Ground Truth Accuracy and Performance of the Matching Pipeline*

Josef Maier Martin Humenberger Oliver Zendel
Markus Vincze
AIT Austrian Institute of Technology, Donau-City-Strasse 1, 1220, Vienna, Austria
Vienna University of Technology, Gußhausstrasse 27-29, 1040, Vienna, Austria
{josef.maier.fl;martin.humenberger;oliver.zendel}@ait.ac.at,
vincze@acin.tuwien.ac.at

Abstract

Feature matching quality strongly influences the accuracy of most computer vision tasks. This led to impressive advances in keypoint detection, descriptor calculation, and feature matching itself. To compare different approaches and evaluate their quality, datasets from related tasks are used. Unfortunately, none of these datasets actually provide ground truth (GT) feature matches. Thus, matches can only be approximated due to repeatability errors of keypoint detectors and inaccuracies of GT. In this paper, we introduce ground truth matches (GTM) for several well known datasets. Based on the provided spacial ground truth, we automatically generate them using popular feature types. Currently, feature matching evaluation is typically performed using precision and recall. The introduced GTM additionally enable evaluation with accuracy and fall-out. The datasets were manually annotated, on the one hand to evaluate the precision and unambiguousness of the GTM, and on the other hand to determine the accuracy of the ground truth provided with the datasets. Using GTM, we present an evaluation of multiple state-of-the-art keypoint-descriptor combinations as well as matching algorithms.

1. Introduction

Over the last decades, detection, tracking, and matching of distinctive parts of images became key components in computer vision. Keypoints (locations in the image) and descriptors (description of the keypoints' neighbourhood), together referred to as features, are most commonly used to localize and describe these parts. Image features enable applications such as camera pose estimation [55, 61], object detection and tracking [19], as well as visual localization [39, 48]. Even though "hand-crafted" features tend to be increasingly replaced by learning pipelines (*e.g.* Yi *et al.* [72] or Kendall *et al.* [28]), many real-time algorithms still rely on reliable and accurate feature detection, matching, and tracking. Various camera pose estimation algorithms use image features to estimate and track a cameras movement. Interesting applications of these algorithms are, among others, navigation for autonomous ground [22] and aerial vehicles [20], augmented reality [46], and structure from motion [24]. The accuracy of *e.g.* feature-based visual odometry (VO) or visual simultaneous localization and mapping (VSLAM) strongly depends on the quality of feature matches. These matches, however, depend on accurate detection of keypoints, reliable and unique properties of the descriptors, and the feature matching algorithm itself. Especially in real-time applications, such as VO, a trade-off between processing time and accuracy has to be found. Therefore, reliable feature matching quality assessment is crucial.

1.1. Problem Description

A series of useful GT datasets exist to assess the accuracy of feature detection and matching algorithms. Some are explicitly provided for evaluation of feature matching, some address more general applications (see Section 2 for a detailed description). GT is provided by either a global image transform (*e.g.* homography) or spacial displacements of the pixels (optical flow or disparity). Generating GT feature matches faces two main challenges which can lead to ambiguity. First, it is not guaranteed that for each keypoint in one image a corresponding keypoint in an other image exists. Even if the GT provides a potential keypoint location, due to *e.g.* perspective distortion, occlusions or dynamic areas, a keypoint must not necessarily be found (or exist) there and, as a consequence, a false correspondence could be established. To overcome these challenges, often a threshold is used to define a neighbourhood for valid keypoints around the GT location (see Section 2). This, however, imposes a bias and hinders identification of true negatives (TN). Second, the accuracy of GT itself is limited as only synthetic

*This work was funded by the Austrian Research Promotion Agency (FFG) projects FarmDrive and AnyView3D (#849909 & #853261).

datasets enable a highly accurate identification of matching positions.

1.2. Contribution

We propose a novel method for performance evaluation of feature matching with the focus on, but not limited to, real-time applications such as VO or VSLAM. This includes the generation of highly accurate ground truth matches (publicly available at `vitro-testing.com/test-data/gtm`) from public benchmark datasets. To summarize, this paper contributes to the research field in several ways:

- A framework for generating ground truth matches is presented and applied on several well known datasets (Section 3).
- Our ground truth matches allow evaluation of descriptors and matchers not only with precision and recall but also with accuracy $ACC = (TP+TN)/(P+N)$ and fall-out $FPR = FP/(FP+TN)$.[1] In addition, an arbitrary inlier ratio can be set and the ambiguity of correspondences using only the original GT data is significantly reduced (Sections 3.1 & 4).
- We present a run-time and quality evaluation of multiple state-of-the-art keypoint-descriptor combinations as well as matching algorithms using the above metrics and various inlier ratios (Sections 4.4 & 4.5).
- To prove the accuracy and unambiguousness of our GTM, we manually annotated matches on the synthetic dataset Sintel [13] and on the real-world datasets KITTI [47, 21], Oxford introduced by Mikolajczyk *et al*. [49, 50], and HCI Training [29] (Sections 4.1 & 4.3).
- Finally, using our annotations, we provide the ground truth accuracy of the mentioned datasets (Section 4.2).

2. Related Work

Several frameworks for evaluating the performance of descriptors and matching algorithms exist. Mikolajczyk and Schmid [49] perform an evaluation of local descriptors. They use their own dataset [50, 49] which provides ground truth homographies to calculate $1 - precision$ and recall. A threshold of 50% on the overlap errors of the image areas used by the descriptors is applied for the calculation of these metrics and the correct correspondences. This threshold leads to a bias as can be seen in their paper where they calculate recall and the number of correct matches for different thresholds. Heinly *et al*. [23] evaluate keypoint detectors and binary descriptors using SIFT [38] and SURF [7] as a baseline. Descriptors are evaluated by runtime, putative match ratio, matching score, precision, and recall. The tests are performed on multiple datasets (Oxford [50], Strecha *et al*. [64], and their own), which feature different properties like blur, illumination, and different transformations. All datasets used provide GT information like homographies, 3D, or epipolar geometry. Moreover, they analyze the performance of different detector-descriptor pairings. To identify TP, FP, and FN, necessary to calculate their metrics, a fixed threshold of 2.5 pixels is used, which can lead to a bias in the calculated parameters. Miksik and Mikolajczyk [51] evaluate the runtime of local detectors, descriptors, and the matching process in addition to precision and recall. The experiments are carried out on the Oxford dataset [50] while using queries of all image sequences together for matching. In addition, they add features from a different dataset to simulate a lower inlier ratio. To determine GT correspondences they use the same procedure as in [49]. Furthermore, they analyzed the runtime of real valued descriptors (*e.g*. SIFT and SURF) utilizing a multiple randomized KD-tree and compared it to the execution time of sequential searches with binary descriptors. Johannson *et al*. [27] test different detector-descriptor combinations on infrared (IR) datasets. For testing, they generated a new IR image dataset that includes various deformations like view-point, scale, rotation, blur, noise, and downsampling. The results based on recall, $1 - precision$, and matching score were obtained utilizing the code from [50], which uses a threshold on the overlap error. A quite interesting dataset and benchmark utility for descriptor evaluation was introduced by Lenc *et al*. [34]. They extract patches from various datasets using positions of a combination of local keypoint detectors and an estimated GT homography. Before a patch is extracted, they apply a small amount of affine jitter to the position from where the patch is extracted to simulate the geometric repeatability error of typical local feature detectors. For benchmarking, a descriptor has to be calculated for every patch. These are evaluated on different sets of patches. Three different evaluation scenarios are addressed: image-to-many-image, image-to-image, and patch classification. For the first and second scenario, the ability of a patch descriptor to retrieve corresponding patches is measured with mean average precision (mAP) of the descriptor distances. The last scenario evaluates the ability of a patch descriptor to discriminate pairs of patches that are in correspondence from non-corresponding ones using a threshold on the descriptor distance. These results are represented using the receiver operating characteristic (ROC) [18] in addition to precision and recall. Madeo and Bober [43] analyze the performance of various binary and a few real valued descriptors with fo-

