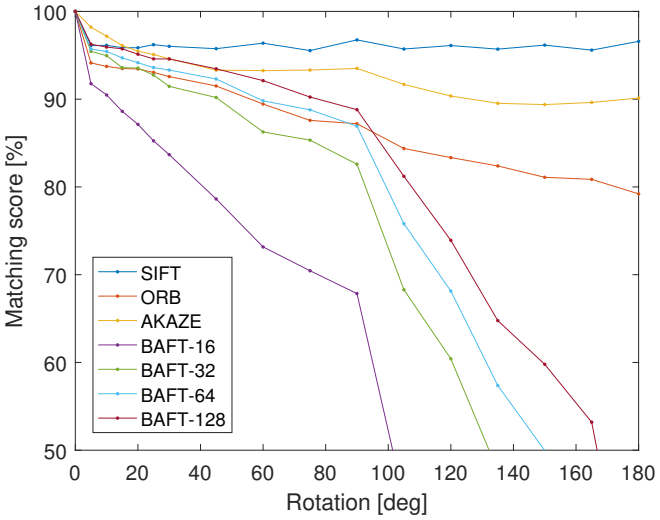

Fig. 5: Quantifying rotation invariance for different descriptors, when matching an image with a copy of itself rotated by a given angle.

Set	Pair	SIFT	ORB	AKAZE	Harris-Affine	Hessian-Affine	BAFT-16	BAFT-32	BAFT-64	BAFT-128
ABS X1	1—2	0.320	0.364	0.056	**0.479**	0.335	0.275	0.315	0.320	0.332
	1—3	0.174	0.279	0.051	**0.424**	0.249	0.264	0.301	0.317	0.332
	1—4	0.232	0.213	0.032	**0.278**	0.204	0.177	0.211	0.216	0.229
	1—5	0.135	0.159	0.015	**0.251**	0.109	0.123	0.156	0.161	0.172
	1—6	**0.120**	0.061	0.004	0.075	0.043	0.061	0.088	0.094	0.102
	1—7	0.094	0.030	0.001	0.091	0.029	0.067	**0.120**	**0.134**	**0.137**
	1—8	0.011	0.000	0.000	0.005	0.002	0.011	**0.017**	**0.016**	**0.019**
	1—9	0.013	0.005	0.000	0.009	0.001	**0.015**	**0.052**	**0.056**	**0.070**
ABS X4	1—2	0.686	0.703	**0.876**	0.646	0.729	0.587	0.656	0.675	0.697
	1—3	0.710	0.641	**0.877**	0.607	0.717	0.571	0.615	0.640	0.657
	1—4	0.646	0.394	**0.652**	0.438	0.506	0.385	0.436	0.452	0.461
	1—5	0.461	0.282	**0.479**	0.354	0.403	0.290	0.356	0.374	0.397
	1—6	**0.214**	0.082	0.134	0.122	0.154	0.080	0.096	0.103	0.109
	1—7	**0.018**	0.002	0.009	0.011	0.008	0.008	0.009	0.014	0.014
	1—8	**0.001**	0.000	0.000	**0.001**	0.000	0.000	0.000	0.000	0.000
	1—9	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
ABS X10	1—2	0.245	0.377	0.051	0.439	0.231	0.344	0.403	0.426	**0.448**
	1—3	0.155	0.316	0.047	0.337	0.182	0.258	0.337	**0.360**	**0.387**
	1—4	0.059	0.040	0.000	0.082	0.038	0.057	**0.086**	**0.093**	**0.109**
	1—5	0.026	0.017	0.002	0.043	0.012	**0.049**	**0.082**	**0.086**	**0.098**
	1—6	0.000	0.000	0.000	0.000	0.001	**0.004**	**0.011**	**0.009**	**0.013**
	1—7	0.002	0.001	0.000	0.016	0.001	**0.020**	**0.022**	**0.029**	**0.045**
	1—8	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	**0.001**
	1—9	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
TRANS T2	1—2	0.517	0.526	**0.620**	0.526	0.600	0.353	0.445	0.483	0.510
	1—3	0.229	0.106	0.144	0.215	0.226	0.172	**0.235**	**0.271**	**0.305**
	1—4	0.065	0.007	0.015	0.043	0.028	0.062	**0.094**	**0.135**	**0.146**
	1—5	0.003	0.001	0.003	0.012	0.007	0.008	**0.025**	**0.033**	**0.036**
	1—6	0.001	0.000	0.000	0.001	0.000	**0.007**	**0.011**	**0.016**	**0.016**
	1—7	0.000	0.000	0.000	0.000	0.000	**0.001**	**0.004**	**0.005**	**0.007**
	1—8	0.000	0.000	0.000	0.000	0.000	**0.001**	**0.001**	**0.006**	**0.005**
	1—9	0.000	0.000	0.000	0.000	0.000	**0.001**	**0.002**	**0.003**	**0.003**
TRANS T4	1—2	0.077	0.051	0.012	0.092	0.047	**0.108**	**0.164**	**0.182**	**0.205**
	1—3	0.002	0.002	0.002	0.009	0.002	**0.013**	**0.021**	**0.026**	**0.030**
	1—4	0.001	0.000	0.000	0.000	0.000	**0.002**	0.001	**0.003**	**0.004**
	1—5	0.000	0.000	0.000	0.000	0.000	0.000	0.000	**0.001**	**0.002**
	1—6	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	1—7	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	1—8	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	1—9	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
All	All	0.403	0.266	0.190	0.427	0.288	0.374	**0.530**	**0.641**	**0.746**

