IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 2, 2022, accepted February 13, 2022, date of publication February 17, 2022, date of current version March 2, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3152539

Dense Feature Matching Based on Homographic Decomposition

SIMON SEIBT[1,*], BARTOSZ VON RYMON LIPINSKI[1,*], AND MARC ERICH LATOSCHIK[2]
[1]Game Tech Laboratory, Faculty of Computer Science, Nuremberg Institute of Technology, 90489 Nuremberg, Germany
[2]Human-Computer Interaction Group, Institute of Computer Science, University of Wuerzburg, 97074 Wuerzburg, Germany

Corresponding author: Simon Seibt (simon.seibt@th-nuernberg.de)

Simon Seibt and Bartosz Von Rymon Lipinski contributed equally to this work.

ABSTRACT Finding robust and accurate feature matches is a fundamental problem in computer vision. However, incorrect correspondences and suboptimal matching accuracies lead to significant challenges for many real-world applications. In conventional feature matching, corresponding features in an image pair are greedily searched using their descriptor distance. The resulting matching set is then typically used as input for geometric model fitting methods to find an appropriate fundamental matrix and filter out incorrect matches. Unfortunately, this basic approach cannot solve all practical problems, such as fundamental matrix degeneration, matching ambiguities caused by repeated patterns and rejection of initially mismatched features without further reconsideration. In this paper we introduce a novel matching pipeline, which addresses all of the aforementioned challenges at once: First, we perform *iterative rematching* to give mismatched feature points a further chance for being considered in later processing steps. Thereby, we are searching for inliers that exhibit the same homographic transformation per iteration. The resulting *homographic decomposition* is used for refining matches, occlusion detection (e.g. due to parallaxes) and extrapolation of additional features in critical image areas. Furthermore, Delaunay triangulation of the matching set is utilized to minimize the repeated pattern problem and to implement *focused matching*. Doing so, enables us to further increase matching quality by concentrating on local image areas, defined by the triangular mesh. We present and discuss experimental results with multiple real-world matching datasets. Our contributions, besides improving matching recall and precision for image processing applications in general, also relate to use cases in image-based computer graphics.

INDEX TERMS Delaunay triangulation, Extrapolation, Feature Matching, Homography matrix, Repeated pattern matching.

I. INTRODUCTION

Feature matching between two color images is an essential step in many computer vision applications, such as image-based rendering, 3D reconstruction, object tracking, change detection, stitching, image registration and photo mosaicking [15], [22], [40], [49]. The conventional feature matching pipeline can usually be divided into the following four subprocesses: feature detection, feature description, preliminary feature matching and outlier removal. For the first two substeps, algorithms such as SIFT [30], SURF [7] or ORB [47] are mostly used.

The preliminary feature matching is based usually on a simple Euclidean distance comparison between the previously computed feature descriptors. The outlier removal is typically performed by using the RANSAC algorithm [18]. It is based on successive attempts to fit a model (usually the fundamental matrix) to a maximum subset of matched features, the so-called inliers. A correctly estimated fundamental matrix describes the geometric relationships between all corresponding image pixels [32].

Robustness and accuracy are crucial for most feature-based image processing applications in practice. In some use cases, particularly in context of image-based computer graphics, dense feature correspondence sets are also required. Examples include image morphing, warping, 3D reconstruction and photogrammetric modeling [13], [41], [52], [54], [61]. Such use cases are relying on many precise matches to reduce ghosting artifacts in interpolated intermediate views or to minimize 3D reconstruction error, for example.

The associate editor coordinating the review of this manuscript and approving it for publication was Yizhang Jiang.

However, these requirements represent a fundamental challenge in computer vision, as the conventional feature matching pipeline has some intrinsic limitations, described below:

The first challenge in conventional feature matching is the degeneration of the fundamental matrix, because its estimation is sensitive if computed for scenes with complex structures or multiple depth layers. A wrong fundamental matrix estimation can also be caused by the RANSAC algorithm itself: In its basic implementation, it typically selects only prominent feature points, which are concentrated just in a local pixel or depth area, especially in real scenes with high depth complexities. In such cases the resulting fundamental matrix is not representative, resulting in false rejection of further (potentially correct) feature correspondences [57]. Fig. 1 illustrates this problem: Even with perfect feature correspondences (a), the estimated fundamental matrix (b) deviates from the correct fundamental matrix (c), which was computed using actual camera parameters in this example. In addition, by using the 5-point [38] or 8-point algorithm [29], RANSAC uses only a small number of potential correspondences from the matching set to estimate the model. This is potentially suboptimal for image pairs with wide baselines, as these will contain a large percentage of outliers [64].

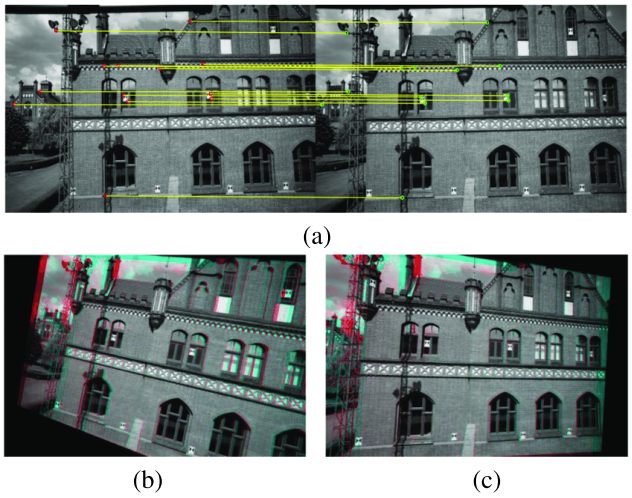


FIGURE 1. (a) Feature correspondences between two views; (b) rectification by the estimated fundamental matrix; (c) rectification by the computed fundamental matrix [37].

A further problem is that feature matching is intended to find image pair correspondences, which represent the same physical point. In conventional feature matching, however, a matched feature point in one image corresponds just to the nearest neighbor based on Euclidean distance comparisons in the other image. So, initial mismatches can be propagated as false positives in following application steps, which is known – in case of ambiguities – as the ''repeated pattern matching problem'' of computer vision [45].

In this paper we introduce a new feature matching method, which addresses all the aforementioned problems in one pipeline. The corresponding goal is to output more precise and denser feature-based correspondences: Our solution extends conventional feature matching to an *iterative rematching* process, allowing us to reconsider previously rejected feature points as potentially correct matches. Furthermore, instead of using one fundamental matrix, our search for correct feature correspondences is executed per iteration with an individually estimated homographic transformation. The result of the rematching process is set of homographies, we call *homographic decomposition*. Using different homographies, each associated with a specific matching area, provides the following advantages:

(a) matching feature points in the target image can be refined by using neighboring homographies from the source image to approximate their exact physical positions, (b) ''critical image areas'' (i.e. containing partially occluded objects) can be detected by combinatorial analysis of the homographic decomposition and considered in following pipeline steps, and (c), additional feature points, which cannot be detected by traditional matching, can be identified by using local homographies for *feature extrapolation*, especially in peripheral zones of critical image areas. Moreover, Delaunay triangulation of the feature point set makes it possible to utilize the resulting triangle mesh as a ''supporting structure'' to implement *Delaunay outlier detection*. On the one hand, this makes it possible to defuse the repeated pattern problem. On the other hand, it allows us to further increase feature point density. Here, we refer to *focused matching*, which incorporates the local re-execution of the matching pipeline within triangle cells that correspond in both images.

In our work we primarily target the above mentioned use cases from image-based computer graphics. Therefore, we focus on datasets with the following properties, common in this application context: (i) inside-out shots of real scenes, typically with high depth complexity, (ii) sequences of pairwise overlapping images from different viewpoints, (iii) representation of only static scenes with predominantly stable illumination situations, and (iv) without strong lens distortions, as in fisheye photography, for example.

We present related work in the following section. Then, section III contains a detailed description of our pipeline, followed by the evaluation and discussion of qualitative and quantitative results in section IV. Conclusions and future work are addressed in section V.

II. RELATED WORK

Since the 1980s, different algorithms have been developed for the detection and description of image features: For example, Harris and Stephens [20] published the Harris Corner Detector. The ''Scale Invariant Feature Transform'' (SIFT) algorithm was presented by Lowe [30]. Bay *et al.* [7] presented the ''Speeded Up Robust Features'' (SURF) algorithm. The ''Binary Robust Independent Elementary'' (BRIEF) descriptor was introduced by Calonder *et al.* [10]. In 2011, based on the ''Features from Accelerated Segment Test'' (FAST) detector [46] and the BRIEF descriptor, ''Oriented FAST''

and "Rotated BRIEF" (ORB) algorithms were introduced by Rublee *et al.* [47], respectively. Recent methods use also deep learning for feature detection and description: For example, "Learned Invariant Feature Transform" (LIFT) was proposed by Yi *et al.* [63]. It is a deep neural network that combines the components of standard pipelines for local feature detection and description into a single differentiable network, supervised by a common "structure from motion" process. DeTone *et al.* [14] published a self-supervised framework, called "Superpoint", for training interest point detectors and descriptors. Ono *et al.* [39] introduced another deep neural architecture, which trains a detector and descriptor end-to-end in a two branch setup. One branch is differentiable and is feeding on the output of the other non-differentiable branch. Lou *et al.* [31] published the "ASLFeat" learning framework for local features of accurate shape and localization. Truong *et al.* [58] introduced a CNN-based feature point detector for specific applications, like medical image matching. It is trained in a semi-supervised manner on pairs of images related by a homographic transformation.

