Fast Image-Based Localization using Direct 2D-to-3D Matching

Torsten Sattler, Bastian Leibe, Leif Kobbelt
RWTH Aachen University
{tsattler@cs, leibe@umic, kobbelt@cs}.rwth-aachen.de

Abstract

Recently developed Structure from Motion (SfM) reconstruction approaches enable the creation of large scale 3D models of urban scenes. These compact scene representations can then be used for accurate image-based localization, creating the need for localization approaches that are able to efficiently handle such large amounts of data. An important bottleneck is the computation of 2D-to-3D correspondences required for pose estimation. Current state-of-the-art approaches use indirect matching techniques to accelerate this search. In this paper we demonstrate that direct 2D-to-3D matching methods have a considerable potential for improving registration performance. We derive a direct matching framework based on visual vocabulary quantization and a prioritized correspondence search. Through extensive experiments, we show that our framework efficiently handles large datasets and outperforms current state-of-the-art methods.

1. Introduction

Image-based localization is an important problem in computer vision. Its applications include localization and navigation for both pedestrians [22, 31, 13] and robots [6, 5], Augmented Reality [1, 3], and the visualization of photo collections [26]. Image-based localization is also an important part in the pipeline of higher-level computer vision tasks such as semantic object annotation [9] and can be used as an initial pose estimate to speed up large-scale reconstructions from Internet photo collections [27].

Traditionally, large-scale image-based localization has been treated as an image retrieval problem. After finding those images in a database that are most similar to the query image, the location of the query can be determined relative to them [22, 31]. The huge progress achieved in the field of image retrieval enables the use of an increasing number of images for the representation of real world scenes [25, 19, 20]. However, the localization accuracy obtained this way cannot be better than the precision of the GPS positions available for the database images. To achieve a higher localization accuracy, more detailed information is needed which can be obtained from a 3D reconstruction of the scene. Using these models additionally permits to estimate the orientation (and thus the complete pose) of the camera and yields a much more structured representation of the scenes. Recent advances in SfM research [27] now make it possible to construct models on a city-scale level consisting of millions of points in only a few hours [8, 29, 21], creating the need for image-based localization methods that can handle such large datasets.

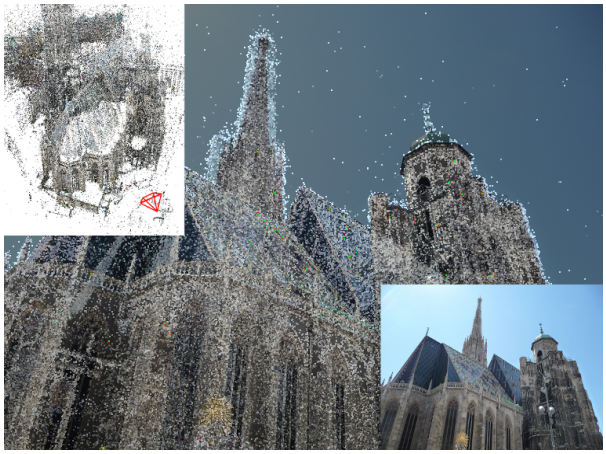

Figure 1: Our approach for image-based localization accurately registers query images (bottom right) to a 3D scene model of an entire city (top left, close-up view) using an efficient 2D-to-3D matching framework.

Essential for image-based localization using 3D models is to establish correspondences between 2D local features in the query image and 3D points in the model. The common approach is to use the feature descriptors, *e.g.* SIFT [17], for the 3D points computed during the reconstruction, formulating the correspondence search as a descriptor matching problem. Following the terminology from [16] we refer to 2D image features and their descriptors as *features* and to 3D points and their descriptors as *points*. We distinguish between direct and indirect 2D-to-3D matching. Direct matching tries to find the 3D point corresponding to a 2D feature by searching for the nearest neighbors of that feature's descriptor in the space containing the 3D point descriptors, while indirect methods use an intermediate construct to represent points and their descriptors which does not preserve the proximity in descriptor space. Classical direct matching approaches such as approximative tree-based

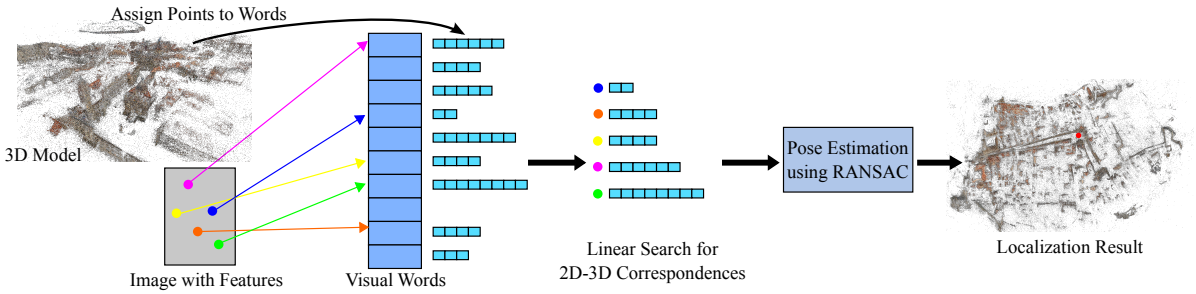

Figure 2: Illustration of our direct 2D-to-3D matching framework consisting of three modular components. By associating 3D points to visual words, we quickly identify possible correspondences for 2D features which are verified in a more detailed linear search. The final 2D-to-3D correspondences are then used to localize the query image using n-point pose estimation.

search [17, 18] provide excellent matching results for directly matching descriptors from the image to descriptors from the model. However, search becomes prohibitively expensive in very large and dense descriptor collections. Recent approaches have therefore proposed indirect matching schemes to handle such huge databases. Examples include techniques that use image retrieval to find similar images which contain a set of possibly matching 3D points (2D-to-2D-to-3D-matching) [13] or that match points from the model to features in the image based on mutual visibility information (3D-to-2D matching) [16]. Although substantially faster than direct tree-based 2D-to-3D matching, those approaches are not yet as effective, as evidenced by the lower number of images that can be registered with them.

