Match or No Match: Keypoint Filtering based on Matching Probability

Alexandra I. Papadaki
Technical University Berlin
Computer Vision & Remote Sensing Department
al.i.papadaki@gmail.com

Ronny Hänsch
German Aerospace Center (DLR)
Department SAR Technology
ronny.haensch@dlr.de

Abstract

Keypoints that do not meet the needs of a given application are a very common accuracy and efficiency bottleneck in many computer vision tasks, including keypoint matching and 3D reconstruction. Many computer vision and machine learning methods have dealt with this issue, trying to improve keypoint detection or the matching process. We introduce an algorithm that filters detected keypoints before the matching is even attempted, by predicting the probability of each point to be successfully matched. This is realised using a flexible and time efficient Random Forest classifier. Experiments on stereo and multi-view datasets of building facades show that the proposed method decreases the computational cost of a subsequent keypoint matching and 3D reconstruction, by correctly filtering 50% of the points that wouldn't be matched while preserving 73% of the matchable keypoints. This enables a subsequent processing with minimal mismatches, provides reliable matches, and point clouds. The presented filtering leads to an improved 3D reconstruction of the scene, even in the hard case of repetitive patterns and vegetation.

1. Introduction

Keypoint matching is a basic operation in almost every computer vision application, including image registration, image retrieval, Structure from Motion (SfM) and Multi-View Stereo (MVS). The standard workflow of image matching starts with the detection and description of keypoints and continues with matching. The detection step searches for adequate and locally distinctive keypoints that can not be easily confused (uniqueness) and that could be easily detected in different images (reliability). The description step represents the detected points using a multi-dimensional vector (descriptor). Finally, for every descriptor in one image, the approximately nearest neighbour in every other overlapping image is searched for. This way, each pair of matched points represents a single point in the scene, which is projected onto the two corresponding images captured from different viewpoints.

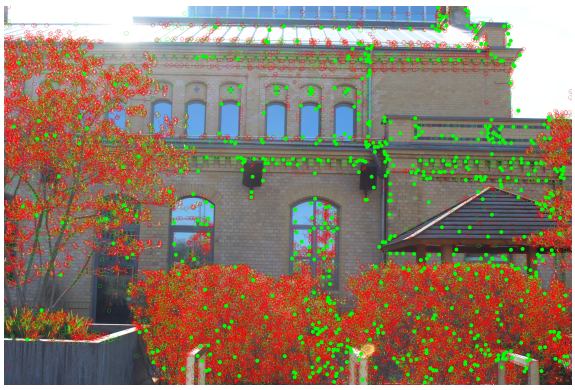

Figure 1. Initially detected keypoints (in red) and those predicted as matchable by the proposed method (in green). Note that most keypoints representing vegetation and repetitive structures are filtered out.

Regarding the computational cost and accuracy of keypoint matching, the main challenge is the large amount of detected keypoints, many of which being unreliable for matching, leading to confusion, false correspondences, and even failure of the reconstruction. The unreliability of the keypoints comes mostly from the varying representation of the same space point within the different images due to different environmental conditions, viewpoints and capturing distances, and equipment. Another reason for confusion is the nature, texture, and complexity of the depicted object itself, especially for vegetation (big amount of unstable, but highly distinctive keypoints) and repetitive patterns (hardly distinguishable keypoints).

We propose an algorithm to improve both, the accuracy and time efficiency of keypoint matching and thus of a complete 3D reconstruction of a scene. We present an alternative approach that decreases the amount of keypoints before matching by filtering the detected keypoints that are unlikely to be matched, as shown in Figure 1. The main contribution is to state the problem as a binary classification task, i.e. classifying a keypoint either as being able to

be matched or not. The developed method proposes a complete classification procedure, suggesting pioneering features for the training of the applied classifier. The algorithm requires only a single input image on which it detects and then describes the keypoints, using widely applied detection and description algorithms. Afterwards and before the keypoint matching, comes its main contribution. In this step, a keypoint classification is performed to predict and preserve the matchable keypoints. The conducted experiments show that the proposed method does reduce the computation cost and increases the accuracy of the example application, i.e. image-based 3D reconstruction.