Image feature descriptors and corresponding distance comparisons can only approximate relationships between physical features, which usually leads to a relatively high number of visual mismatches in practice. Therefore, "outlier removal" has become an important step in feature matching pipelines, typically by trying to approximate global geometric relationships between images. For example, Fischler and Bolles [18] presented the "random sample consensus" algorithm (RANSAC) to remove outliers during estimation of the fundamental matrix by considering epipolar geometry to filter out falsely corresponding feature points. Researchers have already developed a number of methods to improve the efficiency and robustness of the basic RANSAC algorithm, for example to solve the above-mentioned fundamental matrix degeneration problem and thus to obtain a better geometric model estimation. Examples, incorporating local optimization methods, are Chum *et al.* [12] and Frahm and Pollefeys [19], including "Inner RANSAC". Raguram *et al.* [42] implemented the "USAC", an universal framework for random sample consensus, which extends the simple hypothesize-and-verify structure of standard RANSAC and makes it possible to consider various optimizations. The work of Tan *et al.* [55] includes improvements for achieving a more uniform spatial distribution of feature correspondences and filtering mismatches using a smoothed disparity check based on a pre-estimated fundamental matrix. Other researchers proposed alternative model fitting algorithms. For example, Barath *et al.* [4] presented the MAGSAC algorithm that does not require a single inlier-outlier threshold such as RANSAC. By exploiting the residual density, Tiwari and Anand [56] introduced the DGSAC algorithm. In the work of Ranftl and Koltun [44] outliers are removed via geometric model estimation and the underlying fundamental matrix is computed using deep neural networks. More recently, Skoryukina *et al.* [51] proposed a RANSAC scheme with geometrical restrictors, focusing on ID document classification. For this case of planar object matching, improvements in accuracy are achieved.

Especially for image pairs with wide baselines, there is a a major drawback of outlier removal by basic fundamental matrix estimation: Corresponding algorithms typically rely only on small subsets of the data, required to generate the hypothesis. This can result in a high number of outliers [64]. Previous works try to address this limitation at the matching subprocess level. Therefore, they are related to our work, since they pursue the same goal of pruning false matches while finding a high number of robust and accurate correspondences: Ancuti et. al [2] use kernel feature correspondences to estimate geometric relationships between surrounding regions for the generation of additional positive matches. Bian *et al.* [9] reject outliers by converting motion smoothness constraints into statistical measures based on a limited number of feature matches between a region pair. Another related correspondence pruning method by Lin *et al.* [25] aims to detect a coherence-based separability constraint from noisy matches and embed it into a correspondence likelihood model. Exact matches are then obtained by varying the affine motion model. Ma *et al.* [34] proposed in their work an outlier removal method based on preserving local neighborhood structures. They formulate their idea into a mathematical model and derive a closed-form solution with linear time complexity. Jiang *et al.* [21] presented a matching method using adaptive spatial clustering of putative matches based on motion consistency, considering also an additional "mismatch cluster".

Lee *et al.* [23] formulate the problem of the matching subprocess as a Markov random field. They use both, local descriptor distances and relative geometric similarities, to enhance robustness and accuracy. Liu *et al.* [28] presented a new matching method, contributing an advanced consensus of neighborhood topology. Combining it with a guided matching strategy from potential matches for neighborhood construction, results in improved inlier detection. Recent work of Liu's *et al.* [27] also includes a matching method particularly for remote sensing images. Inspired by region growing segmentation, they determine a high-ratio inlier subset as the seed (matching) set. It is then used to extract more reliable matches by an correspondence growing criterion based on motion consistency. Mohammed and El-Sheimy [37] presented a descriptorless feature matching. In addition to geometrical constraints, it also uses template matching to achieve a reasonable prediction of correspondence locations and their distribution.

Yi *et al.* [64] use deep learning for feature matching. Their neural network requires a set of potential sparse matches and the ground truth camera intrinsic parameters as input. It is used to label the test matching set as inliers or outliers and to output the camera motion. In the work of Wang *et al.* [60] learning of local feature matches is realized by solving a differentiable optimal transport problem. Corresponding

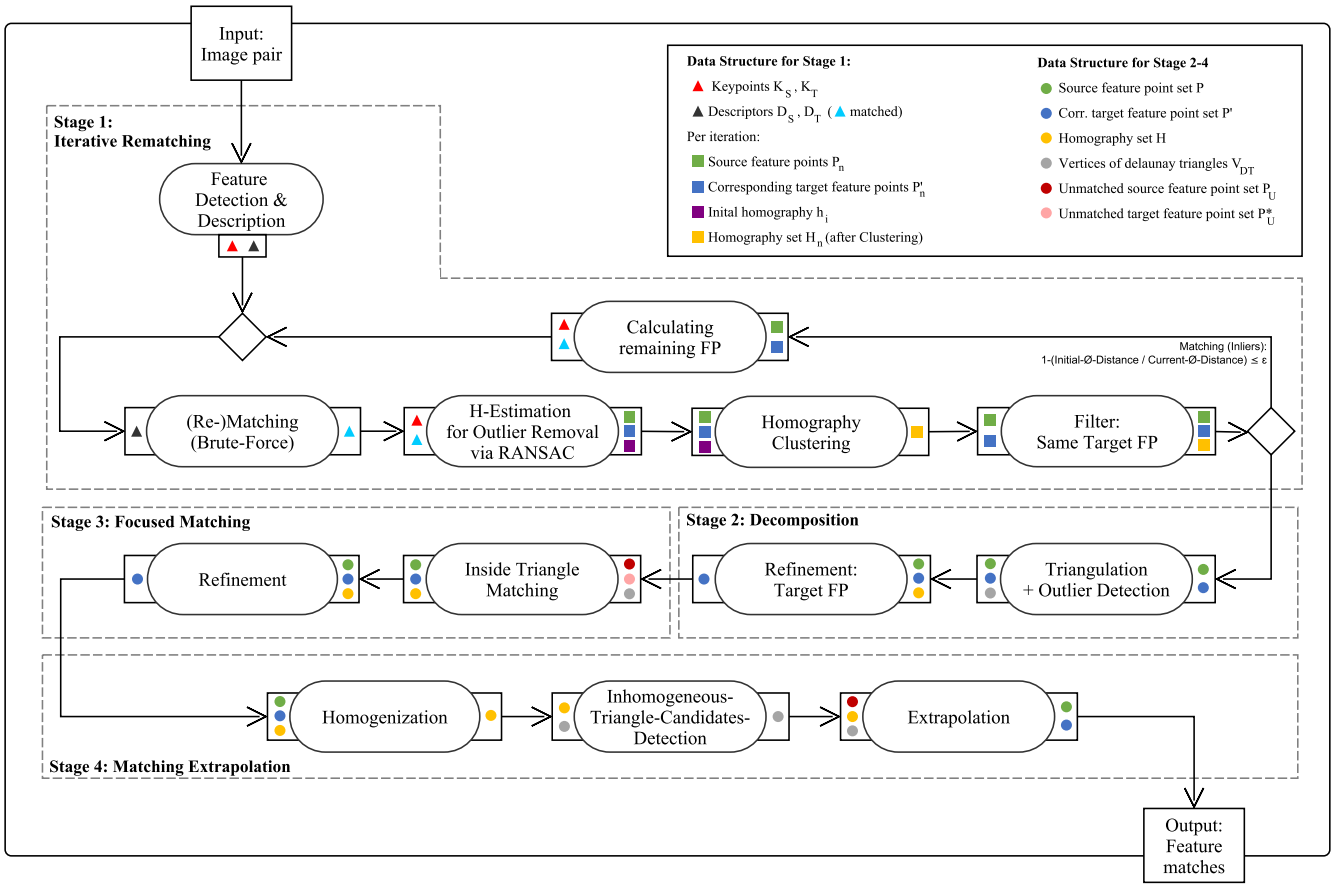


FIGURE 2. **UML-based activity diagram of our feature matching pipeline, including the main stages and the individual processes. The pipeline works with two basic data structures: "reiteration data structure" (stage 1) and "refinement data structure" (stages 2-4). The colored symbols (triangles, squares and circles) represent contained data entries that a process uses as input, or that are modified by a process as output. Further explanations and details are elaborated in section III.**

costs are predicted by a graph neural network. Ma *et al.* [33] and Li *et al.* [24] interpret mismatch removal as a binary classification problem. They use different sets of geometrical properties to describe the putative matches and to feed corresponding match representations to supervised learning procedures.

Other related work by Chen *et al.* [11] refers to stabilization of stereo image correspondences. Starting with a pre-computed set of reliable feature correspondences, each image is divided into triangles using Delaunay triangulation (similar to the triangulation process in our second pipeline stage). Then, the resulting triangle set is processed using a specific "planarity test" in order to reconstruct planes in 3D space by depth calculation. In the next step, further feature correspondences are computed in each planar region using the corresponding homographies. In contrast to our work, Chen at. el. require the estimation of an initial fundamental matrix, which can often be error-prone in practice and thus can have a negative impact on following processing steps. Moreover, the detection of "critical image areas", like occlusions, and further pipeline optimizations are not considered in their solution. Further work on feature matching was presented by Dou *et al.* [16]. They also take advantage of Delaunay triangulation for outlier detection: After initial matching, Dou et. al. try to remove false matches by utilizing sparse approximation theory. Then, the remaining feature points are triangulated separately in each image for final outlier removal by searching for triangles with non-corresponding vertices in the image pair. However, this approach strongly depends on the correctness of the initial matching stage. Even a small set of incorrect matches can lead to locally inconsistent triangulation and thus significantly degenerate the final number of positive matches.

Our work differs from so-called "dense correspondence search methods" that use a pixel-wise alignment in their pipelines: Examples are SIFT-Flow [26], RANSAC-Flow [50], GLU-Net [59] and "patch-based methods", such as PatchMatch [5], [6].

