This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2611662, IEEE Transactions on Pattern Analysis and Machine Intelligence

Efficient & Effective Prioritized Matching for Large-Scale Image-Based Localization

Torsten Sattler, *Member, IEEE,* Bastian Leibe, *Member, IEEE,* and Leif Kobbelt

Abstract—Accurately determining the position and orientation from which an image was taken, i.e., computing the camera pose, is a fundamental step in many Computer Vision applications. The pose can be recovered from 2D-3D matches between 2D image positions and points in a 3D model of the scene. Recent advances in Structure-from-Motion allow us to reconstruct large scenes and thus create the need for image-based localization methods that efficiently handle large-scale 3D models while still being effective, i.e., while localizing as many images as possible. This paper presents an approach for large scale image-based localization that is both efficient and effective. At the core of our approach is a novel prioritized matching step that enables us to first consider features more likely to yield 2D-to-3D matches and to terminate the correspondence search as soon as enough matches have been found. Matches initially lost due to quantization are efficiently recovered by integrating 3D-to-2D search. We show how visibility information from the reconstruction process can be used to improve the efficiency of our approach. We evaluate the performance of our method through extensive experiments and demonstrate that it offers the best combination of efficiency and effectiveness among current state-of-the-art approaches for localization.

Index Terms—Image-based Localization, Location Recognition, Prioritized Feature Matching, Camera Pose Estimation

1 Introduction

Determining the camera pose for a given image, also known as the *image-based localization problem*, is an important step in many Computer Vision applications such as Structure-from-Motion (SfM) [1], location recognition [2], Augmented Reality [3], and visual navigation for autonomous vehicles.

Given a 3D model of the scene, the camera pose can be estimated from 2D-3D matches relating 2D image positions and 3D points in the model by applying an n-point pose solver inside a RANSAC-loop [4]. In the case that the scene model was reconstructed using SfM, each 3D point was triangulated from multiple local features and we can associate it with the corresponding image descriptors. Consequently, we can establish 2D-3D correspondences through nearest neighbor search between the descriptors of local features found in the query image and the descriptors of the 3D points. Recent advances in SfM techniques [1], [5] and the fact that increasingly larger parts of our world are covered by photos available on websites such as Flickr, Google Street View, and Mapillary enable us to efficiently reconstruct 3D structures on a truly large scale. This in turn creates the need to develop image-based localization methods that are able to efficiently handle datasets consisting of millions of 3D points. Besides *efficiency*, we are also interested in the *accuracy* of the estimated poses and the *effectiveness* of localization methods, i.e., the ability to localize as many query images as possible.

In this paper, we propose a novel image-based localization approach that is efficient, effective, and accurate. We make three main contributions: First, we develop a simple but powerful prioritization scheme that allows us to significantly accelerate 2D-to-3D matching. The prioritization scheme uses a visual vocabulary-based quantization of the descriptor space and is thus prone to quantization artifacts. As the second contribution, we show that we can recover matches lost due to quantization through 3D-to-2D search. We prove that our approach is computationally more efficient that the standard approach of searching through multiple visual words [6]. As the third major contribution, we demonstrate how co-visibility information that is readily available from the SfM process can be used to make all stages of our localization pipeline more efficient. Our method offers the fastest localization times published so far while achieving an effectiveness similar or superior to approaches that are orders of magnitude slower.

This paper is based on our previous publications [7], [8]. Besides a more thorough comparison with the current state-of-the-art, additional experiments on more complex datasets, and an updated evaluation of different n-point-pose solvers, we justify our prioritization strategy based on a probabilistic formulation of 2D-to-3D matching. We also evaluate a more advanced implementation of our approach[1] and analyze the scalability of our framework.