2. Related Work

Reducing keypoints to improve the accuracy and computational cost of keypoint matching and also any subsequent image processing has seen considerable research efforts in the last years. However, most of the proposed approaches reduce the amount of keypoints independent of the characteristics of the depicted scene.

Keypoint detectors usually detect keypoints by computing some kind of score on the image points. Hand-crafted detectors (Moravec [23], Förstner [12], Harris [15] and Difference-of-Gaussians (DoG) [20]) tend to detect vast amounts of keypoints, especially on error prone areas like vegetation and repetitive patterns. Recently developed detectors suggest the use of machine learning and introduce new keypoint scores, to reduce the amount of detected keypoints. Such detectors have dealt with detection either as a classification problem (e.g. Task (Task Specific Keypoint detector, [33]) uses a WaldBoost classifier for this task) or as a regression problem (e.g. TILDE (Temporary Invariant Learned Detector, [35])). The very recent work of Barroso-Laguna *et al*. [4] combines hand-crafted and learned CNN filters in a shallow multi-scale architecture. Sample detection scores that have been proposed lately are the repeatability of the keypoints on textured images [37] and succinctness [10], meaning the points need to obtain certain inliers given a detector and a matching algorithm.

In contrast to these methods, the algorithm that is introduced here does not provide a new detector. It suggests to use SIFT [19] because it provides very satisfying results, but it could be easily adapted to other detectors, as long as they provide the information needed for the used features. Similar approaches to the one introduced here suggest also new application-related evaluation criteria (scores). Such a score is keypoint saliency, i.e. a function of the detectability, distinctiveness and repeatability of the points, even under various conditions [8] and the use of points on texture maps [25]. Another powerful criterion is the confusion risk [28, 29], which addresses the problem of mismatches due to repetitive patterns in the scene.

Other approaches that cast keypoint reduction as a classification problem include for example methods based on multi-layer perceptrons (e.g. [9] that exploits keypoint distinctiveness and robustness) as well as Random Forest (RF) classifiers (e.g. in [16]). The latter is probably the most relevant approach to the one presented in this work. Similar to us, they use a RF classifier to predict which SIFT keypoints would probably not survive the matching. Their approach was shown to be very time and precision efficient even in cases of viewpoint changes or images containing highly confusing gradients, like vegetation. The main differences to our approach are the model architecture and the features that are fed to the RF. Hartmann *et al*. used the 128 elements of the SIFT descriptor while our work is based on simple keypoint properties that can be possibly calculated for different detectors, making it more widely applicable. A much faster and more flexible RF of five trees with depth five (compared to 25 trees of depth 25 in [16]) is sufficient to reach similar performance.

In the context of obtaining faster and more reliable matches, several approaches have been developed for reducing image pairs [13], improving descriptors [5, 24, 17, 26, 3, 36, 14, 27, 11, 1] and filtering matches [22, 6, 2, 32, 21] - most notably the work in [18] which casts the matching itself as classification task that is solved with a RF.

3. Methodology

The proposed method - illustrated in Figure 2 - aims to improve accuracy and decrease the computational cost of keypoint matching by filtering out keypoints prior to the matching or any further image processing. This filtering is realized through a two-class classification (matchable vs. non-matchable keypoints) using a RF classifier. While this is similar to Hartmann *et al*. [16], who trained a classifier on the keypoint descriptor, we train the classifier on several distinct keypoint characteristics that go beyond mere appearance.

The training of the classifier requires adequate, representative, and reliable training data. These are obtained off-line using a typical SIFT-based SfM pipeline. Keypoint matching is based on the L2 distance between SIFT descriptors followed by a ratio test [20], a symmetry cross check and outlier removal via RANSAC. We focussed on a typical 3D reconstruction task within urban environments, i.e. of building facades. Thus, mismatches are mostly caused by different image capture conditions, vegetation, and repetitive patterns. The surviving keypoints serve as positive examples, i.e. instances of the matchable class, while the filtered keypoints are used as negative examples, i.e. instances of the non-matchable class.