III. ITERATIVE DENSE FEATURE MATCHING PIPELINE

In the following sections we describe the four main stages of our matching pipeline. Each stage is composed of individual processing steps, as shown in Fig. 2.

A. ITERATIVE REMATCHING

Our first pipeline stage consists of the repetition of the following processing steps, which are re-executed on the set of yet unmatched feature points (including also preliminary mismatches from previous iterations): First, we run brute-force feature matching to determine for each point in the ''source image'' the nearest neighbor in the ''target image''. In the next step, we use the RANSAC algorithm for estimating a homography matrix with most inliers for the current matching set, rejecting associated outliers. A homographic relation can be used to describe feature correspondences for points, which lie on the same plane in 3D space. However, in practice one homographic plane can span over multiple surfaces, e.g. of different objects. Thus, we improve the quality of homography estimation by clustering feature points and recalculation of a (more precise) homographic transformation per cluster. Our motivation here is visual coherence and the observation that a cluster-based re-processing increases the probability for better surface approximation. For automatic clustering we use the ''Density-Based Spatial Clustering of Applications with Noise'' (DBSCAN) algorithm [17].

We have also implemented the so-called *target collapse filter*, which is executed per iteration to detect ''degenerated matches''. This happens, if multiple feature points of the source image are matching with the same point of the target image (see Fig. 3). Matching degeneration occurs typically in the following case: First, close feature points in the source image have a visually comparable local texture. Additionally, too few features could have been detected in the corresponding target image area. Such collapsing feature points basically indicate wrong matches w. r. t. homography estimation. Consequently, they have to be excluded in the following steps to prevent pipeline failures. To support fast convergence of the collapse filter, the exclusion is performed in both feature sets, for the source and for the target image. Then, the last iteration is repeated to trigger the recalculation of the corrected homography matrix.

FIGURE 3. Example of degenerated nearest neighbor matching (distinct features in the source image collapse in the target).

The termination of our iterative rematching stage is controlled by the *relative rematching distance error* (RRDE), which is calculated as follows: Let d_1 be the initial average matching distance (after first iteration) and d_i the average matching distance of the last executed iteration $i \geq 1$, then:

$$RRDE_i = 1 - \frac{d_1}{d_i} \tag{1}$$

Notice that the following always applies: $d_i \geq d_1$. Let $\varepsilon \in [0, 1[$ be a user-defined parameter. Then, rematching is terminated after iteration i, as soon as condition $\varepsilon \geq RRDE_i$ is met the first time. ε is a threshold parameter, which can be used to control the trade-off between the desired matching density and quality of the rematching stage: The larger its value is chosen, the more iterations can be performed. On the one hand, this allows the detection of more feature points and more differentiated homographic relations (each potentially corresponding to a plane in the scene presented by the images). On the other hand, this results in a potentially increased number of false matches due to the continuously increasing matching distances d_i per iteration. For our evaluations in section IV, a heuristically chosen error value $\varepsilon = 2/3$ has proven to be a reasonable trade-off between achieving the presented high matching densities and preserving superior matching quality in terms of accuracy of recall.

The result of the rematching pipeline stage is a base set of feature point pairs (between the source and target image) and a set of homography matrices. Each feature pair is associated with a distinct homography.

B. HOMOGRAPHIC DECOMPOSITION

The next pipeline stage implements stage-two outlier removal and refinement of the feature matching results from the previous step: First, the feature point set of the source image is triangulated using Delaunay mesh generation. Then, the resulting mesh is mapped to the matching point set of the target image in order to detect further outliers. This refers to the ''repeated image patterns problem'', mentioned in section I. We identify false matches based on a target *mesh consistency check*: Feature points, which are incident to an overlapping mesh edge, are successively removed from the base set of matching feature point pairs, as illustrated in Fig. 4.

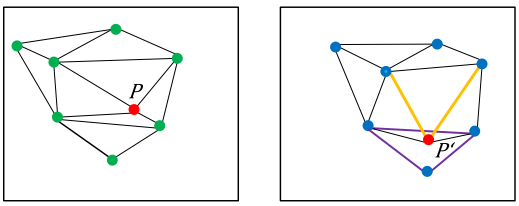


FIGURE 4. Detection of mesh overlaps by mapping from source to target image. The red feature point p' causes a mesh overlap (yellow edges) in the target image. Thus, the corresponding pair (p, p') is deleted from the base set.

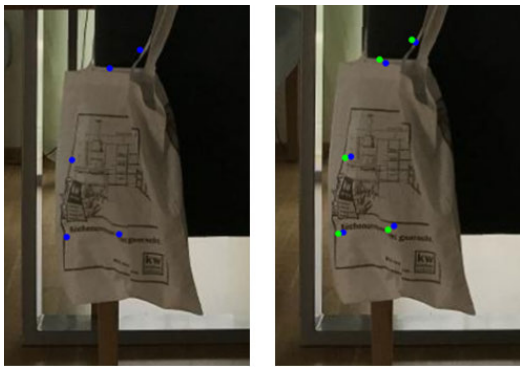
The next step aims at further improvement of the matching accuracy. Each source feature point $p \in P$ is associated with a homography matrix $h \in H$ and embedded in a triangular mesh structure. Hence, we can take advantage of its connectivity information and search for a better match in the target image as follows: (a) Successive transformation of p using neighboring homographies, (b) recalculation of the matching distance at each transformed position, and (c) comparing it to the initial distance. This *feature point refinement* is shown

Algorithm 1 Feature Point Refinement

Input: Source feature point set P, corresponding target feature point set P', homography set H
Output: Refined P'

for each $p \in P$ and corresponding $p' \in P'$ **do**
  $d_{current} :=$ Original matching distance $d(p, p')$
  for each $h \in H \mid$ neighbor of p has homography h **do**
    Transform p into the target image: $p_R := h * p$
    if $d(p, p_R) < d_{current}$ **and** p_R inside neighboring triangles of p **then**
      Replace target feature point p' by p_R
      Replace homography of p by h
      $d_{current} := d(p, p_R)$
    end if
  end for
end for

in algorithm 1. It turned out that this step can significantly contribute to the reduction of matching ambiguities, as illustrated in Fig. 5. We call the refined set of feature point pairs, including the homography associations, a *homographic decomposition*.



(a) (b)

FIGURE 5. (a) Source image with example feature points (blue); (b) Target image with initially matched feature points (blue) and refined points (green).

C. FOCUSED MATCHING

In the third pipeline stage we perform *focused matching* to find additional good corresponding features: We re-execute rematching and feature refinement (as described in sections 3.1 and 3.2), considering the remaining set of unmatched points. But now, each execution is restricted to one corresponding triangular area of the source and target images. This restriction is intended to mimic ''visual focusing''. Thereby, it is possible to detect further detail features and to estimate new local homographies. Focused matching is most effective for (larger) triangles with complex visual structures. So, to assure robust RANSAC-based homography estimation, we skip triangles with too few feature points, recommending a threshold value of at least 16 points per triangle. Finally,

Algorithm 2 Feature Point Extrapolation

Input: Unmatched source feature point set P_U, Vertex set V_t of triangle t, homography set H_{V_t} for V_t, minimal matching distance d_{min} between V_t and target V'_t
Output: Extrapolated target feature point set P'_E, corresponding source feature point set P_E

for each $p \in P_U \mid p \in \text{area}(t)$ **do**
  for each $h \in H_{V_t}$ **do**
    $p'_C := h * p$
    $P'_C := P'_C \cup \{p'_C\}$
  end for
  $p' := \text{p} \in P'_C \mid d(p, p') \rightarrow min.!$
  if $d(p, p') < d_{min}$ **then**
    $P_E := P_E \cup \{p\}$
    $P'_E := P'_E \cup \{p'\}$
  end if
end for

the initial triangle mesh and homographic decomposition are both updated by adding the newly detected features correspondences and homographic matrices, respectively.

D. MATCHING EXTRAPOLATION

In the last stage, we concentrate on the detection of further feature points in ''critical image areas''. We identify these areas by ''inhomogeneous triangles''. *Homogeneity*, in context of our feature point mesh, is defined for a triangle t with vertices $v_i \in V_t$ as follows: Let $h_i \in H$ be the (initially) associated homography matrix with feature point vertex v_i. Then, t is homogeneous, if:

$$\exists h_{hom} \in H, \quad \forall i \in \{1, 2, 3\} : h_{hom} * v_i \approx h_i * v_i. \qquad (2)$$

The ''raw'' homographic decomposition typically exhibits a high degree of variance in homography-to-vertex associations (with initially one homography per vertex). Therefore, in order to improve the detection quality in inhomogeneous triangles we have implemented the so-called *homogenization* process: Every feature point p is transformed successively from the source to the target image using a homography matrix from the set of neighboring feature points. Then, we search for transformed feature points in the target image, whose reprojection error is smaller than the threshold parameter of the RANSAC algorithm. The neighboring homographies, which satisfy the aforementioned condition, are additionally assigned to p. Notice that now a feature point in the resulting *homogenized homography decomposition* can be associated with multiple (locally equivalently transforming) matrices.

Finally, we execute the *feature point extrapolation* algorithm to obtain further good matches in inhomogeneous triangles: First, each yet unmatched source feature point p is transformed successively to the target image using local matrices of the homographic decomposition. The result is a set of candidate target features points P'_C. If the minimal

TABLE 1. Conventional feature matching (baseline) versus dense feature matching (DFM) with different detectors and descriptors.