Table 1: AUC for image sets from the ASIFT dataset.

4. CONCLUSIONS

We presented BAFT, a local image feature descriptor using the second moment matrix of an image patch to adapt the sampling pattern and produce a skew- and stretch-affine descriptor constructed with a winner-take-all hashing strategy. We showed that BAFT can be computed efficiently, similar in speed to ORB, and several times faster than AKAZE or SIFT.

We compared BAFT with five other descriptors over 40 image pairs from the ASIFT dataset, which features large perspective changes. The results show that BAFT is on average twice as performant as ORB in terms of AUC for images, and remains superior to more invariant feature descriptors like SIFT and AKAZE. Based on these results we conclude that BAFT is a useful novel image feature that achieves robustness to perspective change without sacrificing speed.

5. REFERENCES

[1] Josef Sivic and Andrew Zisserman, "Video google: Efficient visual search of videos," in *Toward Category-Level Object Recognition*, Jean Ponce, Martial Hebert, Cordelia Schmid, and Andrew Zisserman, Eds., pp. 127–144. Springer, 2006.

[2] Ce Liu, Jenny Yuen, and Antonio Torralba, "SIFT flow: Dense correspondence across scenes and its applications," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 978–994, 2011.

[3] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M Seitz, and Richard Szeliski, "Building Rome in a day," *Communications of the ACM*, vol. 54, no. 10, pp. 105–112, 2011.

[4] Jonas T Arnfred and Stefan Winkler, "A general framework for image feature matching without geometric constraints," *Pattern Recognition Letters*, vol. 73, pp. 26–32, 2016.

[5] Jonas T Arnfred and Stefan Winkler, "Fast-Match: Fast and robust feature matching on large images," in *Proc. IEEE International Conference on Image Processing (ICIP)*, 2015.

[6] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. International Conference on Computer Vision (ICCV)*. IEEE, 2011, pp. 2564–2571.

[7] Edward Rosten and Tom Drummond, "Machine learning for high-speed corner detection," in *Proc. European Conference on Computer Vision (ECCV)*, 2006, pp. 430–443.

[8] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua, "BRIEF: Binary robust independent elementary features," in *Proc. European Conference on Computer Vision (ECCV)*, 2010, pp. 778–792.

[9] Chris Harris and Mike Stephens, "A combined corner and edge detector," in *Proc. 4th Alvey Vision Conference*, Manchester, UK, 1988, vol. 15, pp. 147–151.

[10] Krystian Mikolajczyk and Cordelia Schmid, "Scale & affine invariant interest point detectors," *International Journal of Computer Vision*, vol. 60, no. 1, pp. 63–86, 2004.

[11] Tony Lindeberg, "Feature detection with automatic scale selection," *International Journal of Computer Vision*, vol. 30, no. 2, pp. 79–116, 1998.

[12] Adam Baumberg, "Reliable feature matching across widely separated views," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2000, vol. 1, pp. 774–781.

[13] Pablo F Alcantarilla, Jesús Nuevo, and Adrien Bartoli, "Fast explicit diffusion for accelerated features in nonlinear scale spaces," in *Proc. British Machine Vision Conference (BMVC)*, 2013.

[14] Jean-Michel Morel and Guoshen Yu, "ASIFT: A new framework for fully affine invariant image comparison," *SIAM Journal on Imaging Sciences*, vol. 2, no. 2, pp. 438–469, 2009.

[15] Martin A Fischler and Robert C Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[16] Krystian Mikolajczyk and Cordelia Schmid, "A performance evaluation of local descriptors," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, 2005.