Related work. Traditionally, image-based localization

- *Torsten Sattler is with the Computer Vision Group at the Department of Computer Science at ETH Zurich.*
- *Bastian Leibe is the head of the Computer Vision Group at RWTH Aachen University.*
- *Leif Kobbelt is the head of the Computer Graphics Group at RWTH Aachen University.*

1. Source code available at https://github.com/tsattler/vps.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2611662, IEEE Transactions on Pattern Analysis and Machine Intelligence

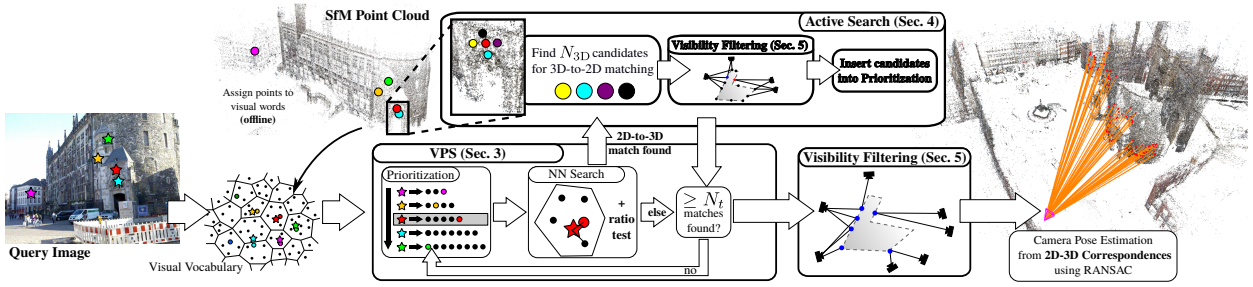

Fig. 1. Overview over the framework for efficient and effective image-based localization proposed in this paper.

has been cast as a place recognition problem, where image retrieval techniques [6], [9], [10], [11] are used to find a set of photos similar to the query in a database of geo-tagged images [12], [13], [14]. Over the years, various improvements have been made to handle confusing [15] and repetitive [16] structures, scale to larger scenes [14], [17], or use the original feature descriptors for retrieval [18], [19]. In contrast to our method, these approaches only approximate the pose of the query camera [14], [17], [18], [20].

Contrary to purely image-based methods, Irschara et al. assume that they are given a SfM reconstruction of the scene [21]. They use retrieval techniques to determine a set of relevant database images and then match the query images against the 3D points visible in the retrieved photos. To increase the robustness of the retrieval stage to viewpoint changes, they add synthetic views to the database. These views can be generated by projecting the 3D points into virtual cameras placed on the ground plane [21] or by rendering synthetic images and extracting features from them [22], [23]. Li et al. show that directly comparing the descriptors of the query features and 3D points is more effective than using image retrieval [2]. Starting with a set of seed points, they employ a prioritized 3D-to-2D matching scheme based on the co-visibility of the scene points to compare 3D points against the query image. Due to the high false positive matching rate of 3D-to-2D search, it is more effective to use 2D-to-3D matching to generate seed points [24]. While these methods focus on 3D-to-2D matching, we show that combining 2D-to-3D and 3D-to-2D search leads to a superior prioritization scheme.

Lowe's ratio test [25], used to detect ambiguous correspondences, rejects more and more correct matches for larger models [26]. In order to enable scalable localization, Li et al. modify RANSAC's sampling scheme to handle higher outlier ratios, enabling them to accept more matches overall, and perform 3D-to-2D search if the pose cannot be estimated from 2D-to-3D matches alone. Both Svarm et al. and Zeisl et al. propose deterministic approaches for camera pose estimation that are able to handle inlier ratios of 1% or less by exploiting IMU measurements about the gravity direction and a rough estimate of the camera height [27], [28]. Sattler et al. demonstrate that retrieval-based approaches can achieve state-of-the-art effectiveness by using a more restrictive voting approach to avoid incorrect votes for unrelated database images [29]. Cao & Snavely further improve retrieval-based methods by dividing the 3D model into distinctive places and learning to distinguish between them [30]. While the previously discussed approaches rely on SIFT descriptors for matching, Sattler et al. perform matching via a fine visual vocabulary of 16M words [31], enabling scalable localization by reducing the memory requirements [32]. Inspired by our approach, they use co-visibility information to select subsets of matches for pose estimation. While these methods are mainly concerned with effectiveness, our approach focusses on efficiency. Still, combining 2D-to-3D and 3D-to-2D search into one prioritization scheme allows us to achieve a competitive localization effectiveness.

Donoser & Schmalstieg formulate 2D-to-3D matching as a classification problem [33]. While this leads to significantly reduced memory requirements, they require a location prior to find good matches. Such priors can also be used to predict which 3D points are potentially visible in a query image [34]. In contrast, our method does not require any pose prior.