		Baseline Matching			DFM		
Dataset	**Feature Detector/Descriptor**	**Average Recall**	**Average Precision**	**Average Q**	**Average Recall**	**Average Precision**	**Average Q**
Oxford [35] [36]	SIFT [30]	82.5%	34.6%	0.10	92.5%	96.8%	0.87
	SURF [7]	79.9%	54.5%	0.24	91.3%	97.0%	0.86
	FAST [46] + BRIEF [10]	76.1%	49.9%	0.19	92.0%	95.9%	0.85
	ORB [47]	84.9%	66.5%	0.38	92.9%	93.4%	0.81
	Superpoint [14]	85.5%	69.7%	0.42	92.1%	97.0%	0.87
	LF-Net [39]	82.3%	75.3%	0.47	89.8%	93.1%	0.78
	LIFT [63]	79.8%	77.6%	0.48	90.7%	94.4%	0.81
	ASLFeat [31]	87.7%	78.3%	0.54	91.9%	96.6%	0.86
AdelaideRMF [62]	SIFT [30]	51.9%	59.7%	0.15	90.9%	97.0%	0.86
	SURF [7]	51.1%	57.9%	0.17	90.1%	94.1%	0.80
	FAST [46] + BRIEF [10]	48.2%	58.8%	0.17	89.5%	96.3%	0.83
	ORB [47]	55.4%	64.3%	0.23	91.8%	93.3%	0.80
	Superpoint [14]	60.1%	71.9%	0.31	90.7%	96.7%	0.85
	LF-Net [39]	69.5%	75.3%	0.39	91.1%	95.5%	0.83
	LIFT [63]	63.7%	72.8%	0.34	89.1%	94.3%	0.79
	ASLFeat [31]	60.5%	73.9%	0.33	92.4%	97.3%	0.87
MultiH [3]	SIFT [30]	52.5%	61.4%	0.13	88.7%	90.7%	0.73
	SURF [7]	53.9%	60.5%	0.20	85.4%	88.7%	0.67
	FAST [46] + BRIEF [10]	50.2%	59.9%	0.18	85.9%	90.5%	0.70
	ORB [47]	59.1%	67.4%	0.27	88.3%	91.6%	0.74
	Superpoint [14]	63.9%	72.5%	0.34	89.3%	90.3%	0.73
	LF-Net [39]	71.1%	78.1%	0.43	91.1%	90.2%	0.74
	LIFT [63]	68.4%	72.3%	0.36	87.8%	91.4%	0.73
	ASLFeat [31]	63.3%	75.8%	0.36	90.1%	91.5%	0.75

matching distance between p and P'_C is smaller than the minimal local matching distance d_{min}, then p and the corresponding $p' \in P'_C$ are added to the extrapolation sets P_E and P'_E, respectively. The union of P_E and P'_E with the respective feature point sets from previous processing steps represent the final result of our matching pipeline. The extrapolation is shown in detail in algorithm 2 and illustrated in Fig. 6.

IV. IMPLEMENTATION AND RESULTS

Our pipeline was developed in C++ using the OpenCV library. Apart from parallelization of the homogenization and extrapolation algorithms (subsection III.D), no other run-time or memory optimizations have been implemented yet. The following benchmarks have been performed on a PC with an Intel i9-9900K CPU, 32 GB RAM and Windows 10.

A. QUANTITATIVE EVALUATION

To evaluate our feature matching pipeline, we have used the following image data: From the classic Oxford matching dataset [35], [36], "Wall" and "Graf" scenes were picked, because (only) these images satisfy our target use cases, defined in section I. "Neem", "Elderhall-A", "Elderhall-B", "Johnsonn-A", "Johnsonn-B", "Ladysymon", "OldClassicsWing", "Sene", "Napier", "UnionHouse", "Hartley" and "Physics" were used from the AdelaideRMF dataset [62], available for testing and comparisons with geometric model fitting methods. All image pairs

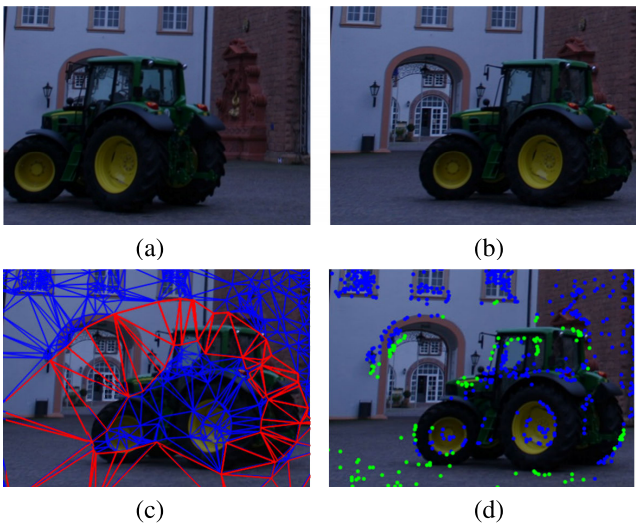


FIGURE 6. (a,b) Input image pair; (c) Target feature point triangulation with homogeneous (blue) and inhomogeneous triangles (red); (d) Initially matched target features (blue) and additionally extrapolated target features (green).

(9 scenes) were picked from the MultiH dataset [3], published for evaluation of multi-plane fitting methods in stereo images. All datasets provide ground truth matching information.

Our quantitative evaluation is based on the numerical indicators "precision" (p) and "recall" (r), as described by Agarwal and Roth [1]. Let n_{TP} be the number of true positive

TABLE 2. Detailed matching results, including ablation studies (w.r.t. executed pipeline stages) and time measurements (Extrapolation configuration: Execution in homogeneous ("H") and/or inhomogeneous ("I") regions, Ablation studies: Pipeline execution respectively without ("w/o") stage 2, stage 3 or stage 4).

ID	Scene	Feature Config.	Extra-polation	Stage 1 Matches	Stage 2 Matches	Stage 3 Matches	Stage 4 Matches	True Positives	False Positives	False Negatives	Run-time (s)
W1	Wall [35] [36]	Sparse	H+I	1.551	1.307	1.712	2.311	2.167	144	198	5.5
W1.1				1.551	**w/o**	1.700	2.510	2.111	399	153	5.1
W1.2				1.551	1.307	**w/o**	1.988	1.883	105	187	5.3
W1.3				1.551	1.307	1.712	**w/o**	1.618	94	231	2.7
W2		Dense	H+I	5.299	5.133	7.136	10.025	9.700	325	785	24.1
W2.1				5.299	**w/o**	6.978	9.899	9.396	503	812	22.4
W2.2				5.299	5.133	**w/o**	8.584	8.307	277	744	22.2
W2.3				5.299	5.133	7.136	**w/o**	6.895	241	797	9.6
G1	Graf [35] [36]	Sparse	H+I	2.912	2.577	4.133	7.192	5.734	1.458	183	23.9
G1.1				2.912	**w/o**	4.222	7.301	5.294	2.007	181	19.8
G1.2				2.912	2.577	**w/o**	5.811	4.729	1.082	169	20.2
G1.3				2.912	2.577	4.133	**w/o**	2.922	1.211	201	13.1
G2		Dense	H+I	1.104	1.043	1.469	2.097	1.747	350	64	5
G2.1				1.104	**w/o**	1.553	2.043	1.593	450	55	4.1
G2.2				1.104	1.043	**w/o**	1.693	1.357	336	60	4.2
G2.3				1.104	1.043	1.469	**w/o**	1.167	302	101	2.7
N1	Neem [62]	Sparse	I	2.007	1.896	2.774	3.290	3.044	246	331	6.1
N1.1				2.007	**w/o**	2.794	3.127	2.755	372	335	5.9
N1.2				2.007	1.896	**w/o**	2.378	2.199	179	302	4.7
N1.3				2.007	1.896	2.774	**w/o**	2.643	131	421	4.3
N2		Dense	I	6.388	5.893	8.132	10.860	10.552	308	1.115	24.3
N2.1				6.388	**w/o**	7.741	10.027	9.025	702	1.121	23.4
N2.2				6.388	5.893	**w/o**	8.322	8.038	284	1.081	17.3
N2.3				6.388	5.893	8.132	**w/o**	7,.934	198	1.333	16.6
N3			H+I	6.388	5.893	8.132	15.362	14.906	456	1.204	30.5
E1	Elderhall [62]	Sparse	I	1.422	1.251	1.798	2.964	2.788	176	289	10.4
E1.1				1.422	**w/o**	1.687	2.832	2,.441	391	421	9.4
E1.2				1.422	1.251	**w/o**	2.218	2.083	135	263	8.5
E1.3				1.422	1.251	1.798	**w/o**	1.695	103	337	6.3
E2		Dense	I	6.549	5.889	7.288	9.556	8.593	963	833	35.8
E2.1				6.549	**w/o**	6.979	9.327	7.676	1.651	931	32.6
E2.2				6.549	5.889	**w/o**	8.155	7.322	833	787	30.5
E2.3				6.549	5.889	7.288	**w/o**	6.627	661	927	20.4
E3			H+I	6.549	5.889	7.288	15.180	13.901	1.279	793	50.1

matches, n_{FP} false positive matches and n_{FN} false negative matches, then:

$$p = n_{TP}/(n_{TP} + n_{FP}), \tag{3}$$

$$r = n_{TP}/(n_{TP} + n_{FN}). \tag{4}$$

Feature pairs whose distance to the ground-truth epipolar line is smaller than a certain threshold d_{in} in both images (see below) are regarded as true positives, or in the other case as false negatives. To guarantee uniform comparability for different image resolutions, the threshold d_{in} is determined by $\alpha\sqrt{h^2 + w^2}$, where h and w are height and width of an image, respectively. The user-defined precision factor α is set to 0.003, as proposed by Bian *et al.* [8].