We selected eight features that on the one hand capture geometrical, textural, topological, and appearance properties but are on the other hand also easy and fast to be computed. The latter is of importance as one goal of the pro-

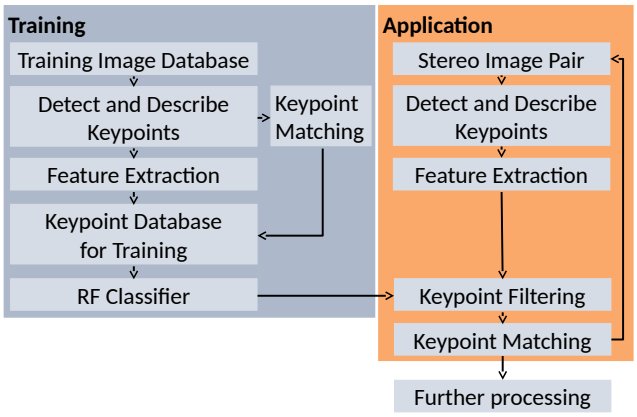

Figure 2. Overall workflow of the proposed algorithm.

posed method is to decrease the computational load which would be hindered by computationally expensive features.

The x- and y-coordinates: The position of a keypoint in an image of a typical sequence often correlates with its value for the matching process, i.e. image borders tend to have a higher probability to overlap with other images than the image centre.

Size s, orientation θ, response r, and octave o as computed by the keypoint detector (i.e. SIFT in our case): The size indicates the neighbourhood used for the description of each keypoint. The response is the score computed by the detector to decide whether a certain image coordinate does depict a keypoint. The octave of the scale space is the pyramid layer at which the keypoint was detected.

The number of dominant orientations (do) as computed by SIFT: This can be interpreted as a measure of textural complexity as well as of reliability of the computed keypoint orientation.

Intensity of the green channel g: Vegetation is often problematic in keypoint matching as it leads to many detections which can be hardly matched due to the temporal decorrelation of the corresponding 3D position and high occlusion.

In principle, more sophisticated features are possible as well. As an example we tested the in-image-similarity of a keypoint, i.e. the minimum distance of a keypoint to all other keypoints within the same image. However, including this feature did not significantly increase accuracy. On the contrary it changed the complexity of the algorithm a lot.

The above stated features are used as input to a RF classifier. We selected a RF due to its efficiency, robustness, and generality regarding used features (e.g. it does not require preprocessing of the features such as whitening). The RF is trained on the obtained samples and features to distinguish between keypoints that were successfully matched by the used SfM pipeline and those that could not be matched. If successful, the trained RF can then be used to pre-filter detected keypoints in a new image based on their predicted probability to be matchable. Consequently, matching would only be performed on a small fraction of all keypoints for which the corresponding matching probability is high.

4. Experiments and Evaluation

The introduced algorithm is tested and evaluated on various stereo and multi-view datasets. In the following we not only report a visual and statistical analysis of the keypoint classifier, but also analyse its actual impact on subsequent computer vision processes, as a good classification performance does not automatically imply good performance in subsequent matching or other processing steps. While Section 4.2 analyses the relevance of the individual features, Section 4.3 analyses the use of this RF to filter non-matchable keypoints and to provide matchable points to subsequent processes. Sections 4.4 and 4.5 illustrate two possible examples, namely matching of a stereo image pair and a 3D reconstruction via SfM.

We compare the proposed method to the standard procedure without any (non-standard) keypoint filtering, as well as to filtering using the features and RF hyperparameters as proposed by Hartmann *et al*. [16]. For the rest of the paper these methods will be called *Proposed-Filter*, *No-Filter* and *Filter-by [16]*, respectively. All experiments are performed using identical data, procedures (except for the setup of the corresponding filtering), and hardware.

4.1. Datasets

The used datasets are images depicting human structures, i.e. building facades. Such environments usually contain a lot of error prone areas like vegetation and repetitive patterns, that lead to confusion and mismatches.