[1] *Precision* $= TP/(TP+FP)$ and *recall* $= TP/(TP+FN)$ with true positives TP, false positives FP, false negatives FN, positives $P = TP + FN$, and negatives $N = TN + FP$.

cus on mobile applications. They use four different scenarios to test descriptors. Two of them are based on image patches [14, 70, 25, 58], one on the Oxford dataset [50], and one is based on their own recorded test set. The results are represented using recall, Kullback-Leibler (KL) divergence for matching and non-matching Hamming distance distributions, and mAP on a descriptor distance weight based on the ratio of descriptor distances from the closest and second closest match. Moreover, they analyzed the runtime of the evaluated binary descriptors and the behaviour on the results for varying ratio thresholds of the ratio test [38].

In addition to the above mentioned datasets, several important datasets for testing various algorithms in the field of autonomous navigation or vision-based driver assistance systems exist. As many of these algorithms are based on a matching pipeline using salient features, an evaluation on features using these datasets is important. A few representatives of the available datasets are KITTI [47, 21], Leuven [32], HCI-Robust [30], Stixel [57], HCI Training [29], and Oxford RobotCar [42]. Some of these are used for evaluations in this paper. Cordes *et al*. [16, 17] performed an evaluation on the accuracy of the GT data of the Oxford dataset [50] and provide more accurate GT information for it. Furthermore, an additional dataset with GT information is provided.

For specific applications not only the right choice of a descriptor but also of a matching algorithm is important. Thus, Maier *et al*. [44] test various matching algorithms for vision-based applications, such as Guided Matching based on Statistical Optical Flow (GMbSOF) [44], CasHash [15], hierarchical clustering tree [52], priority search k-means tree [53], SparseVFC [40, 41] in combination with the hierarchical clustering tree, linear matching and Locality Sensitive Hashing [4] from the FLANN library [53], as well as the randomized KD-tree [62]. For testing, they used the KITTI disparity and flow datasets [47] in addition to the Oxford dataset [50] on SIFT features [37, 38] and FAST keypoints [59] with FREAK descriptors [1]. After matching, they perform a ratio test on the results, which are provided using *ACC*, precision, recall, and *FPR* over various inlier ratios. Moreover, they provide results on the runtime of the different matching algorithms based on the same datasets. Bernhardsson [9] provides a popular Approximate Nearest Neighbour (ANN) benchmark suite to test the performance of ANN matchers. For testing, one thousand randomly extracted queries from a dataset are searched. The results are provided in the form of recall and average query time. Within this benchmark, Malkov and Yashunin [45] are using SIFT [26], GloVe [56], CoPhIR [10], and MNIST [33], in addition to their own generated dataset. They compared the following ANN matchers: Hirarchical Navigable Small World graphs (HNSW) [45], FLANN [53], Annoy [8], VP-tree [11], and FALCONN [5].

3. Evaluation Framework

For testing the performance of descriptors and matching algorithms based on different keypoint detectors, we generate *Ground Truth Matches* (GTM) with reduced correspondence ambiguity from well known datasets that provide GT information (Section 3.1). Besides that, GTM should also cover the weaknesses of detectors, like repeatability errors, for testing applications that depend on keypoint locations and, thus, on the properties of detectors. We use HCI Training 1K flow [29], Oxford [50], and KITTI flow & disparity [47, 21] datasets. KITTI GT is quite sparse, due to the fact that they were created using laser range data. Therefore, a pre-processing step was necessary to allow for a meaningful evaluation. Similar to Maier *et al*. [44], it fills as many invalid GT pixels (where no laser range data was available) as possible using information of the neighbouring pixels.

To proof the accuracy of our generated GTM (Section 4.1), we developed an annotation framework (Section 3.2) and manually annotated the correspondences within the image pairs.

3.1. Ground Truth Matches

We are tackling the challenges of creating unambiguous GTM (mentioned in Section 1.1) in two ways: First, instead of using a fixed distance threshold for defining a neighbourhood (which could result in missing true correspondences while accepting wrong ones), we estimate a tailored threshold for each image pair separately depending on provided GT information and the used keypoint detector. Second, we perform one forward and multiple backward searches in order to identify correct correspondences and true negatives. As a result, using our GTM requires no additional threshold since the correct correspondences are already provided. Furthermore, since our GTM contain true negatives, an arbitrary inlier ratio can be generated.

Threshold estimation: We extract keypoints in both images and search for correspondences using a nearest neighbour algorithm with an initial search radius of 10 pixels (corresponds to a maximal displacement error of ~5% of the upper bound of the KITTI spacial GT magnitudes). From keypoint positions $\mathbf{x}_i^L$ in the left image we calculate search positions $\tilde{\mathbf{x}}_i^R$ in the right image with $\tilde{\mathbf{x}}_i^R = H\mathbf{x}_i^L$ for homographies H or $\tilde{\mathbf{x}}_i^R = \mathbf{x}_i^L + \mathbf{f}_i$ for flow/disparity $\mathbf{f}_i$. For every nearest keypoint at position $\mathbf{x}_i^R,\ i \in [1, \ldots, n]$ of all n possible correspondences, the spatial distances $\mathbf{d}_e = [e_1, \ldots, e_n]$ with $e_i = \left\|\tilde{\mathbf{x}}_i^R - \mathbf{x}_i^R\right\|$ are calculated. To increase robustness, the upper 20% of distances $\mathbf{d}_e$ are ignored for calculating the median distance $\tilde{d}_e$ and its median absolute deviation $\tilde{\sigma}_e$ which are used to reject distances $e_i > \tilde{d}_e + 3.5\tilde{\sigma}_e$. The highest remaining distance equals the radius t_d which encircles the most reasonable candidates (min. $t_d = 2$). For the used datasets, the threshold t_d lies be-

tween 2 and 6.41 pixels. Statistics on t_d for specific datasets can be found in the supplementary material.

Now we address the core problem of finding unambiguous one-to-one matches within those most reasonable candidates.