Additionally, we propose a new numerical indicator for feature matching based on recall and precision. The *Q-indicator*, is a simple single-value measure for determining the overall "matching quality", emphasizing precision:

$$q = r \cdot p^2 \tag{5}$$

In the first part of our evaluation we performed feature matching tests on the aforementioned datasets with the goal of demonstrating the flexibility and robustness of our pipeline. For this purpose, we used various feature detectors and descriptors, including a comparison to baseline matching (cf. Table 3): SIFT [30], SURF [7], FAST [46] with BRIEF [10], ORB [47], SuperPoint [47], LF-Net [39], LIFT [63] and ASLFeat [31].

In the second part, we evaluated the robustness of our pipeline, determined detailed matching results as well as time measurements of our algorithms and performed ablation studies to justify the contribution of the individual pipeline stages (cf. Table 2). For this purpose, we generated sparse and dense feature sets for the following image pairs, respectively: "Wall 1", "Wall 3" (W), "Graf 1", " Graf 3" (G) from the Oxford dataset and "Neem" (N), "Elderhall-B" (E) from the AdelaideRMF dataset. The sparse feature matching configurations have approximately three to four times less initial matches (pipeline stage 1) compared to the dense configurations. In these tests we also enabled the feature point extrapolation for inhomogeneous triangles (see section III.D), tagged with "H+I" in Table 2. This makes it possible to consider even more true positives matches and thus to get a even more representative recall value. A selection of corresponding visual results is shown in Fig. 8.

TABLE 3. Comparison of different matching methods, including per image pair averaging for each dataset (SIFT was used for feature detection and description).

Dataset	Matching method	Average Recall	Average Precision	Average Q	Harmonic Recall	Harmonic Precision	Harmonic Q
Oxford [35] [36]	Baseline	82.5%	34.6%	0.10	79.7%	34.2%	0.09
	GMS [9]	49.8%	51.1%	0.13	46.9%	50.5%	0.12
	LPM [34]	57.4%	55.6%	0.18	53.9%	53.6%	0.15
	CODE [25]	88.9%	81.4%	0.59	88.0%	81.1%	0.58
	PFM [23]	86.9%	82.2%	0.59	86.1%	81.5%	0.57
	RFM-SCAN [21]	90.3%	93.2%	0.78	89.4%	93.0%	0.77
	LC [64]	87.1%	64.4%	0.36	86.3%	62.2%	0.33
	LMR [33]	86.3%	87.3%	0.66	84.6%	86.8%	0.64
	Our (DFM)	**92.5%**	**96.8%**	**0.87**	**91.6%**	**96.2%**	**0.85**
AdelaideRMF [62]	Baseline	49.5%	55.5%	0.15	45.5%	53.4%	0.13
	GMS [9]	60.5%	57.1%	0.20	54.9%	55.9%	0.17
	LPM [34]	58.7%	56.2%	0.19	55.0%	56.5%	0.18
	CODE [25]	90.1%	87.8%	0.69	89.7%	87.5%	0.69
	PFM [23]	82.1%	88.3%	0.64	81.2%	85.6%	0.60
	RFM-SCAN [21]	89.9%	92.2%	0.76	89.1%	92.1%	0.76
	LC [64]	79.6%	70.1%	0.39	78.8%	70.5%	0.39
	LMR [33]	83.7%	86.0%	0.62	83.3%	85.9%	0.61
	Our (DFM)	**90.9%**	**97.0%**	**0.86**	**90.7%**	**97.0%**	**0.85**
MultiH [3]	Baseline	41.2%	55.3%	0.13	36.1%	53.6%	0.10
	GMS [9]	60.2%	58.8%	0.21	59.0%	56.5%	0.19
	LPM [34]	72.3%	61.9%	0.28	63.6%	61.1%	0.24
	CODE [25]	85.9%	79.9%	0.52	85.0%	79.6%	0.54
	PFM [23]	84.4%	81.1%	0.56	83.7%	79.6%	0.53
	RFM-SCAN [21]	85.1%	88.3%	0.66	84.6%	87.4%	0.65
	LC [64]	78.3%	69.9%	0.38	74.4%	68.1%	0.35
	LMR [33]	81.2%	84.1%	0.57	80.6%	83.8%	0.57
	Our (DFM)	**88.7%**	**90.7%**	**0.73**	**89.3%**	**90.6%**	**0.73**

In the last part, we performed similar matching tests as in part 1, but now we compared our proposed pipeline to the following matching methods: GMS [9], LPM [34], CODE [25], PFM [23], RFM-SCAN [21], LC [64] and LMR [33] in combination with RANSAC for geometric model estimation (as we also set up our pipeline with RANSAC). The results are summarized in Table 1, including arithmetic and harmonic mean values, and in Fig. 7, showing cumulative distributions of precision and recall. For this evaluation, we have generated different ''training feature sets'' for the corresponding image pairs in order to tune the parameters of each matching algorithm as best as possible. The results reported in this paper are based on the execution of each corresponding matching algorithm only once (for each image pair per dataset). For this purpose, the initially estimated and fixed parameters are used, thus giving the final ''test feature sets''. For LC and LMR we used the pre-trained model released by the authors. In all tests we used the SIFT algorithm for feature detection and description.

B. RESULTS AND DISCUSSION

As can be seen in Table 1, the comparison with conventional matching shows that our solution achieves consistent and stable improvements in matching quality even for different descriptors and detectors. The use of our pipeline as an ''matching framework'' makes it possible to reach high precision and recall values also with ''traditional'' algorithms, like SIFT. These can then compete even with modern (ML-based) methods, like Superpoint, LF-Net, LIFT and ASLFeat.

The detailed matching results in Table 2 (with all pipeline stages executed) can be summarized as follows: Stage 1 (iterative rematching) detects between 40% and 69% of all matches. Focused matching (stage 3) adds between 10% and 27% matches. For datasets with a planar scene setup (W and G), additional 26% to 43% of matches are included due to feature point extrapolation (stage 4). For datasets with higher depth complexities (N and E), extrapolation contributes to 16% to 40% extra matches (test configuration ''I'') and 47% to 52% matches (test configuration ''H+I''). The results of our ablation studies in Table 2 can be concluded in the following way: The basic homographic decomposition step (stage 2) has not only a direct impact on the total number of matches, but it is also crucial for matching precision: Discarding stage 2 results in a significant increase of n_{FP} by 27% up to 128%. In particular, we can see that the implemented Delaunay mesh consistency check has the potential to significantly reduce the number of false positives, typically caused by repeated image patterns in this case. Skipping focused matching (stage 3) results in an decrease of the overall matching count by 13% up to 27%. But, n_{TP}, n_{FP} and n_{FN} remain roughly stable in proportion to the overall count. If feature point extrapolation (stage 4) is disabled, then the total number of matches decreases by 16% up to 43% (causing also n_{TP} and n_{FP} to decrease). However, n_{FN} increases up to 27%,

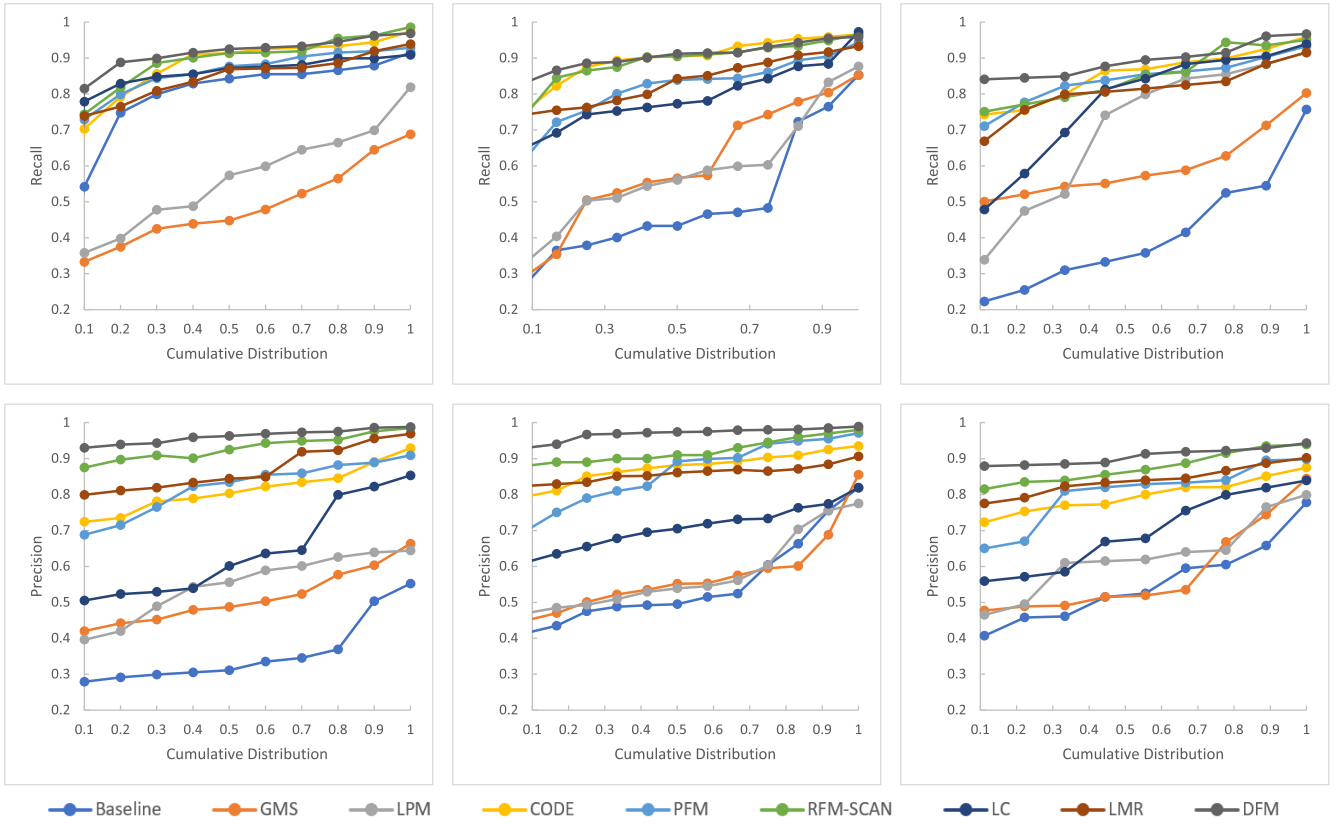


FIGURE 7. Cumulative distributions of recall and precision values for different matching methods and following datasets (left to right): Oxford [35], [36], AdelaideRMF [62] and MultiH [3]. A point (x,y) on one of the curves implies that there are $(100 \cdot x)$% of image pairs whose recall/precision does not exceed y in each case.

which has a negative impact on the matching recall. Since the precision values remain largely stable with and without extrapolation, stage 4 supports the generation of true positive matches in discontinuity regions without causing additional side effects that could have a negative impact on the pipeline results.