Instead of requiring an existing 3D model, Simultaneous Mapping and Localization (SLAM) approaches concurrently build a model of the scene and use it for localization [35], [36]. Even when run in a separate thread [37], mapping the scene quickly becomes computationally infeasible for larger scenes. Recently, Middelberg et al. proposed a hybrid approach for localization on mobile devices [3]. They build a fixed-size map, used for real-time pose tracking, via SLAM and utilize an external localization server, which employs the method presented in this paper, to provide pose estimates relative to a global 3D model.

2 2D-to-3D vs. 3D-to-2D Matching

In this paper, we follow a direct matching strategy and directly compare the descriptors of the 2D query features and the 3D points in the model. Thus, we can choose between matching the 2D features against the 3D points (*2D-to-3D* matching) or the point descriptors against the features (*3D-to-2D* search). In this sec-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2611662, IEEE Transactions on Pattern Analysis and Machine Intelligence

tion, we compare the advantages and disadvantages of the two search directions.

Let $\mathcal{P}$ be the set of all 3D points in the model and let $\mathcal{F}$ be the set of all features found in the query image. Using tree-based approximate search [38], we can match all features against the 3D points in time $\mathcal{O}(|\mathcal{F}| \log |\mathcal{P}|)$. Matching all points against the query image requires time $\mathcal{O}(|\mathcal{P}| \log |\mathcal{F}|)$. For large-scale scenes, there are orders of magnitude more points than query features. Thus, 3D-to-2D matching will only be more efficient than 2D-to-3D search if only a small fraction of all points is considered. Based on this insight, Li et al. propose a prioritized point-to-feature (P2F) matching strategy based on co-visibility information [2]: Starting with a set of seed points selected from all parts of the model, they match points against the query image in order of descending priorities. Once a new match is found for a point p, P2F increases the priorities of all other points that are visible together with p in at least one database image. The search is stopped once 100 matches are found or when a fixed number of points has been tried.

Independently of the search direction, Lowe's ratio test is used to reject ambiguous matches. After finding the two nearest neighbors $\mathbf{d}_1$, $\mathbf{d}_2$ for a query descriptor $\mathbf{d}$, a correspondence between the feature / point corresponding to $\mathbf{d}$ and the point / feature belonging to $\mathbf{d}_1$ is established only if the ratio test

$$\|\mathbf{d} - \mathbf{d}_1\|_2 < \tau \cdot \|\mathbf{d} - \mathbf{d}_2\|_2 \tag{1}$$

is passed [25], where τ is typically from the range $[0.6, 0.8]$. Interestingly, the effectiveness of the ratio test strongly differs for the two search directions: Consider a set of points with similar descriptors, found in unrelated parts of the model. Since 2D-to-3D search performs matching on a global level, the second nearest neighbor is also contained in this set, allowing the ratio test to discard the match. In contrast, 3D-to-2D matching is oblivious to this global ambiguity since it considers each 3D point independently of the others. Thus, it is likely that the ratio test accepts matches for all 3D points in the set if one of the points passes the test, leading to a significantly higher false positive matching rate. At the same time, 2D-to-3D search is more likely to reject correct matches due to such global ambiguities. For larger models, 2D-to-3D matching might thus reject too many correct matches, leading to a reduced localization effectiveness.

We experimentally compare the efficiency and effectiveness of 2D-to-3D and 3D-to-2D matching on three standard datasets (cf. Sec. 6). We use the FLANN library [38], configured to visit a fixed number L of leaf nodes, for kd-tree search and compare it against the P2F method from [2]. For P2F, we also report results obtained by [2] on more compact versions of the datasets, generated by selecting a subset of the points. Both P2F and the kd-tree approach use the ratio test threshold $\tau = 0.7$. In case that multiple matches are found for a single point / feature, only the match with the smallest SIFT descriptor distance is kept. The camera pose is estimated by applying the 6pt DLT solver that computes the projection matrix from 6 matches [39] inside a RANSAC loop [4]. Following [2], [24], [26], a query image is considered to be successfully localized or registered if the best pose found by RANSAC has at least 12 inliers. Fig. 2 shows the number of localized query images as well as the mean times required to localize an image (not including feature extraction) for the two approaches. As can be seen, using a more compact representation boosts the effectiveness of P2F due to finding fewer false positive matches. Still, there is a significant gap in the number of localized images compared to the kd-tree approach, indicating that P2F is prone to terminate its search too early due to still accepting too many wrong correspondences. These results show that 2D-to-3D search achieves a higher localization effectiveness due to its low false positive matching rate. In the following, we thus focus on increasing the efficiency of 2D-to-3D search.