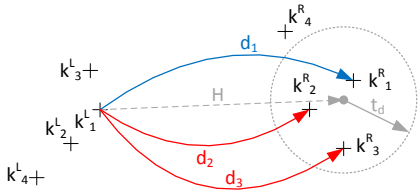
Forward search: For every left feature k_i^L, the right feature candidates $k_{i,j}^R,\ j \in [1,\dots,n_i]$ at $\tilde{\mathbf{x}}_i^R$ within the radius t_d are searched, as can be seen in Figure 1(a). In the case of flow/disparity GT, features at positions without existing GT are rejected. The match with the smallest descriptor distance $d_{i,1}$ within t_d is called best potential match. We first ensure global uniqueness by discarding all left features with the same best potential match. We then ensure local uniqueness by performing a ratio test inspired by Lowe [38] and apply a threshold t_s on the descriptor distance if binary descriptors are used. In detail, if $d_{i,1} > t_s$ or $1.5d_{i,1} > d_{i,j},\ j \in [2,\dots,n_i]$, the feature k_i^L is rejected (*e.g.* row 1 in Table 1). We use $t_s = \alpha_s b$ with binary descriptor size b in bits and $0 \leq \alpha_s \leq 1$. Empirically, we found $\alpha_s = 0.3125$ to be the best.

The forward search is only used to reject left features, as the right features could match different left features or could serve as TN.

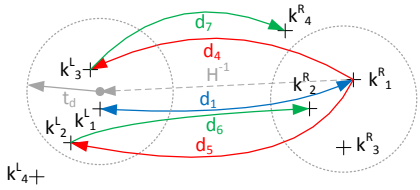
Backward search: To guarantee uniqueness to nearby correspondences that might be within the local neighbourhood but outside the search radius t_d of a best potential match (*e.g.* d_4 and d_7 in Figure 1(b)), we perform an enhanced backward search for each matching candidate $k_{i,j}^R$ within t_d (*e.g.* 3 keypoints in Figure 1(a)) of all remaining left features k_i^L. The neighbourhood is defined by the inverse homography H^{-1} or an interpolated inverse of the flow/disparity $\mathbf{f}$ and t_d. To ensure uniqueness of corresponding features also in the left image, additional ratio tests are performed after the backward search (*e.g.* row 3 in Table 1). We then perform further ratio tests on the found left neighbours and their best potential matches (*e.g.* rows 4-5 in Table 1, green arrows in Figure 1(b)). Assuming that for a small spatial neighbourhood the (affine) distortions between corresponding image regions are quite similar, the descriptor distances should be as well. Thus, the descriptor distances of the best matching features within a small spatial neighbourhood were limited to a dynamic maximum value (*e.g.* rows 6-8 in Table 1) to reject possible false matches. This proofed to be an effective method to reject outliers.

The whole search procedure is repeated until the number of features converges.

Figure 1 shows an example of the forward and one backward search, while Table 1 lists the actual conditions which are used to reject features. For simplicity in Figure 1 and Table 1, the first index of right features k^R and descriptor distances d was skipped. The backward searches for the remaining neighbours can be found in the supplementary material.



(a) Forward search



(b) 1st neighbour backward search

Figure 1. Example of removing the match ambiguity for a single correspondence using forward and backward search. k^L denotes left and k^R right features with descriptor distances d. t_d corresponds to the spatial search radius at the position calculated by the ground truth H. The blue arrows indicate the match under test (best potential match). The red arrows point to spatial nearest neighbours for which their descriptor distance to the query is not the smallest and the green arrows indicate best potential matches of other left features.

Corr. Fig.	#	Condition	Rej. features
1(a)	1	$d_1 > t_s \vee d_2 < 1.5d_1 > d_3$	k_1^L
1(b)	2	$k_1^L \to k_1^R \wedge k_1^R \to k_i^L, i \neq 1$	k_1^R
	3	$d_4 < 1.5d_1 > d_5$	k_1^L, k_1^R
	4	$d_4 < 1.5d_7$	k_3^L, k_4^R
	5	$d_5 < 1.5d_6$	k_2^L, k_2^R
	6	$d_1 < 1.25 \min(d_1, d_6, d_7)$	k_1^L
	7	$d_6 < 1.25 \min(d_1, d_6, d_7)$	k_2^L
	8	$d_7 < 1.25 \min(d_1, d_6, d_7)$	k_3^L

Table 1. Conditions to reject features for a single correspondence illustrated in Figure 1.

GTM are publicly available for several datasets at `vitro-testing.com/test-data/gtm`. Our focus is not providing an automatic benchmarking framework, but providing a possible solution to test algorithms with consistent GT to directly compute various well known performance metrics.

3.2. Correspondence Annotation

For evaluation of the GTM and datasets in general, we implemented a generic annotation framework for flow, stereo, and homography-based GT datasets. The generated GTM using FAST and SIFT keypoints as well as the original dataset GT are evaluated by applying local patch matching constraints. If the strict constraints are not met, manual user-based annotation is used as fall-back. The framework offers guidance and support for this manual annotation task by providing a clean interface for correspondence annotations and showing difference images of both the original

Annot. Type	$\tilde{d}_a$	$\bar{d}_a$	σ_e	Q_1	Q_3	min.	max.	p	n_s
Overall	**0.095**	0.194	0.304	0.050	0.209	0	5.828	0.004	5140
Automatic	0.081	0.149	0.216	0.045	0.160	0.001	3.534	-	4275
Manual	0.257	0.419	0.509	0.123	0.523	0	5.828	-	865
GTM Acc.	0.508	0.624	0.457	0.317	0.795	0	5.946	0.004	5140

Table 2. Accuracy of annotations measured using the synthetic dataset Sintel [13].

patches around the initial hypothesis as well as versions with histogram equalization applied to the patch regions. To enable sub-pixel accuracy for manual annotation, we upscaled the patches and provided the user with functionality for local refinement to achieve accuracies lower than 0.1 pixels. Locality is enforced and perspective/parallax errors are reduced by comparing small-sized patches (48x48 pixels) that are warped by locally estimated or GT (if available) homographies.

Annotating all available GTM of a dataset is unfeasible. Thus, we are estimating a minimal number of samples n_{min} that are necessary to get a meaningful result. It is estimated using the well known equation [31] for a representative sample size

$$n_{min} = \frac{\frac{z^2 p(1-p)}{e_s^2}}{1 + \frac{z^2 p(1-p)}{e_s^2 n_o}}, \quad (1)$$

with the number of GTM n_o within the entire dataset, an error range e_s, estimated error ratio p, and the statistical standard score z. In addition, a minimum of 10 samples was annotated within one image pair. We used $z = 1.96$ which corresponds to a confidence level of 95%. As the error ratio p is not known in the beginning, it is set to a worst case value of $p = 0.5$ and updated for the first time after a few annotations n_a depending on the size n_o. Further updates on p are performed after the annotation of every single image pair. The error range e_s was set according to p. For $p \geq 0.02$, we used an error range of $e_s = 0.01$ and otherwise $e_s = p/2$. We treat a match as defective, if the distance from the annotated position to the keypoint position of a match is larger than t_d used for generating the GTM. Thus, the number of defective matches n_m is used to estimate $p = n_m/n_a$. The typically used sample size is $n_s \approx 2.5 n_{min}$ and the samples are chosen randomly.