In this paper, we show experimentally that 2D-to-3D matching offers considerable potential for improvement over current state-of-the-art localization approaches. Motivated by this result, we propose a framework for efficient direct 2D-to-3D matching based on associating 3D points with a visual vocabulary obtained from clustering feature descriptors, and we explore different association strategies. We limit the search space for each query descriptor to the point descriptors contained in its corresponding visual word, which enables us to estimate that descriptor's search cost. We show how both the registration and rejection times can be improved through a prioritized search which first considers all query features whose visual words lead to cheap correspondences searches. We verify all those findings through extensive experiments and derive useful design guidelines for use in practical applications.[1]

The paper is structured as follows. The remainder of this section discusses related work. Sec. 2 explains the setup for the experiments conducted in this paper. In Sec. 3 we experimentally show the potential of direct 2D-to-3D matching. Based on those results, we derive a framework for efficient direct matching in Sec. 4 and evaluate it in Sec. 5.

Related Work. One of the earliest image localization frameworks has been proposed by Robertson & Cipolla, who estimate the position of images relative to a set of 200 rectified views of facades manually registered onto a city map [22]. In a similar approach, Zhang & Kosecka use SIFT features [17] to match a query image against a set of reference images with known GPS coordinates [31], utilizing the two best matching images to triangulate the position of the camera. Schindler *et al*. achieve image localization on a city-scale, modeled by $30,000$ GPS-annotated images [23], by accelerating image retrieval with a vocabulary tree [19] containing only features unique to their location. In contrast, Zamir & Shah show how to exploit repetitive features found in large urban environments to attain a higher accuracy for the GPS estimates of the query images [30]. To efficiently handle even larger sets of around 1 million geo-tagged images, Avrithis *et al*. combine features from similar images into so-called scene maps, thereby reducing the number of documents to be considered during retrieval [2]. Hays & Efros utilize mean-shift to express the location of an image as a probability distribution over the surface of the Earth, using a database with more than 6 million geo-tagged images for planet-scale localization [11].

Some applications such as robot localization require a higher localization accuracy than the GPS-level precision provided by the approaches described above, which can be achieved using a 3D map of the environment. Se *et al*. simultaneously reconstruct the environment and localize their robot in this map using a stereo camera mounted onto the robot [24]. Modern SLAM methods enable this approach to scale to larger scenes, while still permitting real-time localization and mapping [3, 5, 6]. Irschara *et al*. show that image retrieval techniques can be used to accelerate pose estimation against large 3D models obtained from SfM methods using the 3D points belonging to the features in the retrieved images to establish 2D-to-3D correspondences [13]. The camera pose is then estimated using a 3-point-pose algorithm [7]. Since the use of image retrieval techniques effectively limits the query images that can be registered to views similar to the images used for the reconstruction, Irschara *et al*. propose to generate a set of synthetic camera views using the 3D model. Original and synthetic images are then combined to form the database for the retrieval step. A GPU implementation of the vocabulary tree approach is

[1] Code is available at http://www.graphics.rwth-aachen.de/localization

Dataset	# Cameras	# 3D Points	# Descriptors	Size (MB)	# Query Images
Dubrovnik	6044	1,886,884	9,606,317	1172	800
Rome	15,179	4,067,119	21,515,110	2626	1000
Vienna	1324	1,123,028	4,854,056	593	266

Table 1: The datasets used for evaluation. *Size* details the memory used for all descriptors (128 bytes per descriptor).

used to achieve real-time performance. Furthermore, the set of images in the database is reduced to a distinct subset of the original and synthetic images that covers all 3D points, effectively reducing the size of the inverted files and thus retrieval time. Similarly, Arth *et al*. use subsets of 3D points defined by the images in which they occur to accelerate the computation of 2D-to-3D correspondences for localization on mobile phones [1].

While all of the approaches discussed above try to find the camera pose by matching 2D features in the query image to points in the database, Li *et al*. apply the matching step into the opposite direction [16]. They propose a prioritization scheme to avoid having to match every point in the model against the query image. This way, the priority of a point is directly related to its importance for the reconstruction, *i.e*. the number of cameras from the reconstruction it is visible in. Starting with a subset of points with high priority that covers all cameras, every point that can be matched against the features of the query image increases the priority of the points that can be seen in a camera with it. The matching process stops if enough correspondences are found. Li *et al*. also show that using a reduced set of points of highest priority is better than using all 3D points, as it permits to register more images while reducing the time needed for registration. As their method does not constrain the set of possible views, they are able to demonstrate that their method outperforms the algorithm by Irschara *et al*. in terms of the number of images that can be registered.

In this paper, we also focus on accurate localization. We propose an efficient 2D-to-3D matching approach that achieves better registration performance than both [16] and [13], while reaching competitive to superior runtimes.