3 Vocabulary-based Prioritization

As can be seen in Fig. 2, the higher effectiveness of 2D-to-3D matching comes at the prize of a reduced efficiency. Fortunately, there is considerable potential to improve the efficiency of 2D-to-3D search without sacrificing its effectiveness. As illustrated in Fig. 3, only 10% or less of the features found in most of the query images from the three datasets have a corresponding 3D point. This implies that we can accelerate correspondence search by an order of magnitude for these images if we can determine which features will yield a match before actually performing descriptor matching. Since this seems to be impossible, we consider a slightly different problem: Select a subset of query features that minimizes the search costs under the constraint that we can *expect* to find $N_t > 0$ matches. N_t is a parameter of our approach and controls the balance between run-time efficiency and localization effectiveness. This probabilistic scenario does not require that N_t actual matches are found but that it is likely to find N_t correspondences for the selected features. Unfortunately, this problem is NP-complete. Thus, we first derive an computationally efficient approach that performs close to optimal. We then adapt the solution to the original scenario, resulting in our novel *Vocabulary-based Prioritized Matching* (VPS) scheme.

Probabilistic 2D-to-3D matching. In the considered probabilistic scenario, a random variable $F_i \in \{0, 1\}$ is assigned to the i^{th} feature $f_i \in \mathcal{F}$ in the query image. $F_i = 1$ denotes the event that a matching 3D point can be found for f_i while $F_i = 0$ corresponds to the event that no correspondence can be established. Let P_i denote the probability of finding a match, i.e.,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2611662, IEEE Transactions on Pattern Analysis and Machine Intelligence

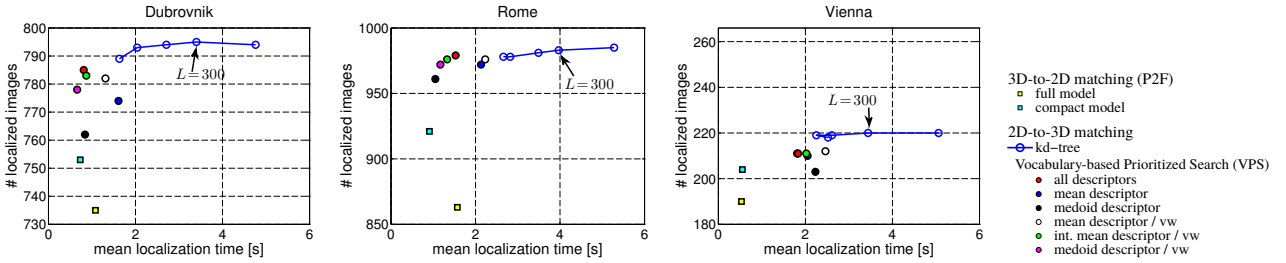

Fig. 2. 2D-to-3D matching approaches (tree-based and VPS) achieve a higher localization effectiveness than methods based on 3D-to-2D search such as Point-to-Feature (P2F) matching [2] on three standard datasets. For kd-tree search, we visit $L = \{50, 100, 200, 300, 500\}$ leaf nodes (left to right).

$\mathtt{Pr}[F_i = 1] = P_i$. We express the selection of a subset $\mathcal{F}' \subset \mathcal{F}$ of all query features by introducing indicator variables $X_i \in \{0, 1\}$, where $X_i = 1$ denotes that the i^{th} feature was selected. The expected number of matches that can be found for a subset $\mathcal{F}'$ is given by

$$\mathtt{E}[\mathcal{F}'] = \sum_i X_i \cdot (P_i \cdot 1 + (1 - P_i) \cdot 0) = \sum_i X_i \cdot P_i \ . \quad (2)$$