We used 150 images for training, resulting in roughly 150,000 sample keypoints even after down-sampling. All images were captured with various cameras and have various resolutions. The total amount of test images is 456. Of these, 68 were taken from [34] and the rest were captured by a CANON 1300D or similar cameras. The images from [34] have a resolution of 3072×2048 pixels, while the rest have 5184×3456 pixels.

Apart from the resolution, these images were taken in different places and under different lighting conditions, capturing geometries and intrinsic parameters (different cameras). They all depict building facades, but encounter a great variety of different architectures and construction materials, which lead to different perspective distortions and textures. Moreover, problems that are usually met in urban environments like repetitive patterns, vegetation and pedestrians were not avoided while capturing, but instead included and successfully faced by the algorithm. The camera-object distance in all cases did not exceed roughly 50m and the baseline was roughly 20m.

4.2. Feature Importance

The features and structure of the used RF classifier are selected to provide reliable, robust, yet efficient results. As a first step, we analyse the relevance of the proposed features based on the permute importance (Mean Decreased Accuracy (MDA [7], i.e. shuffling the values of a single feature and observing the change of performance) and selection frequency (i.e. counting how often a certain feature dimension is selected by the nodes of the decision trees to perform a split).

	x	y	s	θ	r	o	do	g
MDA	7.77	5.12	21.33	2.37	7.83	3.14	19.13	33.31
SF	10.46	6.96	29.21	1.41	8.52	2.15	21.18	20.11

Table 1. Feature importance based on Mean Decreased Accuracy (MDA, [7]) and Selection Frequency (SF)

Table 1 shows that feature relevance is mostly consistent between both methods. Size s, color g, and the number of dominant orientations do are most important. Response r and spatial coordinates x, y have a medium importance, while octave o and orientation angle θ have low importance. As all of them contribute to the final performance (some only by a small margin but none is decreasing accuracy) and all are easy to compute, we included all features into the feature set used in the following experiments. The small number of such simple yet descriptive features allows a well performing, small, flexible and fast classifier, which doesn't require huge amounts of training data.

4.3. Keypoint Filtering

This section evaluates the classification performance of the designed RF classifier on the test data. We opted for a simple and compact architecture to avoid overfitting and keep the computational cost low, i.e. a Random Forest with five trees of depth five. The number of split candidates per node is set to the commonly used square root of the number of features (i.e. $\sqrt{\#features} = \sqrt{8} \approx 3$), the minimum amount of samples per node is set to two. Experiments to optimise these parameters have been performed on the training data.

Figures 1 and 3 show two example results. SIFT detects a large number of keypoints (depicted in red), especially on vegetation, repetitive patterns on the walls and windows frames. The proposed method filters mostly keypoints that lie on repetitive patterns, vegetation, moving objects (such as pedestrians), as well as on homogeneous and texture-less areas while preserving more stable keypoints (i.e. predicted as matchable and shown in green). Most of the keypoints predicted as matchable are actually successfully matched by the subsequent matching algorithm (depicted in green solid circles).

Table 2 provides a quantitative performance evaluation.

Figure 3. Image containing repetitive patterns, homogeneous areas and pedestrians. Initially detected keypoints are depicted in red, predicted as matchable in green, and the finally matched as filled circles.

%	Specificity	Recall	Precision	Accuracy	Sample reduction
Proposed Filter	50.36	73.12	15.51	52.89	48
Filter by [16]	51.65	71.72	15.60	53.88	49

Table 2. Statistical evaluation of the filtering performance for Proposed-Filter and Filter-by [16] methods.

Specificity and recall indicate the effectiveness of the classifier in identifying the negative and positive class respectively. Accuracy describes the overall correctly classified samples and precision the amount of correct true positive samples. In the specific application recall is the most useful measure in terms of the final ability of the algorithm to correctly preserve the matchable keypoints in the images. However, since the time efficiency is related to proper filtering of non-matchable keypoints, precision, accuracy and specificity should be as high as possible.