The performance bottleneck in the current (yet non-optimized) implementation are stages 3 and 4: Feature point refinement requires 31% to 41% and extrapolation 32% to 46% of total run-time. Stage 1 (iterative rematching) requires 7% to 10% and stage 2 (initial homographic decomposition) 12% to 18% of run-time. Beyond that, a direct interpretation or comparison of the DFM run-times would have only limited significance: On the one hand, one key requirement in our work was to achieve high feature densities, while maintaining high precision (rather than high run-time performance). Consequently and as motivated in section I, we are focusing on pre-processing, such as for image-based computer graphics applications (and not real-time feature matching, for example). One the other hand, our pipeline corresponds to a multi-stage designed software framework in which individual (''one-step'') matching methods can be integrated and then executed iteratively (as demonstrated in Table 1). In Table 4 we give an overview of the worst-case time complexities of all pipeline algorithms.

TABLE 4. Time complexities for our pipeline. An overview of the variables used in this table can be found in Fig. 2. Additionally, T_{DT} and V_{DT} refer to the Delaunay triangle set and corresponding vertex set from subsection III.B.

Pipeline processing step		Time complexity
Stage 1	Matching	$O(\|K_S\| \cdot \|K_T\|)$
	H-Estimation	$\rightarrow$ RANSAC [42], [43]
	Clustering	$O(\|P_n\| \cdot log\|P_n\|)$
	Filter	$O(\|P_n\|)$
Stage 2	Triangulation	$O(\|P\|^2)$
	Outlier Detection	$O(\|P\| \cdot \|T_{DT}\|)$
	Refinement	$O(\|P\| \cdot \max_{v \in V_{DT}} deg(v))$
Stage 3	Inside Matching	$O(\|T_{DT}\| * \|P_U\|)$
	Refinement	See stage 2
Stage 4	Homogenization	$O(\|P\| \cdot \max_{v \in V_{DT}} deg(v))$
	Inhom.-DT-Detection	$O(\|T\|)$
	Extrapolation	$O(\|P_U\| \cdot \max_{v \in V_{DT}} deg(v))$

From the average recall and precision results of Table 3, we can see that our pipeline significantly improves both indicators on all datasets in comparison to the other evaluated

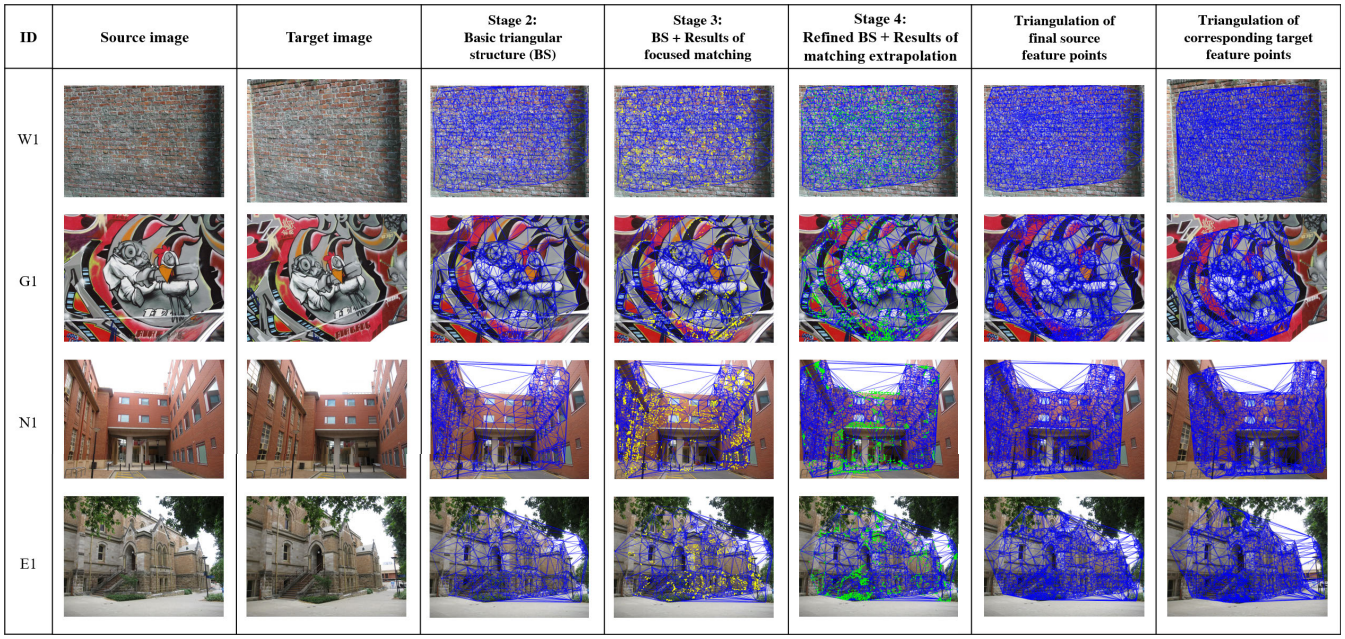


FIGURE 8. Visual results of each pipeline stage, including Delaunay mesh (blue), feature points from focused matching (yellow) and extrapolation (green).

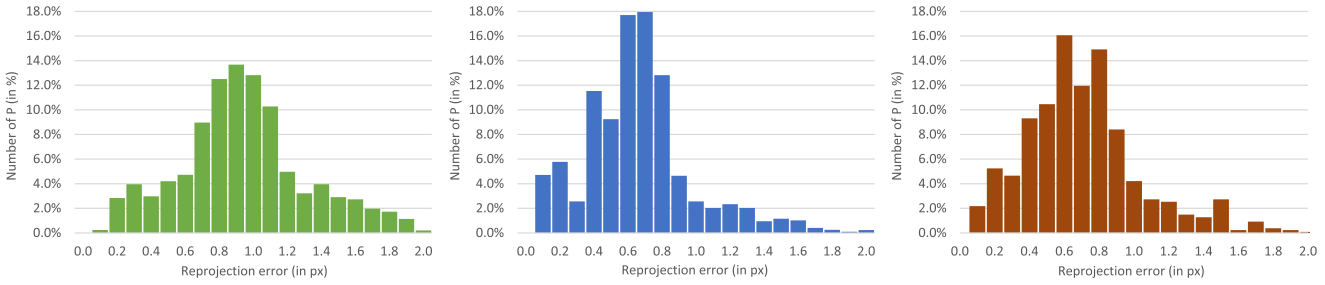


FIGURE 9. Histograms of reprojection errors for the following datasets (from left to right): Oxford [35], [36], AdelaideRMF [62] and MultiH [3].

matching methods. The precision and recall distributions are shown in Fig. 7. The corresponding key results can be summarized as follows: In particular, CODE and RFM-SCAN have mostly just slightly lower recall values than our DFM, but they do not achieve the same consistently high precision values. The evaluated ML-based methods (LC and LMR) have lower recall and precision values in all tests (except for a single peak recall value for LC and the AdelaideRMF dataset). We interpret this as a consequence of the limited or specific pre-trained models available.

C. APPLICATION EXAMPLES

One possible target use case of our pipeline is ''image morphing'': It is an image processing technique that generates smooth transitions between image pairs. Basic image morphing consists of the following two sub-steps: warping and blending [41]. In our example, we use triangle-based warping. The triangle vertices represent the feature correspondences of our matching pipeline. A dense and accurate feature matching set is crucial to reduce visual distortions as well as ghosting artifacts due to blending. Furthermore, image areas with occlusions or disocclusions are a challenge for image morphing, as visual artifacts occur especially there. Using our matching pipeline, we can detect such areas. We therefore plan to further utilize this information in our future research on feature matching and in particular image-based rendering applications. Corresponding first visual results are shown in Fig. 10. We have chosen the Castle-P30 dataset [53] to illustrate feature-oriented image morphing, because of the clearly available parallaxes.