The problem of finding a subset $\mathcal{F}'$ that minimizes the overall search cost while resulting in N_t or more expected matches can be expressed as

$$\min_{\{X_i\}} \sum_i X_i \cdot c_i \text{ s.t. } \sum_i X_i P_i \geq N_t \ , \ X_i \in \{0, 1\} \ . \quad (3)$$

Both the probabilities p_i and search costs c_i can be computed efficiently by quantizing the descriptor space using a visual vocabulary $\mathcal{W}$ obtained through (approximate) k-means clustering [11]: In an offline step, we assign each descriptor of a 3D point to its closest visual word [38]. For each visual word $\omega \in \mathcal{W}$, we store the number $n_D(\omega)$ of 3D point descriptors assigned to it. Given a set of training images, we also assign each image feature to its closest visual word, again counting the number $n_A(\omega)$ of image features assigned to each word. For each feature f with descriptor $\mathbf{d}_f$ assigned to word $\omega(\mathbf{d}_f)$, we find the two 3D points p_1, p_2, $p_1 \neq p_2$, belonging to the two nearest neighboring descriptors $\mathbf{d}_{p_1}$, $\mathbf{d}_{p_2}$ of $\mathbf{d}_f$ by exhaustively searching through all 3D point descriptors assigned to $\omega(\mathbf{d}_f)$. Applying Lowe's ratio test with $\tau = 0.7$ then yields a set of 2D-3D matches for each training image that can be used for RANSAC-based pose estimation. For each visual word ω, we count the number $n_I(\omega)$ of 2D-3D matches that are inliers to a pose estimated with RANSAC if the pose has at least 12 inliers. The probability P_i of the i^{th} image feature f_i is then given as $P_i = n_I(\omega(\mathbf{d}_{f_i}))/n_A(\omega(\mathbf{d}_{f_i}))$. Since we compare the query descriptor $\mathbf{d}_{f_i}$ against all point descriptors assigned to $\omega(\mathbf{d}_{f_i})$, the search cost c_i are given by $n_D(\omega(\mathbf{d}_{f_i}))$. Assuming a vocabulary of constant size, the probabilities and search costs for all image features can be computed efficiently in $\mathcal{O}(|\mathcal{F}|)$.

Unfortunately, solving (3) is NP-complete. Thus, we propose a simple greedy algorithm: First, determine the subset $\mathcal{F}'' = \{f_i \in \mathcal{F} | P_i > 0\}$ containing all features that might result in a correct match. Next, sort the features from $\mathcal{F}''$ in ascending order of search costs, resulting in a permutation σ s.t. $c_{\sigma_i} \leq c_{\sigma_j}$ for $i \leq j$. Finally, select the first m features according to σ such that $\sum_{i=1}^{m} P_{\sigma_i} \geq N_t$, i.e., m is chosen such the number of expected matches is at least N_t.

Let $C_{\text{greedy}} = \sum_{i=1}^{m} c_{\sigma_i}$ be the overall search cost of the greedy strategy and C_{OPT} the search cost of an optimal solution to (3). Fig. 3 shows the approximation factor, defined as the ratio $C_{\text{greedy}}/C_{\text{OPT}}$, of greedy for various values for N_t on the three datasets from Sec. 2. As can be seen, the greedy strategy performs close to optimal for the majority of query images, independently of the choice for N_t.

Vocabulary-based Prioritized Search (VPS). For the last experiment, we circumvented the problem of finding a suitable training set by using the query images to also train the probabilities P_i. In general, obtaining realistic probabilities P_i is a hard problem. However, we can easily adapt the greedy approach to the case in which only the search costs c_i are given: Instead of using the probabilities to determine the number m of features that should be used for matching, we consider all features in order of ascending search costs and stop the correspondence search once N_t actual matches have been found. Again, we use linear search through all descriptors assigned to the same word to find the two nearest neighboring points for a given query feature. Again, we accept a match with the nearest neighboring point if the ratio test with $\tau = 0.7$ is passed[2]. Once N_t matches have been found, we proceed with RANSAC-based camera pose estimation. The resulting 2D-to-3D matching framework, termed *Vocabulary-based Prioritized Search* (*VPS*), is illustrated in Fig. 1. The results from Fig. 3 indicate that the strategy of first considering features with cheap search costs performs close to optimal. Thus, we can expect our VPS framework to work well in practice.