As there are usually much more keypoints detected than are successfully matched, the used test data is highly imbalanced, i.e. there is an order of magnitude less matchable samples than non-matchable samples. The proposed method leads to a recall that exceeds 73%, i.e. preserving most of the matchable keypoints. The specificity of roughly 50% means that half of the non-matchable keypoints are successfully filtered out. The low precision of less than 16% and medium accuracy are mainly caused by the unbalanced data and the survival of a lot of non-matchable keypoints. This however has no effect on the objectively good performance on preserving the matchable keypoints (high recall).

The last row of Table 2 presents the corresponding statistics for the Filter-by [16] method. Both filtering methods lead to statistically similar results. The Filter-by [16] method filtered out slightly more non-matchable keypoints (specificity 51.65% instead of 50.36%) leading

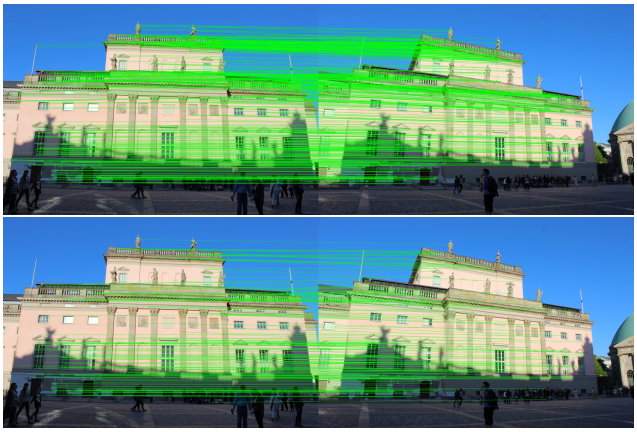

Figure 4. Matches without (top) and with (bottom) keypoint filtering.

to a marginally better keypoint reduction. This comes with the cost of losing a few more matchable keypoints (recall 71.71% instead of 73.12%). It should be noted that Filter-by [16] is based on a much larger RF of 25 trees of depth 25 as well as 128 features (the SIFT descriptor).

4.4. Impact on Pairwise Matching

The previous Section 4.3 showed that the proposed classifier is able to filter a large portion of non-matchable keypoints while preserving the majority of matchable keypoints. However, a good filtering does not guarantee a subsequent good matching. Since the classification is performed independently for every image, it is possible that essential matches are lost, if the corresponding points are not preserved in both images (i.e. note that correctly classifying a matchable keypoint pair has a chance probability of $p^2 = 53\%$ where $p = 73\%$ is the recall of Section 4.3).

In order to investigate how the proposed filtering affects the matching, several stereo-pairs are matched with and without pre-filtering using Colmap [30, 31]. The proposed method preserves up to 73% of the matches provided by the standard matching without filtering which is considerably larger than chance and shows that the prediction is stable. The loss of some matches is expected since the proposed method filters around 27% of matchable keypoints (see Section 4.3). Furthermore, even though the matches after filtering are fewer, they might still be more reliable, since matches on error prone areas are avoided. Figure 4 shows an example where matches are avoided that lie at the top of the building on repeated identical columns and can easily cause mismatches.

The computation time of the proposed method, the Filter-by [16] method, as well as applying no filter at all are reported in Table 3. We excluded the keypoint detection and description by SIFT as this is identical for all three methods. The first column reports the prediction time per image which is obviously zero if no prediction-based filtering is applied. For the proposed method this accumulates to only 0.2s, i.e. 10 times less time than the Filter-by [16] method. This significant performance improvement is achieved by using fast computable features and a small (in number of trees and maximal tree depth) RF model.

	Prediction time per image	Pairwise matching time
No Filter	0	16
Proposed Filter	0.2	11
Filter by [16]	2	10

Table 3. Prediction and matching time performance in seconds(s).

	Reprojection Error	Number of reconstructed points
No Filter	0.64	100570
Proposed Filter	0.55	107717
Filter by [16]	0.56	94501

Table 4. 3D Reconstruction performance.