4. Results

In this section, we present results on the precision of annotations (Section 4.1), an accuracy evaluation of public available datasets (Section 4.2), an evaluation of the accuracy and unambiguousness of the GTM (Section 4.3), and a performance and runtime analysis of state-of-the-art keypoint-descriptor combinations (Section 4.4) and various state-of-the-art matching algorithms (Section 4.5) in terms of *ACC*, precision, recall, and *FPR* over different inlier ratios ranging from 1% to 100% using our GTM.

Dataset	Detector	GTM Descr.	$\tilde{d}_a^{GT}$	max.	Occ./‰	p	n_s
HCI F.	FAST	FREAK	0.248	21.637	0.41	0.006	17150
KITTI 15 F.	FAST	FREAK	0.475	15.409	0.84	0.022	2373
	SIFT	FREAK	0.613	33.711	2.61	0.011	2678
	SIFT	SIFT	0.663	23.569	7.60	0.044	2763
KITTI 15 D.	FAST	FREAK	0.700	18.561	6.26	0.029	2556
	SIFT	FREAK	0.859	67.795	2.55	0.024	2354
KITTI 12 D.	FAST	FREAK	0.485	63.249	5.08	0.022	2361
Oxford wall	FAST	FREAK	0.966	27.939	1.69	0.044	1779
	SIFT	FREAK	1.103	33.310	1.23	0.014	2434
	SIFT	SIFT	1.234	57.187	7.27	0.106	3025
Oxford graff.	FAST	FREAK	0.597	10.662	5.65	0.048	2300
Oxford bark	FAST	FREAK	1.576	24.779	0.47	0.171	4251

Table 3. GT accuracy of various datasets. The median $\tilde{d}_a^{GT}$ and maximum distances from the annotated positions to the GT were calculated. Occ. specifies the fraction of occluded matches that could not be annotated. p specifies the error ratio described in Section 3.2 and n_s the number of randomly selected samples. Additional results can be found in the supplementary material.

4.1. Accuracy of Annotations

To check the accuracy of the annotation itself, we annotated the synthetic dataset Sintel [13] using the FAST detector and achieved accuracies typically below 0.1 pixels (median). Details are listed in Table 2 using median $\tilde{d}_a$, mean $\bar{d}_a$, standard deviation σ_e, lower quartile Q_1, upper quartile Q_3, minimum, and maximum distances from the annotated positions to the synthetic GT. Moreover, the automatic annotation achieves higher accuracy than the manual annotation which confirms the applicability of the used local patch matching constraints. The small error ratio p (see Table 2) stems from 20 manual annotations for which the annotation error was above the threshold t_d. A few of such incorrect annotations are also part of the annotation results described in Section 4.2, but the larger portion contributing to the value of p is caused by errors of the underlying original GT (like KITTI). Table 2 also lists the accuracy of the GTM which is typically below 1 pixel (median of 0.5 pixels). This inaccuracy is mainly caused by small repeatability errors of the detector and the missing sub-pixel accuracy of the FAST detector. For the Sintel dataset, not a single wrong match of the GTM was found as the underlying GT has no errors. This shows the effectiveness of our algorithm to generate unambiguous GTM.

4.2. Ground Truth Accuracy

We evaluated the accuracy of multiple existing datasets using the annotation framework: KITTI 2012 [21], KITTI 2015 [47], Oxford [50], and HCI Training 1K flow [29] dataset. KITTI and Oxford were chosen due to their popularity and HCI represents a new dataset with a high test case coverage. KITTI and HCI provide, among others, GT for stereo and flow, Oxford provides homographies.

Table 3 summarizes key statistical accuracy values from the evaluation and Figure 2 shows histograms of the measured error distributions.

All datasets offer GT with median errors $\tilde{d}_a^{GT} < 1.6$ pix-

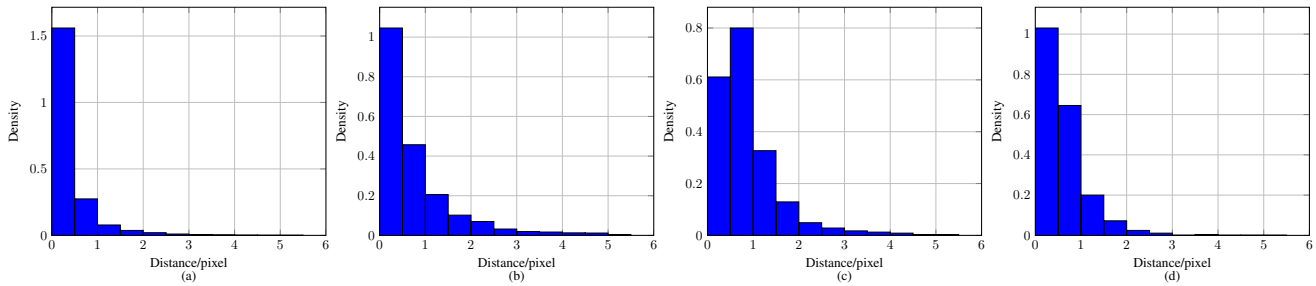


Figure 2. Density over the pixel distances from annotated positions to the GT (error) for (a) HCI flow, (b) KITTI 2015 flow, (c) KITTI 2015 disparity, and (d) KITTI 2012 disparity. The annotated positions are based on randomly selected matches of the GTM using FAST keypoints. Additional results can be found in the supplementary material.

els. The typical accuracies of KITTI datasets are better than 1 pixel and the HCI dataset even achieves $1/4$ of a pixel although its resolution is much larger offering more than six times the number of pixels per frame as compared to the KITTI datasets.

The Sintel accuracy test in Section 4.1 shows that the used annotation framework introduces a median error of about 0.1 pixels for automatic annotations and 0.25 for manual annotations. A larger portion of all annotations were done automatically so that all median error values are more than two times as high as the annotation's framework accuracy thus creating a reliable comparison of the evaluated GT.