2. Experimental Setup

In the following, we introduce the datasets and evaluation measures used throughout this paper. Based on this setup, we experimentally show in Sec. 3 that direct 2D-to-3D matching offers considerable potential for improvement over current indirect approaches and use these findings to derive a direct matching framework in Sec. 4.

Datasets. We evaluate all methods on the three datasets presented in [13, 16], kindly provided by Li *et al*. [16]. The two larger datasets, Dubrovnik and Rome, were reconstructed using photos taken from Flickr, while the images used to build the Vienna dataset were systematically taken with a single calibrated camera. The query images have a maximum dimension (height and width) of 1600 pixels. For Vienna, they were obtained by selecting 266 images from the Panoramio website that depict parts of the model [13]. Li *et al*. obtained query images for Dubrovnik and Rome by removing a random subset of images from the reconstructions, together with their descriptors and all 3D points that are visible in only one other camera not included in the set. Details on the datasets can be found in Table 1. Since Li *et al*. removed points that were too ill-conditioned or behind cameras before publishing the data, the Dubrovnik and Vienna datasets contain slightly fewer 3D points than in [16].

The three datasets are representatives for different scenarios: The Vienna model depicts parts of urban scenes constructed from photos taken at uniform intervals. Dubrovnik represents a large urban scene constructed from a more clustered set of views that is usually found when using photos from Internet photo collections, and Rome consists of reconstructions of a set of distinct landmarks.

Evaluation measures. The main criteria for evaluating all localization approaches are the number of images that they can successfully register, as well as the average time they need to register or reject an image. As proposed in [16], we accept a query image as registered if the best pose estimated by RANSAC from the established 2D-to-3D correspondences has at least 12 inliers. Except for the results in Table 2, every experiment was repeated 10 times to account for variations due to the random nature of RANSAC. In those cases, we report the mean number of images that can be registered together with its standard deviation, as well as the variation of the mean recognition times.

3. 2D-to-3D Matching

As discussed in Sec. 1, most current localization methods use an indirect way of establishing correspondences between 2D features and 3D points. As a result, those approaches trade off registration performance for faster search. This trade-off is well-known. However, it has so far escaped attention how large the potential for improvement through direct 2D-to-3D matching actually is. In order to demonstrate this potential, we perform the following systematic experiment: To establish 2D-to-3D correspondences, we use the kd-tree based approach proposed by Lowe [17], based on the FLANN library [18]. We represent each 3D point by the mean of its SIFT descriptors obtained from the reconstructions. A 2D-to-3D correspondence is accepted if the two nearest neighbors pass the SIFT ratio test with the threshold set to 0.7. Contrary to [16], we find that this direct application of the test gives excellent results. If more than one 2D feature matches to a 3D point, we keep only the correspondence with the most similar descriptor (measured by the Euclidean distance). The 6-point DLT algorithm [10] is used in a RANSAC loop [4] to estimate the 3D pose of the camera. We enforce that all inliers lie in front of the camera. Since several query images have an ex-

	Method	Dubrovnik registered images # reg. images	Dubrovnik registered images # corr.	Dubrovnik registered images inlier ratio	Dubrovnik registered images corr. [s]	Dubrovnik registered images RNSC. [s]	Dubrovnik registered images total [s]	Dubrovnik rejected time [s]	Rome registered # reg. images	Rome registered time [s]	Rome rejected time [s]	Vienna registered # reg. images	Vienna registered time [s]	Vienna rejected time [s]
tree-based	flann 50 leaves	789	545.3	0.60	1.29	0.34	1.63	8.92	978	2.66	5.36	218	2.52	8.84
	flann 100 leaves	793	581.8	0.62	1.64	0.39	2.04	11.54	978	2.82	8.52	219	2.25	4.64
	flann 200 leaves	794	609.9	0.64	2.40	0.31	2.71	12.21	981	3.49	5.69	219	2.61	2.93
	flann 300 leaves	**795**	620.1	0.64	3.11	0.30	**3.40**	**14.45**	**983**	**3.97**	**6.27**	**220**	**3.44**	**2.72**
	flann 500 leaves	794	629.1	0.65	4.55	0.21	4.76	23.86	985	5.28	3.28	220	5.06	3.65
VPS	**all descriptors**	**785**	537.0	0.64	0.66	0.15	**0.81**	**2.19**	**979**	**1.53**	**4.07**	**211**	**1.83**	**9.95**
	mean	774	472.9	0.62	1.08	0.53	1.61	2.36	972	2.13	1.28	210	2.05	9.19
	medoid	762	412.5	0.62	0.50	0.34	0.84	1.58	961	1.05	3.74	203	2.23	9.40
	mean per vw	782	523.8	0.64	0.99	0.32	1.31	5.25	976	2.23	6.50	212	2.46	6.87
	integer mean per vw	**783**	523.0	0.64	0.56	0.31	**0.87**	**5.35**	**976**	**1.33**	**5.92**	**211**	**2.02**	**7.59**
	medoid per vw	778	480.7	0.63	0.52	0.14	0.66	4.34	972	1.17	7.27	211	1.81	8.25
P2F [16]		753	-	-	-	-	0.73	2.70	921	0.91	2.93	204	0.55	1.96