By reducing the amount of keypoints by around 50%, both filtering methods accelerate the actual pairwise matching by approximately 32-37%. It should be noted that the time in the cases of pre-filtering include loading the corresponding keypoint files from disk which is not necessary in the case of no filtering. Consequently, if the filtering is incorporated into the SfM pipeline directly, this margin will even increase. The proposed method leads to a slightly slower pairwise matching time than filtering by [16] (11 instead of 10s), because it filters slightly less keypoints (48% instead of 49%). The overall image pair matching using the proposed filter is around 7% faster than using the filter by [16], but still slower than if no filtering is applied. It should be noted that the time for keypoint filtering grows linearly with the number of images, while the matching cost grows quadratically. It can thus be expected that for larger image collections the overhead in keypoint filtering is more than compensated by the smaller matching cost.

Summarising, the proposed method provides very stable matches. It preserves the majority of good matches while minimising mismatches by filtering out keypoints on error prone areas containing vegetation, repetitive patterns and moving objects. Finally, by reducing the number of keypoints it decreases the required pairwise comparisons and thus the time needed for keypoint matching.

4.5. Impact on 3D Reconstruction

This section evaluates the proposed method in an end-to-end SfM and 3D reconstruction pipeline. We use Colmap [30, 31] as well as several multi-view datasets of 21-98 images of building facades. These datasets have different geometries and characteristics of the scene, that facilitates the evaluation of the performance of the proposed filtering for different conditions.

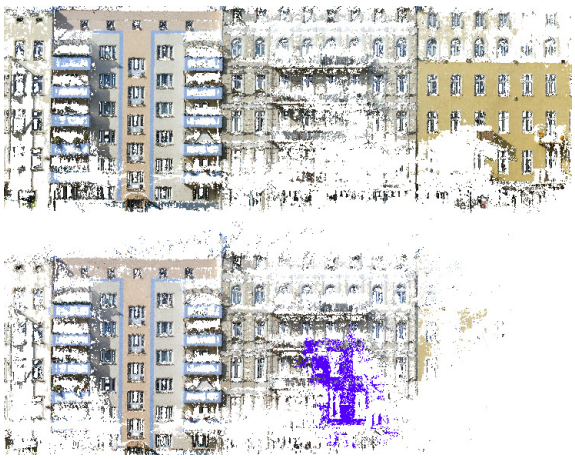

Figure 5. Dense point clouds after applying Proposed-Filter (top) and Filter-by [16] (bottom) methods. The purple points indicate wrongly reconstructed points.

The results reported in Table 4 show that the proposed method outperforms the compared methods regarding the final 3D reconstruction accuracy of the scene. Compared to SfM without filtering and SfM with filtering based on [16], it achieves 14% and 1.5% smaller reprojection error while having 7% and 14% more reconstructed points, respectively.

Figure 5 presents different point clouds of building facades produced by applying the two different filtering methods. The scene contains a large amount of repetitive pattern, e.g. identical and symmetric windows. In this example, the proposed method reconstructed 32% more points than filtering based on [16]. This large difference comes mainly (but not only) from the reconstruction of the right-most facade which is correctly reconstructed by the proposed method while large parts are either missing or mismatched with the center facade (marked in purple) if filtering is based on [16].

Figure 6 provides another characteristic example showing point clouds of the inner yard of a rectangular building (the lower part of the figure shows a horizontal slice to better illustrate the differences between the different methods). We construct a reference dense point cloud (shown in blue) based on 30 images using Agisoft's PhotoScan® software representing the whole building, preserving its rectangular geometry. An orthogonal corner of this building facade is reconstructed based on two images only using the proposed filter (shown in green), the Filter-by [16] (shown in magenta) and No-Filter (shown in orange if based on Colmap and red if based on VisualSFM). Obviously, none of the applied methods leads to very dense point clouds as only two images are used. However, the only point cloud that reconstructs the two facades as being orthogonal and coincides with the control point cloud is the one produced by using the proposed filtering.