4.3. GTM accuracy

To test the accuracy of our GTM, we used the presented annotation framework of Section 3.2. The annotations were performed using the datasets KITTI 2015 flow and disparity [47], KITTI 2012 disparity [21], and the sequences "wall", "graffity" and "bark" from the Oxford dataset [50]. Table 4 shows the results for these annotations. Compared to the median distances $\tilde{d}_a^{GT}$ from annotated positions to the GT (Section 4.2), the GTM accuracy $\tilde{d}_a^{GTM}$ is approximately within the same range. The difference $\Delta\tilde{d}_a = \tilde{d}_a^{GTM} - \tilde{d}_a^{GT}$ of the GTM median error $\tilde{d}_a^{GTM}$ to the GT median error $\tilde{d}_a^{GT}$ of annotations using FAST keypoints is slightly worse for KITTI datasets than for annotations using SIFT features as can be seen in Table 4. This can be traced back to a better repeatability of the SIFT detector. For the Oxford dataset, on the contrary, typically a lower error is achieved using the keypoints from the GTM compared to the underlying GT. This is caused by the relatively large error of the GT, which is often compensated by the GTM as a result of the filtering procedures described in Section 3.1. For GT errors $d_a^{GTM} > t_d$, most matches are correctly rejected as their discriptor distances are too large. Some, however, remain classified as good matches. To verify the matchability of such matches, their precision, which is calculated using SIFT descriptors and sequential search, is compared to those with $d_a^{GTM} \leq \check{t}_d$ and $\check{t}_d = \min(t_d, 4)$. In general, the precision decreases with higher distance to the annotated positions but does not drastically drop to zero (*i.e.* many of these matches are still matchable) taking into account the extremely reduced number of GTM for $d_a^{GTM} > \check{t}_d$. Table 4 lists the calculated precisions of matches grouped by their GT errors d_a^{GTM} as well as the precision of occluded matches. This confirms the enforced uniqueness for GTM which also holds for most false matches that are a result of inaccurate original GT.

				Matching precision using SIFT descriptors			
Dataset	Detector	GTM Descr.	$\Delta\tilde{d}_a$	$d_a^{GTM} < \check{t}_d$	$4 < d_a^{GTM}, d_a^{GTM} \leq 8$	$d_a^{GTM} > 8$	Occ.
KITTI 15 F.	FAST	FREAK	0.154	0.761	0.476	0.333	0
	SIFT	FREAK	-0.037	0.848	0.154	0	0.143
	SIFT	SIFT	0.005	0.836	0.200	0.238	0.143
KITTI 15 D.	FAST	FREAK	0.102	0.790	0.636	0.333	0.313
	SIFT	FREAK	-0.048	0.798	0.250	0.222	0.500
KITTI 12 D.	FAST	FREAK	0.205	0.788	0.500	0.333	0.417
Oxford wall	FAST	FREAK	-0.419	0.764	0.500	0	0
	SIFT	FREAK	-0.579	0.759	0	0	0
	SIFT	SIFT	-0.512	0.766	0	0.016	0
Oxford graff.	FAST	FREAK	0.483	0.529	0.228	0.2	0
Oxford bark	FAST	FREAK	-0.387	0.245	0.064	0.016	0

Table 4. GTM accuracy for various datasets. The precision is calculated for GTM featuring different classes of distances d_a^{GTM}. Additional results can be found in the supplementary material.

To check the influence of the descriptor type for GTM generation, we generated and annotated the GTM using SIFT descriptors in addition to FREAK descriptors for the KITTI 2015 flow and Oxford "wall" datasets. As can be seen in Table 3, the median distances for SIFT descriptors are slightly higher (approx. 0.1 pixel) than for FREAK descriptors. This is within the uncertainty of the annotation described in Section 4.1. Moreover, the fraction of occluded correspondences and the error ratio p (see Table 3) are significantly higher using SIFT descriptors. This is due to the fact that in contrast to using binary descriptors (like FREAK), no thresholding is performed on the SIFT descriptor distances. Thus, for small GT errors, the choice of the descriptor type has no influence on the GTM. To reduce the influence of higher GT errors on the GTM, thresholding on the descriptor distance is favorable, which is performed by default in our GTM generation (see Section 3.1).

4.4. Evaluations on Descriptors

We tested various keypoint-descriptor pairings in terms of *ACC*, precision, recall, and *FPR* in addition to the runtime using our GTM generated on the datasets KITTI 2015 flow & disparity [47] and Oxford [50]. For matching we used the linear matching algorithm from the FLANN library [53]. Evaluations are performed using the following descriptors: KAZE [2], AKAZE [3], BOLD [6], BRISK [35], DAISY [65], FREAK [1], LATCH [36], ORB [60], RIFF [71], SIFT [38], BGM-Bilinear [69], BGM-Hard [69], BGM [69], LBGM [69], BinBoost [67, 68] with a descriptor size of 64 bits, 128 bits, and 256 bits, in addition to the VGG descriptor [63] with a descriptor size of 48 bits, 64 bits, 80 bits, and 120 bits. These descriptors are tested using the following keypoint detectors: SIFT [38], KAZE [2], AKAZE [3], MSD [66], FAST [59], BRISK [35], and ORB [60]. We used the OpenCV [12] implementations with default configurations of all mentioned detectors and descriptors except RIFF and BOLD. For those, the code provided by the authors was used. For BGM, BGM-Hard, BGM-Bilinear, LBGM, and all variants of BinBoost and VGG the parameters were tuned according to the keypoint type.

Descriptor Quality: We tested all possible combinations of keypoints and descriptors mentioned above. Unfortunately, some descriptors are incompatible with a few keypoints (see Table 5). All descriptors are tested with varying inlier ratios ranging from 1% to 100%. For each inlier ratio and dataset the average *ACC*, precision, recall, and *FPR* over all image pairs in a dataset is estimated. For evaluations on the Oxford dataset [50] all possible image pair combinations (15) are used.

Table 5 shows the mean *ACC* for every keypoint-descriptor combination. For every value shown in Table 5, the average is calculated over the mean *ACC* of every tested inlier ratio and the datasets KITTI 2015 flow and disparity [47] in addition to the sequences "bark", "bikes", "boat", "graffiti", "JPEG", "light", and "wall" of the Oxford dataset [50]. We are using the mean *ACC* values in Table 5, as *ACC* enables to quantify the closeness of an algorithm's output to the true solution. For most descriptors, SIFT keypoints are the best choice in terms of *ACC* as can be seen in Table 5. For a high recall, AKAZE keypoints deliver better results and for a high precision MSD or SIFT keypoints should be used. Moreover, the choice of the keypoint detector and descriptor depends on the underlying data. For KITTI flow and disparity [47], AKAZE keypoints deliver the best results for most descriptors in terms of *ACC* and recall. Our evaluations showed that FAST and ORB keypoints perform worst independent of the used dataset. One exception is the ORB descriptor which performs best with ORB keypoints for most datasets.

Descriptor Runtime: The descriptor runtimes are evaluated using the GTM of the first 10 image pairs of KITTI flow [47] with an inlier ratio of 100%. Time measurements for one image pair are performed using the smallest runtime of 20 runs on an Intel Xeon E5-1620 v3 3.5GHz CPU. The runtime of one descriptor t_f is estimated by $t_f = {}^{t_i}/_{(n_f^L + n_f^R)}$ with the descriptor computation time t_i of both, the first and second image and the number of keypoints n_f^L and n_f^R in the first and second image. From the resulting 10 time measurements, statistics (mean $\bar{t}_f$, median $\tilde{t}_f$, min. $\check{t}_f$, max. $\hat{t}_f$) are generated. Table 6 shows the average runtimes $\bar{t}_{\bar{t}}$ over $\check{t}_f$ for all compatible keypoints shown in Table 5.