Table 2: Comparison of different direct 2D-to-3D matching approaches. For every dataset, we list the number of successfully registered images (#reg. images), as well as the average time needed to register / reject an image. For the Dubrovnik dataset, we furthermore report for successfully registered images the average number of correspondences (#corr.), the average inlier ratio, the average time needed to compute the correspondences (corr.) and the average time RANSAC needs to estimate the pose (RNSC.). As can be seen, direct 2D-to-3D matching considerably increases the number of images that can be registered, compared to the state-of-the-art [16]. In the following, we propose further improvements to also reduce its runtime.

tremely low inlier ratio, RANSAC is stopped after 1 minute. For every dataset, we try to register its query images. Results for different variants of the tree-based search (imposing different limits on the number of visited leaf nodes) can be found in Table 2. Since the behavior of the methods is similar on all datasets, we only give detailed results for the Dubrovnik dataset. As the comparison with the indirect P2F method proposed by [16] shows, direct matching bears significant potential as it is able to register many more images (795 *vs*. 753 for Dubrovnik, 985 *vs*. 921 for Rome, and 220 *vs*. 204 for Vienna). However, the runtime of tree-based direct search is not competitive. In the following, we therefore explore direct matching methods in order to derive an approach that keeps the good registration performance of traditional tree search but facilitates fast localization.

4. Vocabulary-based Prioritized Search (VPS)

The query images used in our experiments contain about 10k features on average. As becomes evident by the number of correspondences found (*c.f*. Table 2), tree-based search thus spends about 93% of its search time on features that do not lead to correspondences. For the case of 3D-to-2D matching, [16] show how to prioritize the descriptor matching step based on an estimate of the likelihood that a 3D point will lead to a correspondence. For 2D-to-3D matching, there is however no way of determining a-priori whether or not a 2D feature will lead to a correspondence. Therefore, we propose a prioritization scheme that is based on an estimate of the matching cost of each feature.

Figure 2 illustrates our proposed framework. We store the 3D points and their descriptors in a visual vocabulary for faster indexing. Similarly, each feature in the query image is assigned to a visual word. The feature's first and second approximate nearest neighbors are found by linear search through all 3D points associated with that word, and the ratio test is again used to accept 2D-to-3D correspondences. The number of descriptors stored in a visual word thus gives a good estimate of the matching cost for this particular query feature. To speed up descriptor matching, we propose to process the features in ascending order of their matching costs, starting with features whose activated visual words contain only few descriptors. The search stops once N_t correspondences have been found, which are then used for subsequent pose estimation.

In the remainder of this paper, we experimentally evaluate parameters and design variants of the proposed framework in order to answer the following questions: (1) How do we have to choose a set of descriptors to obtain a good representation of the 3D points? (2) What is the effect of the prioritized search on the performance? (3) How can rejection times be optimized? (4) What influence does the choice of the visual vocabulary have? (5) How do the variants compare to current state-of-the-art methods? (6) What localization accuracy can be achieved with different pose estimation algorithms?

5. Experimental Evaluation of VPS

For all of the following experiments, we again use the 6-point DLT algorithm in combination with RANSAC for pose estimation. Unless specified otherwise, we use a generic set of 100k visual words obtained from an image set not related to the used datasets. All visual words were obtained with the approximate k-means method from [20]. In all our experiments we use a common quantized representation for the SIFT descriptors, converting floating point descriptor entries to integer values in the range $[0, 255]$. The advantage of this representation is that it requires four times less memory compared to a floating point representation.

	N_t	all descriptors				integer mean per vw			
		# reg.	linear search [s]	RANSAC [s]	total [s]	# reg.	linear search [s]	RANSAC [s]	total [s]
Dubrovnik	50	778.90 ± 1.52	0.04	0.05	0.23 ± 0.00	775.80 ± 1.48	0.03	0.05	0.21 ± 0.00
	100	$\mathbf{783.90 \pm 1.60}$	0.10	0.08	$\mathbf{0.31 \pm 0.01}$	$\mathbf{782.00 \pm 0.82}$	0.08	0.08	$\mathbf{0.28 \pm 0.01}$
	150	783.90 ± 1.10	0.16	0.08	0.36 ± 0.01	781.80 ± 1.40	0.12	0.08	0.32 ± 0.01
	200	784.40 ± 1.26	0.20	0.08	0.40 ± 0.01	782.50 ± 1.35	0.15	0.08	0.35 ± 0.01
	∞	784.60 ± 1.17	0.47	0.08	0.68 ± 0.01	782.50 ± 1.08	0.34	0.08	0.54 ± 0.01
Rome	50	972.00 ± 1.41	0.06	0.02	0.18 ± 0.00	971.30 ± 1.25	0.05	0.02	0.16 ± 0.00
	100	$\mathbf{976.90 \pm 1.29}$	0.15	0.05	$\mathbf{0.29 \pm 0.00}$	$\mathbf{974.60 \pm 1.65}$	0.11	0.05	$\mathbf{0.25 \pm 0.00}$
	150	977.80 ± 1.32	0.23	0.06	0.39 ± 0.01	976.50 ± 1.51	0.17	0.06	0.33 ± 0.01
	200	979.20 ± 1.75	0.30	0.07	0.46 ± 0.01	976.90 ± 1.52	0.22	0.07	0.38 ± 0.00
	∞	980.10 ± 0.88	0.81	0.07	0.98 ± 0.00	976.90 ± 1.20	0.57	0.07	0.74 ± 0.00
Vienna	50	200.40 ± 1.26	0.02	0.13	0.28 ± 0.01	199.10 ± 1.20	0.02	0.10	0.26 ± 0.01
	100	$\mathbf{207.70 \pm 1.06}$	0.06	0.30	$\mathbf{0.50 \pm 0.02}$	$\mathbf{206.90 \pm 0.88}$	0.05	0.28	$\mathbf{0.46 \pm 0.02}$
	150	208.20 ± 0.92	0.09	0.30	0.52 ± 0.03	207.90 ± 0.74	0.07	0.29	0.50 ± 0.03
	200	208.80 ± 1.23	0.11	0.29	0.54 ± 0.04	208.20 ± 1.14	0.08	0.30	0.52 ± 0.03
	∞	207.90 ± 1.29	0.24	0.27	0.65 ± 0.03	208.20 ± 0.42	0.17	0.28	0.59 ± 0.03