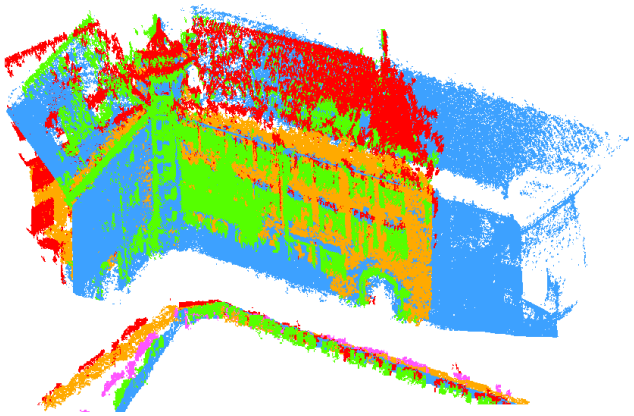

Figure 6. The 90° corner of the building is reconstructed correctly only by the Proposed-Filter SfM point cloud (green), which coincides with the reference one (blue). Point clouds produced by the Filter-by [16] (magenta) and No-Fiter SfM (orange if using Colmap and red if using VisualSFM) fail to preserve the geometry, providing much wider angles. The lower part of the figure shows a horizontal slice to better illustrate the differences between the different methods.

Summarising, the proposed keypoint filtering improves the 3D reconstruction significantly in particular in hard cases with repetitive patterns, vegetation, etc. It outperforms the reference method in terms of accuracy (reprojection error and SfM points), precision (reconstruction close to ground truth), time efficiency (fast filtering) and data prerequisites.

5. Conclusions

We suggest keypoint filtering prior to matching by casting it as a classification task and predicting the probability of a keypoint being part of a valid match. This is achieved by using a very small but time efficient RF classifier with only eight simple features and five trees with depth five. The method preserves 73% of the all matchable keypoints and filters 50% of the non-matchable keypoints especially those on error prone areas with vegetation, repetitive patterns, moving objects like pedestrians as well as homogeneous and texture-less areas. By reducing the keypoints, mismatches are avoided and pairwise matching is accelarated by more than 30%. The proposed filtering increases significantly the accuracy (lower reprojection error and more complete point cloud) and precision (true scene geometry) of a 3D reconstruction, compared to SfM without filtering and SfM with filtering according to [16].

References

[1] Hani Altwaijry, Andreas Veit, and Serge Belongie. Learning to detect and match keypoints with deep architectures.

In *British Machine Vision Conference (BMVC)*, York, UK, 2016.

[2] Sunil Arya, David M Mount, Nathan S Netanyahu, Ruth Silverman, and Angela Y Wu. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *Journal of the ACM (JACM)*, 45(6):891–923, 1998.

[3] Vassileios Balntas, Edgar Riba, Daniel Ponsa, and Krystian Mikolajczyk. Learning local feature descriptors with triplets and shallow convolutional neural networks. In *BMVC*, volume 1, page 3, 2016.

[4] Axel Barroso-Laguna, Edgar Riba, Daniel Ponsa, and Krystian Mikolajczyk. Key.Net: Keypoint Detection by Handcrafted and Learned CNN Filters. In *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision*, 2019.

[5] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006.

[6] Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, 1975.

[7] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[8] Simone Buoncompagni, Dario Maio, Davide Maltoni, and Serena Papi. Saliency-based keypoint selection for fast object detection and matching. *Pattern Recognition Letters*, 62:32–40, 2015.

[9] Gustavo Carneiro and Allan D Jepson. The quantitative characterization of the distinctiveness and robustness of local image descriptors. *Image and Vision Computing*, 27(8):1143–1156, 2009.

[10] Titus Cieslewski, Konstantinos G Derpanis, and Davide Scaramuzza. Sips: Succinct interest points from unsupervised inlierness probability learning. In *2019 International Conference on 3D Vision (3DV)*, pages 604–613. IEEE, 2019.

[11] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. 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*, pages 8092–8101, 2019.