Time measurements showed that the descriptor computation time is not linear to the number of keypoints for the tested implementations of descriptors ORB, FREAK, RIFF, SIFT, AKAZE, KAZE, BRISK, and DAISY. The temporal behaviour for most of them is best described by $t_i = t_0 + t_f \left(n_f^L + n_f^R\right)$ with a fixed time t_0 depending

Descr. / Keypoint	SIFT	KAZE	AKAZE	MSD	FAST	BRISK	ORB
AKAZE	-	-	**0.714**	-	-	-	-
KAZE	-	**0.734**	-	-	-	-	-
ORB	-	0.557	0.570	0.569	0.576	0.684	**0.731**
LATCH	0.475	0.475	0.553	0.424	0.458	**0.568**	0.568
BOLD	0.576	**0.598**	0.594	0.541	0.598	0.573	0.565
BGM BILINEAR	**0.587**	0.556	0.565	0.470	0.510	0.527	0.547
BGM HARD	**0.652**	0.601	0.614	0.507	0.604	0.575	0.621
BINBOOST 64	**0.674**	0.671	0.665	0.640	0.639	0.649	0.662
BRISK	0.676	0.687	0.691	0.659	0.614	**0.751**	0.683
BINBOOST 256	**0.688**	0.637	0.649	0.583	0.589	0.645	0.649
RIFF	0.709	0.720	0.719	0.720	0.659	**0.761**	0.685
BGM	**0.716**	0.662	0.674	0.603	0.643	0.667	0.694
BINBOOST 128	**0.725**	0.683	0.687	0.620	0.673	0.673	0.698
DAISY	**0.736**	0.703	0.703	0.670	0.700	0.709	0.692
FREAK	0.741	0.732	0.730	0.740	0.687	**0.815**	0.716
LBGM	**0.746**	0.730	0.729	0.720	0.722	0.721	0.739
SIFT	**0.754**	0.745	0.741	-	0.709	0.714	0.534
VGG 48	0.761	0.759	0.757	**0.807**	0.755	0.796	0.773
VGG 64	0.778	0.772	0.765	**0.810**	0.760	0.801	0.781
VGG 80	0.784	0.774	0.767	**0.808**	0.756	0.807	0.784
VGG 120	0.786	0.771	0.764	0.784	0.717	**0.809**	0.776

Table 5. Mean *ACC* of different detector-descriptor combinations. The bold values mark the best performing combination for every descriptor. The colored cells mark the highest, second, and third highest mean descriptor *ACC* for every detector. Some cells are empty due to a incompatible keypoint-descriptor combination. Detailed figures of additional performance metrics over varying inlier ratios can be found in the supplementary material.

Descriptor	$\bar{t}_{\bar{t}}/\mu s$	Descriptor	$\bar{t}_{\bar{t}}/\mu s$	Descriptor	$\bar{t}_{\bar{t}}/\mu s$
BOLD	5.4	BINBOOST 256	53.4	VGG 48	400.1
ORB	14.4	BGM	62.2	BRISK	424.7
BGM HARD	17.9	FREAK	80.2	VGG 64	440.8
BGM BILINEAR	20.0	RIFF	95.5	VGG 80	441.4
BINBOOST 64	36.6	LBGM	98.0	VGG 120	445.5
BINBOOST 128	42.1	AKAZE	151.2	DAISY	566.4
LATCH	47.3	SIFT	390.4	KAZE	613.9

Table 6. Mean over the minimum average descriptor computation times $\bar{t}_{\bar{t}}$ in μs for one descriptor over all compatible keypoints stated in Table 5. Additional results can be found in the supplementary material.

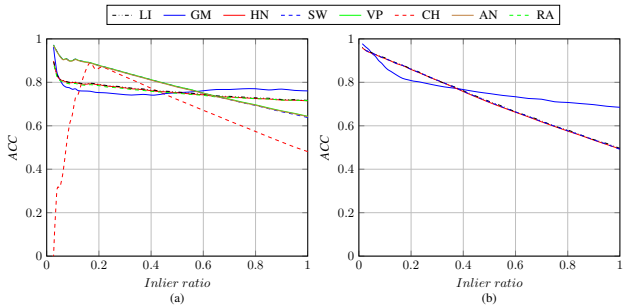


Figure 3. Inlier ratio compared to mean *ACC* using (a) SIFT features and (b) FAST keypoints & FREAK descriptors for the entire KITTI disparity dataset from Menze and Geiger [47]. On the results of all algorithms, a ratio test was performed. Additional results can be found in the supplementary material.

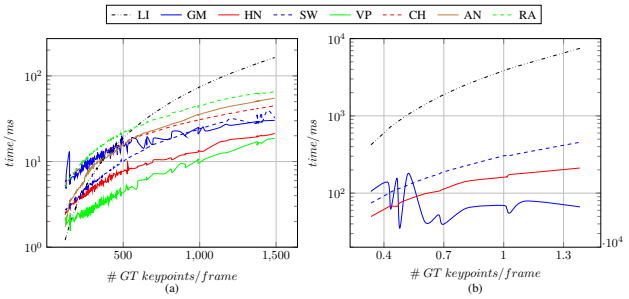


Figure 4. Runtime analysis using (a) SIFT features on the KITTI flow dataset from Menze and Geiger [47] and (b) FAST keypoints & FREAK descriptors on the Oxford "wall" sequence [49, 50]. Time measurements were performed using the smallest runtime of 100 runs on an Intel Xeon E5-1620 v3 3.5GHz CPU. Each datapoint stems from a different image pair with an inlier ratio of 75%. Additional results can be found in the supplementary material.

on the image characteristics and keypoint type. For SIFT, ORB, FREAK and RIFF descriptors $t_0 \gg t_f$. For RIFF, AKAZE, KAZE, BRISK, and DAISY, a high variance in t_f could be observed over the tested image pairs and keypoint detectors.

4.5. Matching Quality & Runtime Evaluations

We tested the following matching algorithms in terms of runtime in addition to *ACC*, precision, recall, and *FPR*: GMbSOF [44] (GM), HNSW [45] (HN) from the NMSLIB [54], linear matching (LI) from the FLANN library [53], Small World Graph [54] (SW), VP-tree [54] (VP), CasHash [15] (CH), ANNOY [9] (AN), and the randomized KD-tree [62] (RA). For testing HN and SW, we tuned them by varying their parameters based on the Oxford "wall" sequence using FAST keypoints & FREAK descriptors in addition to SIFT features.

To asses the matching quality of the different matching algorithms we used the above mentioned performance metrics and the GTM of datasets KITTI 2015 disparity & flow in addition to the Oxford sequences "bark", "boat", "graffiti", and "wall". Figure 3 shows an example of these evaluations. Most matchers support real valued descriptors but only a few, like GM, LI, and HN are capable of matching binary descriptors. HN performs similar to LI in terms of matching quality although it is an ANN matching algorithm. Comparing their runtime (see Figure 4), HN is orders of magnitudes faster than LI. This shows the impressive improvements on ANN algorithms in the last years. GM and HN have similar *ACC* for real valued descriptors but GM outperforms HN using binary descriptors. Typically, GM provides better results in terms of recall but slightly worse results for precision. The reason for this is the completely different matching approach, as GM constrains the search space by utilizing spatial statistics from a small subset of pre-matched and filtered correspondences [44].