Table 3: Improving the registration times: Results obtained when stopping the search for correspondences when N_t correspondences are found. The results show that a cutoff of $N_t = 100$ preserves the good registration performance while significantly reducing the runtime for successfully registered images. For all experiments, a minimal inlier ratio of 20% for RANSAC was assumed.

(1) 3D point representations. We evaluate different possible representations of 3D points through their corresponding descriptors. The simplest representation is to use *all descriptors* for each 3D point. In this case, we adopt the linear search such that the two nearest neighbors found belong to different 3D points. A much more compact representation can be obtained by using the *mean / medoid* of the point descriptors. The means / medoids are assigned to all visual words activated by any of the points' descriptors to reduce the discretization artifacts introduced by quantizing the descriptor space. The *mean / medoid per vw* representation first assigns the descriptors of a point to visual words. If more than one descriptor of the same 3D point is assigned to one visual word, we represent this set of descriptors by its mean / medoid. This representation finds a compromise between using all descriptors and a single descriptor per point. Contrary to *mean / medoid*, it is able to adapt to the set of visual words while saving memory by not using all descriptors. A slight modification, *integer mean*, rounds the entries of the means of descriptors to the nearest integer values.

Table 2 compares the different representations in the experimental setup of Sec. 3, with $N_t = \infty$. As can be seen, representations adapting to the subdivision induced by the visual words clearly outperform those where the representative descriptor is selected independently of the quantization. Medoid descriptors perform worse than means of descriptors, while there is nearly no difference between mean and integer mean, except faster correspondence search for the latter since integer arithmetics can be used. The best representation for the 3D points can thus be obtained by using *all descriptors* or *integer mean per vw*.

(2) Prioritized search. The prioritized search proposed in Sec. 4 first searches for correspondences in those parts of descriptor space that contain fewer descriptors, which are most likely regions with a sparser distribution. This might bias the correspondence search towards false correspondences since a sparser distribution makes it much more likely that an unrelated feature and 3D point might pass the ratio test. Table 3 details the effect of different choices for N_t on both registration performance and registration times. We do not report rejection times, as the number of correspondences found for rejected images is usually so low that the threshold N_t is rarely reached. Choosing $N_t = 100$ preserves the good registration performance, while resulting in significantly faster search times. Note that the average time spent on RANSAC-based pose estimation does not change significantly for $N_t \geq 100$, indicating that the average inlier ratio for registered images does not change too much between different values for N_t. We therefore conclude that the threshold on the number of correspondences does not bias the search towards false correspondences.

(3) Improving the rejection times. As shown in Table 2, the average time needed to reject an image is very large, especially for the Vienna dataset. This is due to the low initial inlier ratio of $12/N$ when N correspondences are found but no pose can be estimated. Since rejection time is also critical for some applications, we set the initial inlier ratio for RANSAC to $\max(12/N, R)$, which limits the maximal number of samples taken by RANSAC. The value for R should be chosen such that it significantly reduces the number of RANSAC iterations, while not rejecting too many images that could otherwise be registered. Results for $N_t = \infty$ and different values for R are shown in Fig. 3, based on which we select $R = 0.2$.

As all query images belonging to our evaluation datasets can potentially be registered, we test the robustness of our method against false positives by trying to register query images from the other datasets. While none of those images could be registered, we notice that the rejection times for images from other datasets are substantially lower than

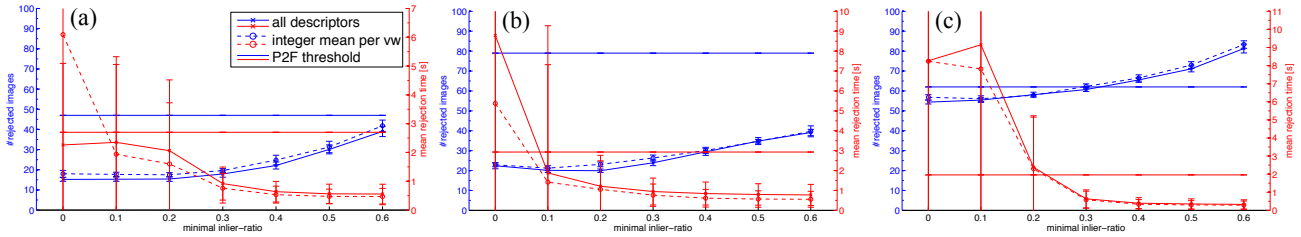

Figure 3: Improving the rejection times: The figure shows the relationship between RANSAC's minimal inlier ratio parameter and both the number of images that could not be registered and the average time needed to reject an image for (a) Dubrovnik, (b) Rome, and (c) Vienna. The horizontal blue line denotes the number of images that P2F [16] cannot register; the red line shows the average time P2F needs to reject an image. As can be seen, setting the minimal inlier ratio to 20% has almost no effect on our approach's registration performance, while significantly reducing its rejection times.