[12] Wolfgang Förstner and Eberhard Gülch. A fast operator for detection and precise location of distinct points, corners and centres of circular features. In *Proc. ISPRS intercommission conference on fast processing of photogrammetric data*, pages 281–305, 1987.

[13] Jan-Michael Frahm, Pierre Fite-Georgel, David Gallup, Tim Johnson, Rahul Raguram, Changchang Wu, Yi-Hung Jen, Enrique Dunn, Brian Clipp, Svetlana Lazebnik, et al. Building rome on a cloudless day. In *European Conference on Computer Vision*, pages 368–381. Springer, 2010.

[14] Georgios Georgakis, Srikrishna Karanam, Ziyan Wu, Jan Ernst, and Jana Košecká. End-to-end learning of keypoint detector and descriptor for pose invariant 3d matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1965–1973, 2018.

[15] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, pages 10–5244. Citeseer, 1988.

[16] Wilfried Hartmann, Michal Havlena, and Konrad Schindler. Predicting matchability. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9–16, 2014.

[17] Yan Ke and Rahul Sukthankar. Pca-sift: A more distinctive representation for local image descriptors. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–II. IEEE, 2004.

[18] V. Lepetit and P. Fua. Keypoint recognition using randomized trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9):1465–1479, 2006.

[19] David G Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.

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

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

[22] Kwang Moo Yi, Eduard Trulls, Yuki Ono, Vincent Lepetit, Mathieu Salzmann, and Pascal Fua. Learning to find good correspondences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2666–2674, 2018.

[23] Hans P Moravec. Obstacle avoidance and navigation in the real world by a seeing robot rover. Technical report, STANFORD UNIV CA DEPT OF COMPUTER SCIENCE, 1980.

[24] Jean-Michel Morel and Guoshen Yu. Asift: A new framework for fully affine invariant image comparison. *SIAM journal on imaging sciences*, 2(2):438–469, 2009.

[25] Prerana Mukherjee, Siddharth Srivastava, and Brejesh Lall. Salient keypoint selection for object representation. *2016 Twenty Second National Conference on Communication (NCC)*, pages 1–6, 2016.

[26] Arun Mukundan, Giorgos Tolias, and Ondrej Chum. Multiple-kernel local-patch descriptor. *arXiv preprint arXiv:1707.07825*, 2017.

[27] Jerome Revaud, Philippe Weinzaepfel, César Roberto de Souza, and Martin Humenberger. R2D2: repeatable and reliable detector and descriptor. In *NeurIPS*, 2019.

[28] Emilien Royer, Thibault Lelore, and Frédéric Bouchara. Core: A confusion reduction algorithm for keypoints filtering. In *VISAPP (1)*, pages 561–568, 2015.

[29] Emilien Royer, Thibault Lelore, and Frédéric Bouchara. Confusion reduction (core) algorithm for local descriptors, floating-point and binary cases. *Computer Vision and Image Understanding*, 158:115–125, 2017.

[30] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[31] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016.

[32] Josef Sivic and Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. In *null*, page 1470. IEEE, 2003.

[33] Christoph Strecha, Albrecht Lindner, Karim Ali, and Pascal Fua. Training for task specific keypoint detection. In *Joint Pattern Recognition Symposium*, pages 151–160. Springer, 2009.

[34] Christoph Strecha, Wolfgang Von Hansen, Luc Van Gool, Pascal Fua, and Ulrich Thoennessen. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. Ieee, 2008.

[35] Yannick Verdie, Kwang Moo Yi, Pascal Fua, Vincent Lepetit, et al. Tilde: A temporally invariant learned detector. In *CVPR*, volume 2, page 5, 2015.

[36] Amir R Zamir, Tilman Wekel, Pulkit Agrawal, Colin Wei, Jitendra Malik, and Silvio Savarese. Generic 3d representation via pose estimation and matching. In *European Conference on Computer Vision*, pages 535–553. Springer, 2016.

[37] Linguang Zhang and Szymon Rusinkiewicz. Learning to detect features in texture images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.