Finally, to demonstrate the accuracy of our multi-homography decomposition, we computed the reprojection errors for each feature matching pair. Therefore, we used the results from our pipeline and the provided ground truth data of the corresponding datasets. Minimal reprojection errors (i.e. on average clearly smaller than one pixel) are important for high-quality visual results in context of image pre-processing for multi-view 3D reconstruction and photogrammetric 3D modeling applications, for example [13], [54]. In Fig. 9 we show the resulting reprojection error

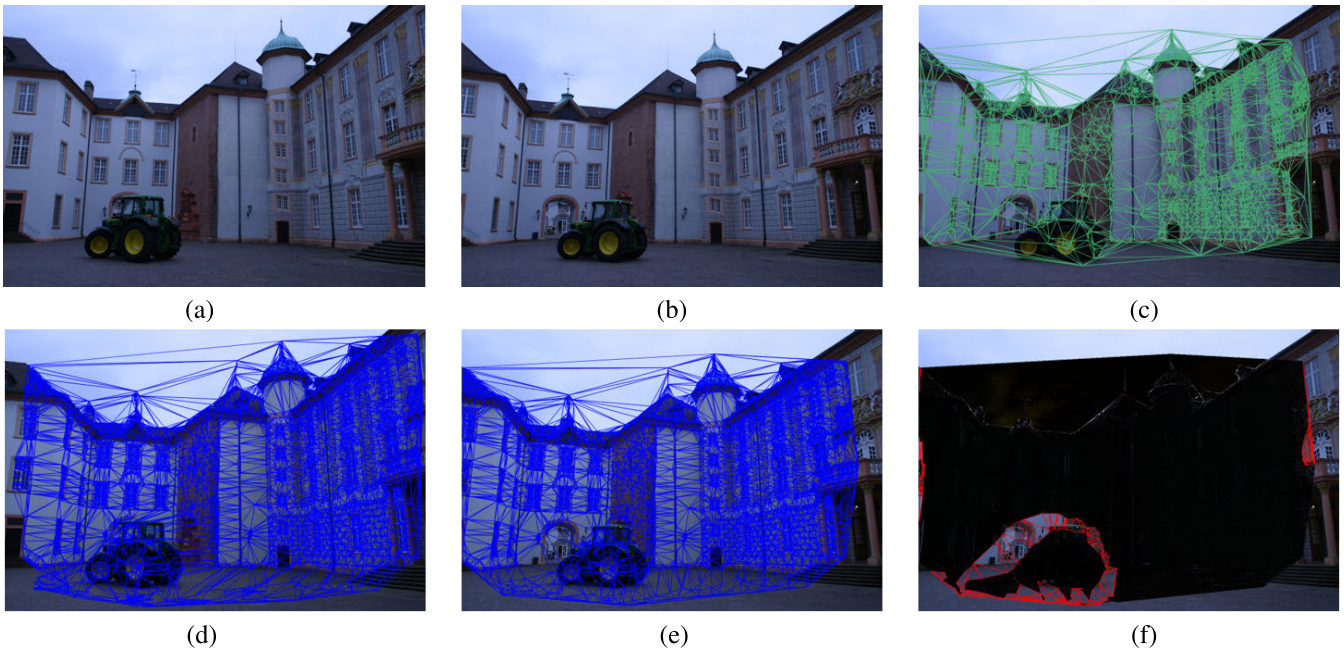


FIGURE 10. **(a,b) Source and target input images; (c) Triangulation of target feature points using the conventional pipeline; (d,e) Triangulation of source and target feature points using our pipeline; (f) Difference image after mesh-based source-to-target warping, highlighting detected visual occlusions ("inhomogeneous triangles") in red color.**

histograms of all our evaluation datasets (for a RANSAC reprojection threshold of 2.1 pixels). The corresponding ''root mean square reprojection error'' (RMSE) [48] is approximately 0.94 pixels for the Oxford dataset, 0.66 pixels for AdelaideRMF and 0.70 pixels for MultiH, respectively.

V. CONCLUSIONS AND FUTURE WORK

In this paper a novel feature matching approach was presented, which detects significantly more robust and accurate feature correspondences, compared to conventional and related state-of-the-art methods. A dense feature matching set can be generated also for scenes with high depth complexity. This opens up new application opportunities e. g. for use cases in computer graphics, including morphing, warping, 3D reconstruction, image-based modeling etc. Our work addresses the prevailing challenges commonly encountered in the development of feature-based image processing applications, providing a single-pipeline solution: The first pipeline stage, iterative rematching, comprises the homographic decomposition and cluster analysis of the image space. This bypasses the ''fundamental matrix degeneration problem'' and makes it possible to handle visually disturbing effects in following pipeline steps. Our Delaunay outlier detection (second stage) removes false positive matches, which are especially caused by ''repeated image patterns''. Additionally, matching accuracy is increased due to refinement of the matching set by taking advantage of our multi-homographic decomposition. Focused matching (third stage) simulates ''visual focusing'', resulting in the identification of additional detail feature points. Homogenization (last pipeline stage) supports the detection of ''inhomogeneous'' image regions that are typically caused by parallax effects. Even just their peripheral areas are difficult to match in practice, but still important for many of the aforementioned use cases. Our feature extrapolation makes it possible to detect further matches in these ''critical areas'', resulting in a refined multi-homography decomposition.

Current limitations of our pipeline concern the restrictions on the types of input datasets supported: Our method is designed for image sequences of RGB-colored photo or video shots with sufficient pairwise overlaps. The images should represent static scenes, i.e. excluding significant object motions, lighting changes and (specular) effects. Image data with large camera distortions, such as in ultra wide-angle or fisheye photography, has not yet been tested. The present focus is on high-quality matching and uses cases in context of offline (pre-)processing. Therefore, our algorithms are currently not optimized for performance-critical applications, much less real-time application scenarios.

Our future work includes research in context of automatic tuning of matching parameters, which would allow an easier-to-use and wider range of applications using homographic decomposition. Additionally, we plan to implement improvements of our algorithms, in particular with respect to runtime. This includes low-level and algorithmic optimizations, further multi-threaded processing and GPU acceleration.

ACKNOWLEDGMENT

This project was promoted by the Bavarian Academic Forum (BayWISS), as a part of the joint academic partnership digitalization program.