Time measurements are performed for the above mentioned matching algorithms and each image pair of the entire KITTI 2015 disparity, flow, and Oxford "wall" datasets separately. An example of the results is shown in Figure 4. GM is fast for a large amount of features but is outperformed by most other matching algorithms for a number of features smaller than 500. For small numbers of features and real valued descriptors, VP is the fastest followed by HN which outperforms VP at higher feature numbers due to a smaller gradient (see Figure 4b). Thus, the right choice of a matching algorithm depends on the underlying data including the expected inlier ratio and if a high recall or precision is desired. For most datasets, GM, HN, and VP give the best results in terms of processing time and matching quality.

5. Conclusion

We presented a method to generate ground truth matches (GTM) based on the original ground truth of well known datasets. The GTM provide unambiguous and correct correspondences in addition to a user specific inlier ratio for testing various vision based algorithms like descriptors or feature matchers. Tests on keypoint-descriptor combinations showed that modern binary descriptors achieve comparable results in terms of quality but significantly lower processing times compared to real valued descriptors. The best results were achieved using SIFT, AKAZE, and MSD keypoints while FAST and ORB performed worst independent of the used dataset. For matching descriptors, GMbSOF, HNSW and the VP-tree achieved the best results in terms of processing time and matching quality but their performance differs depending on the number of features, dataset, and the expected inlier ratio. Annotations showed that the HCI Training 1K flow dataset provides the highest GT accuracy followed by KITTI.

References

[1] A. Alahi, R. Ortiz, and P. Vandergheynst. FREAK: Fast Retina Keypoint. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 510–517, 2012. 3, 7

[2] P. F. Alcantarilla, A. Bartoli, and A. J. Davison. *KAZE Features*, pages 214–227. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. 7

[3] P. F. Alcantarilla, J. Nuevo, and A. Bartoli. Fast explicit diffusion for accelerated features in nonlinear scale spaces. In *BMVC*, 2013. 7

[4] A. Andoni and P. Indyk. Near-optimal Hashing Algorithms for Approximate Nearest Neighbor in High Dimensions. *Communications of the ACM*, 51(1):117–122, 2008. 3

[5] A. Andoni and I. Razenshteyn. Optimal data-dependent hashing for approximate near neighbors. In *Proceedings of the Forty-seventh Annual ACM Symposium on Theory of Computing*, STOC '15, pages 793–801, New York, NY, USA, 2015. ACM. 3

[6] V. Balntas, L. Tang, and K. Mikolajczyk. Bold - binary online learned descriptor for efficient image matching. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2367–2375, June 2015. 7

[7] H. Bay, T. Tuytelaars, and L. Van Gool. *SURF: Speeded Up Robust Features*, pages 404–417. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006. 2

[8] E. Bernhardsson. Annoy: Approximate nearest nneighbor in c++/python optimized for memory usage and loading/saving to disk, 2016. Available at https://github.com/spotify/annoy (Date last accessed 10-March-2017). 3

[9] E. Bernhardsson. Approximate nearest neighbor benchmarks, 2016. Available at https://github.com/erikbern/ann-benchmarks (Date last accessed 10-March-2017). 3, 8

[10] P. Bolettieri, A. Esuli, F. Falchi, C. Lucchese, R. Perego, T. Piccioli, and F. Rabitti. Cophir: A test collection for content-based image retrieval. *CoRR*, abs/0905.4627, 2009. 3

[11] L. Boytsov and B. Naidan. *Engineering Efficient and Effective Non-metric Space Library*, pages 280–293. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. 3

[12] G. Bradski. The opencv library. *Dr. Dobb's Journal of Software Tools*, 2000. 7

[13] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. *A Naturalistic Open Source Movie for Optical Flow Evaluation*, pages 611–625. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. 2, 5

[14] V. Chandrasekhar, G. Takacs, D. M. Chen, S. S. Tsai, M. Makar, and B. Girod. Feature matching performance of compact descriptors for visual search. In *Data Compression Conference*, pages 3–12, March 2014. 3

[15] J. Cheng, C. Leng, J. Wu, H. Cui, and H. Lu. Fast and Accurate Image Matching with Cascade Hashing for 3D Reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2014. 3, 8

[16] K. Cordes, B. Rosenhahn, and J. Ostermann. Increasing the accuracy of feature evaluation benchmarks using differential evolution. In *2011 IEEE Symposium on Differential Evolution (SDE)*, pages 1–8, April 2011. 3

[17] K. Cordes, B. Rosenhahn, and J. Ostermann. *High-Resolution Feature Evaluation Benchmark*, pages 327–334. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. 3

[18] J. Davis and M. Goadrich. The Relationship Between Precision-Recall and ROC Curves. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 233–240, New York, NY, USA, 2006. ACM. 2

[19] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2010. 1

[20] C. Forster, M. Pizzoli, and D. Scaramuzza. Svo: Fast semi-direct monocular visual odometry. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 15–22. IEEE, 2014. 1

[21] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3354–3361, June 2012. 2, 3, 5, 6

[22] A. Geiger, J. Ziegler, and C. Stiller. StereoScan: Dense 3d reconstruction in real-time. In *IEEE Intelligent Vehicles Symposium*, pages 963–968, 2011. 1

[23] J. Heinly, E. Dunn, and J.-M. Frahm. *Comparative Evaluation of Binary Features*, pages 759–773. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. 2

[24] J. Heinly, J. L. Schonberger, E. Dunn, and J.-M. Frahm. Reconstructing the world* in six days*(as captured by the yahoo 100 million image dataset). In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3287–3295, 2015. 1

[25] H. Jegou, M. Douze, and C. Schmid. *Hamming Embedding and Weak Geometric Consistency for Large Scale Image Search*, pages 304–317. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. 3

[26] H. Jegou, M. Douze, and C. Schmid. Product quantization for nearest neighbor search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(1):117–128, January 2011. 3

[27] J. Johansson, M. Solli, and A. Maki. *An Evaluation of Local Feature Detectors and Descriptors for Infrared Images*, pages 711–723. Springer International Publishing, 2016. 2

[28] A. Kendall, M. Grimes, and R. Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE international conference on computer vision*, pages 2938–2946, 2015. 1

[29] D. Kondermann, R. Nair, K. Honauer, K. Krispin, J. Andrulis, A. Brock, B. Güssefeld, M. Rahimimoghaddam, S. Hofmann, C. Brenner, and B. Jähne. The hci benchmark suite: Stereo and flow ground truth with uncertainties for urban autonomous driving. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 19–28, June 2016. 2, 3, 5

[30] D. Kondermann, A. Sellent, B. Jähne, and J. Wingbermühle. Robust vision challenge, 2012. 3

[31] R. V. Krejcie and D. W. Morgan. Determining sample size for research activities. *Educational and Psychological Measurement*, 30(3):607–610, 1970. 5