	all descriptors, $N_t = 100$, $R = 0.2$					integer mean per vw, $N_t = 100$, $R = 0.2$					
Vocabulary	#reg. images	linear search [s]	RANSAC [s]	reg. [s]	rej. [s]	#reg. images	linear search [s]	RANSAC [s]	reg. [s]	rej. [s]	# desc.
Dubrovnik (G)	783.90 ± 1.60	0.10	0.08	0.31 ± 0.01	2.22 ± 0.26	782.00 ± 0.82	0.08	0.08	0.28 ± 0.01	1.70 ± 0.18	6706575
Dubrovnik (S)	782.90 ± 1.29	0.08	0.05	0.25 ± 0.01	1.35 ± 0.30	781.50 ± 0.97	0.06	0.06	0.24 ± 0.01	0.93 ± 0.12	6414463
Rome (G)	976.90 ± 1.29	0.15	0.05	0.29 ± 0.00	1.90 ± 0.10	974.60 ± 1.65	0.11	0.05	0.25 ± 0.00	1.66 ± 0.10	14577088
Rome (S)	973.60 ± 1.26	0.11	0.05	0.25 ± 0.01	1.59 ± 0.14	970.80 ± 1.48	0.08	0.05	0.23 ± 0.00	1.42 ± 0.07	14205926
Vienna (G)	207.70 ± 1.06	0.06	0.30	0.50 ± 0.02	2.40 ± 0.06	206.90 ± 0.88	0.05	0.28	0.46 ± 0.02	2.43 ± 0.08	3420234
Vienna (S)	212.80 ± 1.40	0.05	0.28	0.46 ± 0.03	1.65 ± 0.09	210.60 ± 0.70	0.04	0.28	0.45 ± 0.01	1.60 ± 0.03	3191015

Table 4: Results of using a generic (G) visual vocabulary and a vocabulary specifically trained (S) for the dataset. Specific vocabularies mainly improve registration and rejection times without significantly affecting the registration performance.

for images belonging to the same set. This is caused by a smaller number of correspondences found for those images, such that $12/N > R = 0.2$ holds, which in turn significantly accelerates RANSAC.

(4) Influence of the vocabulary. So far, all tests were performed with a generic set of 100k visual words trained on an unrelated image set. In order to check how the choice of visual vocabulary affects performance, we trained for every dataset a specific set of 100k visual words from all descriptors of all 3D points in the corresponding model. Fig. 4 compares the distribution of the number of descriptors per visual word for the generic vocabulary (top row) with the distribution for the specifically trained one (bottom row). We observe that the specific vocabularies contain less empty visual words, providing a better representation for the 3D point descriptors. Table 4 shows the effect on registration performance and runtimes. As can be seen, the specific vocabularies result in an improved registration speed, visible in both the linear search and in the RANSAC stage. However, they can only improve registration performance for Vienna by 5 images and do not lead to increased registration rates for the Dubrovnik and Rome datasets.

We also tested vocabularies with 10k and 1M visual words and found that the quantization provided by 1M visual words was too fine, resulting in too few correspondences per image. Using 10k had little positive impact on the number of images that could be registered, while the linear search became much more expensive.

(5) Comparison with state-of-the-art. Table 5 compares our approach (choosing $N_t = 100$, $R = 0.2$, and the generic set of visual words) to published results of current state-of-the-art methods for localization. P2F (points-to-features) denotes the prioritized feature matching proposed by Li *et al*. [16]. P2F+F2P is an improvement also proposed by Li *et al*., matching features from the images against points in the database when P2F failed. Results from the GPU-based image retrieval method by Irschara *et al*. [13] are only available for the Vienna dataset. For the other datasets, we report the vocabulary tree-based image retrieval implementation from [16], using either all features in the database images or only features belonging to 3D points. Both implementations retrieve the 10 top ranked images for each query image and then perform a pairwise matching between the images followed by pose estimation.

Our experiments were performed on an Intel i7-920 processor with 2.79GHz and the results from [16] were reported for a 2.80GHz Xeon processor, both using a single thread. The results from [13] were obtained on an Intel Pentium D 3.2GHz with a GeForce GTX 280 graphics card.

As can be seen from this comparison, our approach clearly outperforms all other published methods on the Dubrovnik and Rome datasets, both in terms of the number of registered images and in the time needed to register / reject an image. For the Vienna dataset, we reach comparable performance to P2F, with faster registration and slightly higher rejection times.

(6) Localization accuracy. The previous experiments have shown that our approach can register considerably more images than P2F [16] based on the 12-inlier criterion. It could be argued that those additional images may have a very

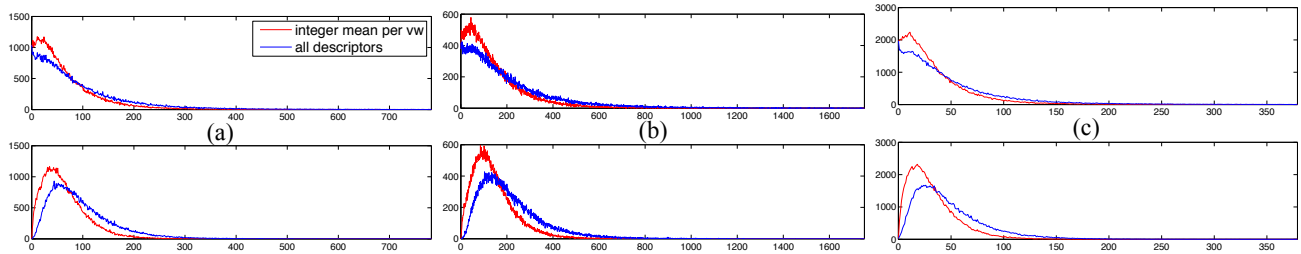

Figure 4: Distribution of the number of descriptors per visual word for (a) Dubrovnik, (b) Rome and (c) Vienna. The top row displays the results for a generic vocabulary, the bottom row for a vocabulary specifically trained for each reconstruction.