REFERENCES

[1] S. Agarwal and D. Roth, "Learning a sparse representation for object detection," in *Proc. Eur. Conf. Comput. Vis.*, 2002, pp. 113–127.
[2] C. Ancuti, C. O. Ancuti, and P. Bekaert, "An efficient two steps algorithm for wide baseline image matching," *Vis. Comput.*, vol. 25, nos. 5–7, pp. 677–686, May 2009.
[3] D. Barath, J. Matas, and L. Hajder, "Multi-H: Efficient recovery of tangent planes in stereo images," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2016, pp. 13.3–13.13.
[4] D. Barath, J. Matas, and J. Noskova, "MAGSAC: Marginalizing sample consensus," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10197–10205.
[5] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, "PatchMatch: A randomized correspondence algorithm for structural image editing," *ACM Trans. Graph.*, vol. 28, no. 3, p. 24, 2009.
[6] C. Barnes, E. Shechtman, D. B. Goldman, and A. Finkelstein, "The generalized PatchMatch correspondence algorithm," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 29–43.
[7] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Comput. Vis. Image Understand.*, vol. 110, no. 3, pp. 346–359, Jan. 2008.
[8] J.-W. Bian, Y.-H. Wu, J. Zhao, Y. Liu, L. Zhang, M.-M. Cheng, and I. Reid, "An evaluation of feature matchers for fundamental matrix estimation," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2019, pp. 89.1–89.14.
[9] J. Bian, W.-Y. Lin, Y. Matsushita, S.-K. Yeung, T.-D. Nguyen, and M.-M. Cheng, "GMS: Grid-based motion statistics for fast, ultra-robust feature correspondence," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4181–4190.
[10] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: Binary robust independent elementary features," in *Computer Vision—ECCV*. Berlin, Germany: Springer, 2010.
[11] C.-I. Chen, D. Sargent, C.-M. Tsai, Y.-F. Wang, and D. Koppel, "Stabilizing stereo correspondence computation using Delaunay triangulation and planar homography," in *Advances in Visual Computing* (Lecture Notes in Computer Science), vol. 5358, G. Bebis, Ed. Berlin, Germany: Springer, 2008, pp. 836–845.
[12] O. Chum, J. Matas, and J. Kittler, "Locally optimized RANSAC," in *Pattern Recognition* (Lecture Notes in Computer Science), vol. 2781, B. Michaelis and G. Krell, Eds. Berlin, Germany: Springer, 2003, pp. 236–243.
[13] A. Delaunoy and M. Pollefeys, "Photometric bundle adjustment for dense multi-view 3D modeling," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 1486–1493.
[14] D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperPoint: Self-supervised interest point detection and description," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 224–236.
[15] J. F. Dou and J. X. Li, "Automatic image mosaic based on SIFT using bidirectional matching," *Adv. Mater. Res.*, vols. 457–458, pp. 841–847, Jan. 2012.
[16] J. Dou, Q. Qin, and Z. Tu, "Robust image matching with cascaded outliers removal," *Pattern Recognit. Image Anal.*, vol. 27, no. 3, pp. 480–493, Jul. 2017.
[17] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd Int. Conf. Knowl. Discovery Data Mining*, 1996, pp. 226–231.
[18] M. A. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.
[19] J.-M. Frahm and M. Pollefeys, "RANSAC for (quasi-)degenerate data," in *Proc. Conf. Comput. Vis. Pattern Recognit. Workshop*, C. Schmid, S. Soatto, and C. Tomasi, Eds., Piscataway, NJ, USA, 2006, pp. 453–460.
[20] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. Alvey Vis. Conf.*, C. J. Taylor, Ed., 1988, pp. 23.1–23.6.
[21] X. Jiang, J. Ma, J. Jiang, and X. Guo, "Robust feature matching using spatial clustering with heavy outliers," *IEEE Trans. Image Process.*, vol. 29, pp. 736–746, 2020.
[22] K. Krishnakumar and S. I. Gandhi, "Video stitching based on multi-view spatiotemporal feature points and grid-based matching," *Vis. Comput.*, vol. 36, no. 9, pp. 1837–1846, Sep. 2020.
[23] S. Lee, J. Lim, and I. H. Suh, "Progressive feature matching: Incremental graph construction and optimization," *IEEE Trans. Image Process.*, vol. 29, pp. 6992–7005, 2020.
[24] Y. Li, Q. Huang, Y. Liu, Y. Huang, and X. Sun, "Efficient properties-based learning for mismatch removal," *IEEE Access*, vol. 7, pp. 149612–149622, 2019.
[25] W.-Y. Lin, F. Wang, M.-M. Cheng, S.-K. Yeung, P. H. S. Torr, M. N. Do, and J. Lu, "CODE: Coherence based decision boundaries for feature correspondence," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 1, pp. 34–47, Jan. 2018.
[26] C. Liu, J. Yuen, A. Torralba, J. Sivic, and W. T. Freeman, "Sift flow: Dense correspondence across different scenes," in *Proc. Eur. Conf. Comput. Vis.*, in Lecture Notes in Computer Science, vol. 5304. Berlin, Germany: Springer, 2008, pp. 28–42.
[27] Y. Liu, Y. Li, L. Dai, T. Lai, C. Yang, L. Wei, and R. Chen, "Motion consistency-based correspondence growing for remote sensing image matching," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022.
[28] Y. Liu, Y. Li, L. Dai, C. Yang, L. Wei, T. Lai, and R. Chen, "Robust feature matching via advanced neighborhood topology consensus," *Neurocomputing*, vol. 421, pp. 273–284, Jan. 2021.
[29] H. C. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections," *Nature*, vol. 293, no. 5828, pp. 133–135, 1981.
[30] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.
[31] Z. Luo, L. Zhou, X. Bai, H. Chen, J. Zhang, Y. Yao, S. Li, T. Fang, and L. Quan, "ASLFeat: Learning local features of accurate shape and localization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6589–6598.
[32] Q.-T. Luong and O. D. Faugeras, "The fundamental matrix: Theory, algorithms, and stability analysis," *Int. J. Comput. Vis.*, vol. 17, no. 1, pp. 43–75, 1996.
[33] J. Ma, X. Jiang, J. Jiang, J. Zhao, and X. Guo, "LMR: Learning a two-class classifier for mismatch removal," *IEEE Trans. Image Process.*, vol. 28, no. 8, pp. 4045–4059, Aug. 2019.
[34] J. Ma, J. Zhao, J. Jiang, H. Zhou, and X. Guo, "Locality preserving matching," *Int. J. Comput. Vis.*, vol. 127, no. 5, pp. 512–531, 2019.
[35] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool, "A comparison of affine region detectors," *Int. J. Comput. Vis.*, vol. 65, nos. 1–2, pp. 43–72, Nov. 2005.
[36] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1615–1630, Oct. 2005.
[37] H. Mohammed and N. El-Sheimy, "A descriptor-less well-distributed feature matching method using geometrical constraints and template matching," *Remote Sens.*, vol. 10, no. 5, p. 747, May 2018.
[38] D. Nister, "An efficient solution to the five-point relative pose problem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 6, pp. 756–777, Jun. 2004.
[39] Y. Ono, E. Trulls, P. Fua, and K. M. Yi, "LF-Net: Learning local features from images," in *Advances in Neural Information Processing Systems*, vol. 31. Red Hook, NY, USA: Curran Associates, 2018.
[40] O. Özyeşil, V. Voroninski, R. Basri, and A. Singer, "A survey of structure from motion," *Acta Numerica*, vol. 26, pp. 305–364, May 2017.
[41] A. Patel and P. Lapsiwala, "Image morphing algorithm: A survey," *Int. J. Comput. Appl.*, vol. 5, no. 8, pp. 360–372, 2015.
[42] R. Raguram, O. Chum, M. Pollefeys, J. Matas, and J.-M. Frahm, "USAC: A universal framework for random sample consensus," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 2022–2038, Aug. 2013.
[43] R. Raguram, J.-M. Frahm, and M. Pollefeys, "A comparative analysis of RANSAC techniques leading to adaptive real-time random sample consensus," in *Computer Vision—ECCV*. Berlin, Germany: Springer, 2008.
[44] R. Ranftl and V. Koltun, "Deep fundamental matrix estimation," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 284–299.
[45] R. Roberts, S. N. Sinha, R. Szeliski, and D. Steedly, "Structure from motion for scenes with large duplicate structures," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Piscataway, NJ, USA, Jun. 2011, pp. 3137–3144.
[46] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Proc. Eur. Conf. Comput. Vis.*, 2006, pp. 430–443.
[47] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, Piscataway, NJ, USA, Nov. 2011, pp. 2564–2571.
[48] W. Saif and A. Alshibani, "Smartphone-based photogrammetry assessment in comparison with a compact camera for construction management applications," *Appl. Sci.*, vol. 12, no. 3, p. 1053, Jan. 2022.
[49] S. Saxena and R. K. Singh, "A survey of recent and classical image registration methods," *Int. J. Signal Process., Image Process. Pattern Recognit.*, vol. 7, no. 4, pp. 167–176, Aug. 2014.
[50] X. Shen, F. Darmon, A. A. Efros, and M. Aubry, "RANSAC-Flow: Generic two-stage image alignment," in *Proc. 16th Eur. Conf. Comput. Vis.*, 2020, pp. 618–637.

[51] N. S. Skoryukina, I. A. Faradjev, K. B. Bulatov, and V. V. Arlazarov, "Impact of geometrical restrictions in RANSAC sampling on the ID document classification," in *Proc. 12th Int. Conf. Mach. Vis. (ICMV)*, Jan. 2020, pp. 35–41.
[52] N. Snavely, S. M. Seitz, and R. Szeliski, "Photo tourism: Exploring photo collections in 3D," in *Proc. ACM SIGGRAPH Papers*, New York, NY, USA, 2006, pp. 835–846.
[53] C. Strecha, W. von Hansen, L. Van Gool, P. Fua, and U. Thoennessen, "On benchmarking camera calibration and multi-view stereo for high resolution imagery," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Piscataway, NJ, USA, Jun. 2008, pp. 1–8.
[54] B.-Y. Sung and C.-H. Lin, "A fast 3D scene reconstructing method using continuous video," *EURASIP J. Image Video Process.*, vol. 2017, no. 1, pp. 1–14, Dec. 2017.
[55] X. Tan, C. Sun, X. Sirault, R. Furbank, and T. D. Pham, "Feature matching in stereo images encouraging uniform spatial distribution," *Pattern Recognit.*, vol. 48, no. 8, pp. 2530–2542, 2015.
[56] L. Tiwari and S. Anand, "DGSAC: Density guided sampling and consensus," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2018, pp. 974–982.
[57] P. H. S. Torr, A. Zisserman, and S. J. Maybank, "Robust detection of degenerate configurations for the fundamental matrix," in *Proc. IEEE Int. Conf. Comput. Vis.*, Los Alamitos, CA, USA, Jun. 1995, pp. 1037–1042.
[58] P. Truong, S. Apostolopoulos, A. Mosinska, S. Stucky, C. Ciller, and S. D. Zanet, "GLAMpoints: Greedily learned accurate match points," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 10732–10741.
[59] P. Truong, M. Danelljan, and R. Timofte, "GLU-Net: Global-local universal network for dense flow and correspondences," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6258–6268.
[60] A. Wang, Y. Pruksachatkun, N. Nangia, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman, "SuperGLUE: A stickier benchmark for general-purpose language understanding systems," in *Advances in Neural Information Processing Systems*, vol. 32. Red Hook, NY, USA: Curran Associates, 2019.
[61] G. Wolberg, "Image morphing: A survey," *Vis. Comput.*, vol. 14, no. 8, pp. 360–372, 1998.
[62] H. S. Wong, T.-J. Chin, J. Yu, and D. Suter, "Dynamic and hierarchical multi-structure geometric model fitting," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, Piscataway, NJ, USA, Nov. 2011, pp. 1044–1051.
[63] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua, "LIFT: Learned invariant feature transform," in *Computer Vision—ECCV*. Cham, Switzerland: Springer, 2016.
[64] K. M. Yi, E. Trulls, Y. Ono, V. Lepetit, M. Salzmann, and P. Fua, "Learning to find good correspondences," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Piscataway, NJ, USA, Jun. 2018, pp. 2666–2674.

SIMON SEIBT received the master's degree in computer science from the Nuremberg Institute of Technology, in 2018. He is currently a Research Assistant with the Game Tech Laboratory, Nuremberg Institute of Technology. His research interests include image processing and image-based rendering. He is a member of the Gesellschaft für Informatik (GI).

BARTOSZ VON RYMON LIPINSKI studied computer science at the University of Bonn. He received the Ph.D. degree from the Technical University of Munich, in 2007. After a professorship at the Media Design University of Applied Sciences, Munich, he became a Professor of media informatics at the Nuremberg Institute of Technology, in 2014. Since then, he supports teaching and research activities of the Faculty of Computer Science in the areas of interactive 3D applications and games engineering and heads the Game Tech Laboratory. His research interests include computer vision for computer graphics, image-based modeling, and rendering.

MARC ERICH LATOSCHIK studied mathematics and computer science at the University of Paderborn, the New York Institute of Technology, and Bielefeld University, where he headed the AI & VR Laboratory, until 2007. After professorships in Berlin and Bayreuth, he became the Chair for Human-Computer Interaction at Würzburg University, in 2011. His research interests include highly interactive and immersive interfaces and applications of virtual, augmented, and mixed reality. He is an Active Member of several academic and industrial societies, including the Association for Computing Machinery (ACM) and the Gesellschaft für Informatik (GI).