[32] L. Ladický, P. Sturgess, C. Russell, S. Sengupta, Y. Bastanlar, W. Clocksin, and P. H. S. Torr. Joint optimization for object class segmentation and dense stereo reconstruction. *International Journal of Computer Vision*, 100(2):122–133, 2012. 3

[33] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998. 3

[34] K. Lenc, V. Balntas, and A. Vedaldi. Local Features: State of the Art, Open Problems and Performance Evaluation. ECCV workshop, 2016. Available at http://www.iis.ee.ic.ac.uk/ComputerVision/DescrWorkshop/index.html (Date last accessed 09-March-2017). 2

[35] S. Leutenegger, M. Chli, and R. Y. Siegwart. Brisk: Binary robust invariant scalable keypoints. In *International Conference on Computer Vision*, pages 2548–2555, November 2011. 7

[36] G. Levi and T. Hassner. LATCH: learned arrangements of three patch codes. *CoRR*, abs/1501.03719, 2015. 7

[37] D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157, 1999. 3

[38] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 2, 3, 4, 7

[39] G. Lu, Y. Yan, L. Ren, J. Song, N. Sebe, and C. Kambhamettu. Localize me anywhere, anytime: a multi-task point-retrieval approach. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2434–2442, 2015. 1

[40] J. Ma, J. Zhao, J. Tian, A. L. Yuille, and Z. Tu. Robust Point Matching via Vector Field Consensus. *IEEE Transactions on Image Processing*, 23(4):1706–1721, 2014. 3

[41] J. Ma, H. Zhou, J. Zhao, Y. Gao, J. Jiang, and J. Tian. Robust Feature Matching for Remote Sensing Image Registration via Locally Linear Transforming. *IEEE Transactions on Geoscience and Remote Sensing*, 53(12):6469–6481, 2015. 3

[42] W. Maddern, G. Pascoe, C. Linegar, and P. Newman. 1 Year, 1000km: The Oxford RobotCar Dataset. *The International Journal of Robotics Research (IJRR)*, 36(1):3–15, 2017. 3

[43] S. Madeo and M. Bober. Fast, Compact, and Discriminative: Evaluation of Binary Descriptors for Mobile Applications. *IEEE Transactions on Multimedia*, 19(2):221–235, February 2017. 2

[44] J. Maier, M. Humenberger, M. Murschitz, O. Zendel, and M. Vincze. *Guided Matching Based on Statistical Optical Flow for Fast and Robust Correspondence Analysis*, pages 101–117. Springer International Publishing, 2016. 3, 8

[45] Y. A. Malkov and D. A. Yashunin. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *CoRR*, abs/1603.09320, 2016. 3, 8

[46] E. Marchand, H. Uchiyama, and F. Spindler. Pose estimation for augmented reality: a hands-on survey. *IEEE transactions on visualization and computer graphics*, 22(12):2633–2651, 2016. 1

[47] M. Menze and A. Geiger. Object scene flow for autonomous vehicles. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3061–3070, 2015. 2, 3, 5, 6, 7, 8

[48] S. Middelberg, T. Sattler, O. Untzelmann, and L. Kobbelt. Scalable 6-dof localization on mobile devices. In *European conference on computer vision*, pages 268–283. Springer, 2014. 1

[49] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005. 2, 8

[50] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool. A Comparison of Affine Region Detectors. *International Journal of Computer Vision*, 65(1-2):43–72, 2005. 2, 3, 5, 6, 7, 8

[51] O. Miksik and K. Mikolajczyk. Evaluation of local detectors and descriptors for fast feature matching. In *21st International Conference on Pattern Recognition (ICPR)*, pages 2681–2684, 2012. 2

[52] M. Muja and D. G. Lowe. Fast Matching of Binary Features. In *Proceedings of the 9th Conference on Computer and Robot Vision*, CRV '12, pages 404–410. IEEE Computer Society, 2012. 3

[53] M. Muja and D. G. Lowe. Scalable Nearest Neighbor Algorithms for High Dimensional Data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(11):2227–2240, 2014. 3, 7, 8

[54] B. Naidan and L. Boytsov. Non-metric space library manual. *CoRR*, abs/1508.05470, 2015. 8

[55] D. Nistér, O. Naroditsky, and J. Bergen. Visual odometry. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–I. Ieee, 2004. 1

[56] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543, 2014. 3

[57] D. Pfeiffer, S. Gehrig, and N. Schneider. Exploiting the power of stereo confidences. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 297–304, June 2013. 3

[58] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, June 2007. 3

[59] E. Rosten and T. Drummond. Machine Learning for High-Speed Corner Detection. In *European Conference on Computer Vision (ECCV)*, volume 3951 of *Lecture Notes in Computer Science*, pages 430–443. Springer Berlin Heidelberg, 2006. 3, 7

[60] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: An efficient alternative to sift or surf. In *Proceedings of the International Conference on Computer Vision*, ICCV '11, pages 2564–2571, Washington, DC, USA, 2011. IEEE Computer Society. 7

[61] D. Scaramuzza and F. Fraundorfer. Visual odometry [tutorial]. *IEEE robotics & automation magazine*, 18(4):80–92, 2011. 1

[62] C. Silpa-Anan and R. Hartley. Optimised KD-trees for fast image descriptor matching. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008. 3, 8

[63] K. Simonyan, A. Vedaldi, and A. Zisserman. Learning local feature descriptors using convex optimisation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(8):1573–1585, August 2014. 7

[64] C. Strecha, W. von Hansen, L. V. Gool, P. Fua, and U. Thoennessen. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2008. 2

[65] E. Tola, V. Lepetit, and P. Fua. Daisy: An efficient dense descriptor applied to wide-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(5):815–830, May 2010. 7

[66] F. Tombari and L. D. Stefano. *Interest Points via Maximal Self-Dissimilarities*, pages 586–600. Springer International Publishing, Cham, 2015. 7

[67] T. Trzcinski, M. Christoudias, P. Fua, and V. Lepetit. Boosting binary keypoint descriptors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2874–2881, Washington, DC, USA, 2013. IEEE Computer Society. 7

[68] T. Trzcinski, M. Christoudias, and V. Lepetit. Learning image descriptors with boosting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):597–610, March 2015. 7

[69] T. Trzcinski, M. Christoudias, V. Lepetit, and P. Fua. Learning image descriptors with the boosting-trick. In *Proceedings of the 25th International Conference on Neural Information Processing Systems*, NIPS'12, pages 269–277, USA, 2012. Curran Associates Inc. 7

[70] S. Winder, G. Hua, and M. Brown. Picking the best daisy. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 178–185, June 2009. 3

[71] S. Wu and M. S. Lew. Riff: Retina-inspired invariant fast feature descriptor. In *Proceedings of the 22Nd ACM International Conference on Multimedia*, MM '14, pages 1129–1132, New York, NY, USA, 2014. ACM. 7

[72] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua. Lift: Learned invariant feature transform. In *European Conference on Computer Vision*, pages 467–483. Springer, 2016. 1