Method	Dubrovnik			Rome			Vienna		
	# reg. images	registr. time [s]	rejection time[s]	# reg. images	registr. time [s]	rejection time [s]	# reg. images	registr. time [s]	rejection time [s]
all descriptors	783.9 ± 1.60	0.31 ± 0.01	2.22 ± 0.26	976.9 ± 1.29	0.29 ± 0.00	1.90 ± 0.10	207.7 ± 1.06	0.50 ± 0.02	2.40 ± 0.06
integer mean per vw	782.0 ± 0.82	0.28 ± 0.01	1.70 ± 0.18	974.6 ± 1.65	0.25 ± 0.00	1.66 ± 0.10	206.9 ± 0.88	0.46 ± 0.02	2.43 ± 0.08
P2F [16]	753	0.73	2.70	921	0.91	2.93	204	0.55	1.96
P2F+F2P [16]	753	0.70	3.96	924	0.87	4.67	205	0.54	3.62
Voc. tree (all)[16]	668	1.4	4.0	828	1.2	4.0	-	-	-
Voc. tree (points)[16]	677	1.3	4.0	815	1.2	4.0	-	-	-
Voc. tree GPU [13]	-	-	-	-	-	-	165	≤ 0.27	

Table 5: Comparison of the methods proposed in this paper with other state-of-the-art localization approaches. As can be seen, our approach achieves superior registration performance at faster registration times.

large localization error. In order to verify whether this is the case, we evaluate our approach's localization accuracy using a metric model of the Dubrovnik dataset provided by Li *et al*. [16]. Localization errors are computed by estimating the average camera position for each registered image over 10 repetitions and measuring the distance in meters between the average position and the ground truth position.

Besides the standard 6-point DLT algorithm for pose estimation (p6p), we also experiment with two other pose estimation strategies: (1) When an image's focal length is available from the EXIF tag, we use the 3-point-pose solver by [7] to estimate the camera pose instead of the 6-point solver (p3p/p6p). (2) We employ a 4-point-pose solver that explicitly estimates the focal length and radial distortion of the camera [14] in order to refine the results obtained from the DLT algorithm (p6p+p4pfr). For this experiment, we use the MATLAB implementation provided by [14] which needs around 60ms for a single pose estimate.

Table 6 shows the results of this experiment. For most images, localization is far more accurate than for P2F (as evidenced by the median and quartiles), with only very few images having a large localization error. We manually inspected all those images that have a localization error above 400m. Similar to [16], we found that the large deviations are due to errors in the focal length estimate of the cameras, in most cases caused by 3D points on a plane parallel to the image plane. Another source of error are badly localized 3D points that are kilometers away from the images used to reconstruct them. In some query images, only those are found as correspondences, again leading to a low localization accuracy. The p4pfr solver [14] can be used to improve the localization error in such cases, since it reduces the degrees of freedom in the estimate of the internal camera calibration. Using the p3p-based methods, we found that 5 of the images with a localization error larger that 400m had a focal length estimate in their EXIF tag. Comparing those to the estimates from the ground-truth reconstruction, the focal length estimates differed significantly for 4 of the images. Still, using the focal length estimates from the EXIF tags of the images reduced the error for most images. Both of those modifications, p3p and p4pfr, can therefore be recommended for practical applications.

6. Conclusion & Future Work

In this paper we have demonstrated that direct 2D-to-3D descriptor matching yields registration results superior to the previously developed indirect methods. We have presented a modular framework for direct matching that is easy to implement and to extend. By exploring the design space of our framework through extensive experiments, we have shown how our approach can be adapted to offer faster image-based localization than the current state-of-the-art at significantly higher registration rates. In addition, our method yields localization results that are more accurate for most images. The comparison with tree-based descriptor search shows that there is still a considerable remaining potential for direct 2D-to-3D matching, waiting to be unlocked in future work. Another interesting question is the scalability of direct 2D-to-3D matching to even larger datasets, especially if their descriptor spaces eventually become too dense for the traditional SIFT ratio test. In this case, it might be necessary to remove confusing descriptors [15] or to initially allow ambiguous matches which are resolved by RANSAC [12]. The memory needed to store such large datasets is another important aspect. Here, re-