The impact of N_t. Ideally, the threshold N_t should be

2. Limiting the nearest neighbor search to a word rather than using a kd-tree leads to different ratio test results as the 2nd nearest neighbor might fall into a different word. This in turn results in higher RANSAC times as more wrong matches are accepted.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2611662, IEEE Transactions on Pattern Analysis and Machine Intelligence

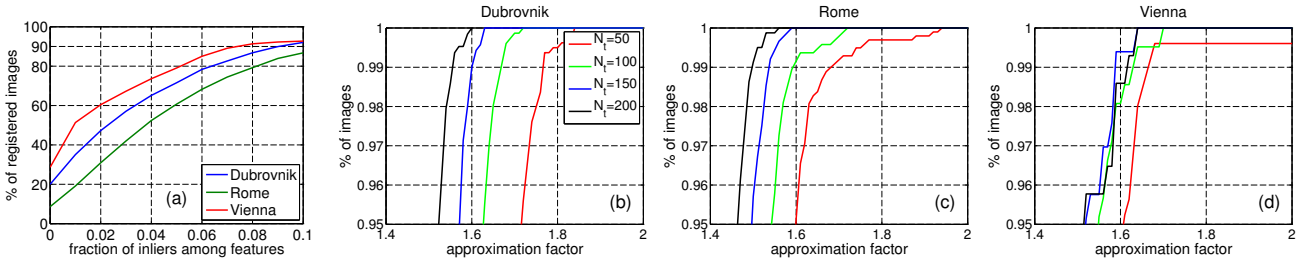

Fig. 3. (a) Cumulative histograms showing that only a small fraction of query features matches to 3D points. (b-d) The approximation factor of the greedy algorithm on the (b) Dubrovnik, (c) Rome, and (d) Vienna datasets. Though the probabilities were trained on the query images, the greedy solution is close to the optimal approximation factor of 1 for most queries. Queries for which (3) has no solution were not included in the plots.

chosen as small as possible in order to terminate the correspondence search as soon as a sufficient number of matches is found. Yet, there is the inherent risk of terminating too early: Visual words containing only few point descriptors usually correspond to sparser parts of the descriptor space, i.e., parts of the descriptor space where it is more likely that a wrong match passes the ratio test. Consequently, N_t has to be large enough to prevent terminating the correspondence search before enough correct matches are found.

Improving rejection times. If an image cannot be localized, VPS usually finds fewer than N_t matches. Since the ratio test also accepts some wrong matches, RANSAC might require many iterations to determine that no valid pose can be found. In order to limit this number of iterations, we simply assume that the inlier ratio is at least $\max(R, 12/|\mathcal{M}|)$, where $\mathcal{M}$ is the set of found matches and $R \in [0,1]$ is a parameter of our method. This enables us to limit the maximum number $k_{\max}$ of RANSAC iterations to

$$k_{\max} \leq \log(0.05)/\log(1-\max(R, 12/|\mathcal{M}|)^n) \qquad (4)$$

when using an n-point pose algorithm and requiring to find the correct model with a probability of at least 95% [4]. R should be chosen as large as possible to minimize the number of RANSAC iterations while ensuring that we can still correctly localize query images with low inlier ratios.

Point representations. So far, we have simply used *all descriptors* of each 3D point. While this offers the most accurate model of the local appearance of a 3D point, it also induces high memory requirements as some points might be associated with hundreds of descriptors. A much more compact representation can be obtained by storing a single descriptor per point, using either the *mean* or the *median descriptor* for each point. In order to limit the impact of quantization artifacts, we assign the mean and median descriptors to any words activated by all descriptor of a 3D point. While this representation results in a minimal memory footprint, it might be a poor approximation of the variation in local appearance of the points and thus might negatively impact the effectiveness of VPS. As a compromise, we propose a *mean descriptor per visual word* (*mean / vw*) representation. For each word ω activated by a given point p, we represent p by the mean of all of p's descriptors assigned to ω. The *median descriptor per word* strategy is defined accordingly.

In this paper, we rely on SIFT descriptors [25] and use a common quantization that stores each descriptor entry using a 1 byte integer value. Using floating point values to represent mean descriptors thus increases the memory requirements. To reduce the memory footprint, we obtain *integer mean* descriptors by rounding the entries to their closest integer values.