Method	# reg. images	Mean [m]	Median [m]	1st Quartile [m]	3rd Quartile [m]	Max [m]	#images with error	
							< 18.3m	> 400 m
P2F [16] (p6p)	753	18.3	9.3	7.5	13.4	~ 400	655	-
all descriptors (p6p)	783.9 ± 1.60	53.9	1.4	0.4	5.9	7934.3	685	16
integer mean per vw (p6p)	782.0 ± 0.82	47.0	1.3	0.5	5.1	7737.1	675	13
all descriptors (p6p+p4pfr)	783.9 ± 1.60	21.6	0.8	0.2	3.0	2336.1	705	10
integer mean per vw (p6p+p4pfr)	782.0 ± 0.82	17.2	0.8	0.2	3.6	875.6	700	9
all descriptors (p3p/p6p)	773.0 ± 2.98	15.6	1.5	0.5	5.7	799.3	696	8
integer mean per vw (p3p/p6p)	773.0 ± 1.41	17.7	1.5	0.5	5.6	1899.8	692	8
all descriptors (p3p/(p6p+p4pfr))	773.0 ± 2.98	15.7	1.3	0.4	5.4	776.9	694	7
integer mean per vw (p3p/(p6p+p4pfr))	773.0 ± 1.41	14.9	1.3	0.4	5.2	777.5	695	7

Table 6: Localization errors for Dubrovnik using $N_t = 100$, $R = 0.2$, and the generic vocabulary. As can be seen, our approach achieves a higher localization accuracy than P2F [16] for most images. Although p6p pose estimation returns some extreme outlier results, those can be corrected by an additional p4pfr estimation step on p6p's inliers. p3p also improves both localization accuracy and runtime and is recommended as a replacement for p6p whenever the focal length is available.

cently developed binarization methods such as [28] promise to drastically decrease memory consumption, while also enabling faster descriptor comparisons.

Acknowledgments: We gratefully acknowledge support by UMIC (DFG EXC 89) and Mobile ACcess (EFRE 280401102). We also thank Noah Snavely, Yunpeng Li and Arnold Irschara for kindly providing their datasets and for helpful discussions.

References

[1] C. Arth, D. Wagner, M. Klopschitz, A. Irschara, and D. Schmalstieg. Wide area localization on mobile phones. In *ISMAR*, 2009.

[2] Y. Avrithis, Y. Kalantidis, G. Tolias, and E. Spyrou. Retrieving landmark and non-landmark images from community photo collections. In *ACM Multimedia*, 2010.

[3] R. O. Castle, G. Klein, and D. W. Murray. Video-rate localization in multiple maps for wearable augmented reality. In *ISWC*, 2008.

[4] O. Chum and J. Matas. Optimal Randomized RANSAC. *PAMI*, 30(8):1472 –1482, 2008.

[5] M. Cummins and P. Newman. FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance. *IJRR*, 27(6):647–665, 2008.

[6] E. Eade and T. Drummond. Scalable monocular slam. In *CVPR*, 2006.

[7] M. Fischler and R. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Comm. ACM*, 24(6), 1981.

[8] J.-M. Frahm, P. Fite-Georgel, D. Gallup, T. Johnson, R. Raguram, C. Wu, Y.-H. Jen, E. Dunn, B. Clipp, S. Lazebnik, and M. Pollefeys. Building rome on a cloudless day. In *ECCV*, 2010.

[9] S. Gammeter, L. Bossard, T. Quack, and L. V. Gool. I know what you did last summer: object-level auto-annotation of holiday snaps. In *ICCV*, 2009.

[10] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge Univ. Press, 2nd edition, 2004.

[11] J. Hays and A. A. Efros. im2gps: estimating geographic information from a single image. In *CVPR*, 2008.

[12] E. Hsiao, A. Collet, and M. Hebert. Making specific features less discriminative to improve point-based 3d object recognition. In *CVPR*, 2010.

[13] A. Irschara, C. Zach, J.-M. Frahm, and H. Bischof. From structure-from-motion point clouds to fast location recognition. In *CVPR*, 2009.

[14] K. Josephson and M. Byröd. Pose estimation with radial distortion and unknown focal length. In *CVPR*, 2009.

[15] J. Knopp, J. Sivic, and T. Pajdla. Avoiding confusing features in place recognition. In *ECCV*, 2010.

[16] Y. Li, N. Snavely, and D. P. Huttenlocher. Location recognition using prioritized feature matching. In *ECCV*, 2010.

[17] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[18] M. Muja and D. G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *VISAPP*, 2009.

[19] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *CVPR*, 2006.

[20] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, 2007.

[21] M. Pollefeys *et al.*(19 authors). Detailed real-time urban 3d reconstruction from video. *IJCV*, 78(2-3):143–167, 2008.

[22] D. Robertson and R. Cipolla. An image-based system for urban navigation. In *BMVC*, 2004.

[23] G. Schindler, M. Brown, and R. Szeliski. City-scale location recognition. In *CVPR*, 2007.

[24] S. Se, D. Lowe, and J. Little. Global localization using distinctive visual features. In *IROS*, 2002.

[25] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *ICCV*, 2003.

[26] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3d. In *SIGGRAPH*, 2006.

[27] N. Snavely, S. M. Seitz, and R. Szeliski. Modeling the world from internet photo collections. *IJCV*, 80(2):189–210, 2008.

[28] C. Strecha, A. M. Bronstein, M. M. Bronstein, and P. Fua. LDAHash: Improved matching with smaller descriptors. In *Technical Report*, 2010.

[29] C. Strecha, T. Pylvanainen, and P. Fua. Dynamic and Scalable Large Scale Image Reconstruction. In *CVPR*, 2010.

[30] A. R. Zamir and M. Shah. Accurate image localization based on google maps street view. In *ECCV*, 2010.

[31] W. Zhang and J. Kosecka. Image based localization in urban environments. In *3DPVT*, 2006.