Fig. 2 compares the performance of VPS with the different point representations without early termination ($N_t = \infty$) and $R = 0$. The best effectiveness can be obtained using *all descriptors*, but all strategies outperform P2F in the number of localized images. We do not notice any loss in effectiveness when rounding mean descriptors and the *integer mean per visual word* (*int. mean / vw*) strategy achieves nearly the same effectiveness as using *all descriptors* while reducing the memory requirements for storing the descriptors by about 1/3. We thus strongly recommend to use either *all descriptors* or the *int. mean / vw* strategy.

4 Active Search

As can be seen in Fig. 2, VPS does not achieve the same localization effectiveness as the tree-based approach. This is due to quantization artifacts as VPS does not find a match if an image feature and its corresponding 3D point are assigned to different words. We could use *soft assignments* [11] and search through multiple words to recover such lost matches. Yet, this also increases the search costs for all other features. In this section, we propose an *Active Search* mechanism that recovers matches through 3D-to-2D matching and has a lower computational complexity.

Active Search. Each 2D-3D match essentially represents a hypothesis about which part of the model is visible in the query image. Consequently, we can determine a set of points that might be visible in the query image very efficiently at little additional memory overhead by using nearest neighbor search in 3D.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2611662, IEEE Transactions on Pattern Analysis and Machine Intelligence

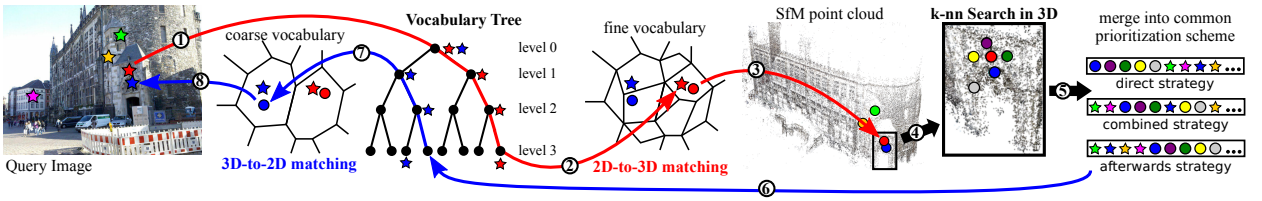

Fig. 4. Once a matching 3D point is found through 2D-to-3D search (red), *Active Search* finds candidate points for 3D-to-2D search through nearest neighbor search in 3D. The candidates are inserted into a common prioritization scheme and are later used to recover matches originally lost to due to quantization artifacts (blue).

Instead of waiting until these points are eventually considered during 2D-to-3D matching, we propose to *actively* match them against the query image using 3D-to-2D search. Obviously, we want to avoid comparing each candidate point against all features in the query image. Thus, we again use a visual vocabulary to accelerate the correspondence search. Since there are orders of magnitude fewer features in the query image than there are 3D points in the model, we need to use a coarser vocabulary for 3D-to-2D matching than the one already employed by VPS. Such a coarser quantization naturally induces fewer artifacts, allowing us recover some of the matches lost to VPS.

Fig. 4 illustrates the stages of our Active Search approach: Starting with a 2D-to-3D match found by VPS (1-2), Active Search utilizes the matching point's 3D position (3) and finds the point's N_{3D} nearest neighboring points in 3D (4). Each of these points is a candidate for 3D-to-2D search and all candidates are inserted into a prioritization scheme shared with VPS (5). Features and points are represented in Fig. 4 by stars and circles, respectively. Once the prioritization scheme considers a 3D point, Active Search matches the point against the query image. Similar to VPS, matching is performed by applying nearest neighbor search inside a visual word (7). We utilize a coarse vocabulary to reduce quantization artifacts (6). A 3D-to-2D match (8) is established if the correspondence passes the ratio test. We obtain the coarse dictionary by building a vocabulary tree [10] on top of the fine vocabulary used for VPS. Each level in the tree defines a coarser dictionary and we can select the level used for 3D-to-2D matching adaptively to regulate the search costs for the 3D points. The assignments to the leaf nodes, computed during VPS, uniquely define the corresponding words on the lower levels. The assignments to the coarser vocabularies are thus obtained at no additional costs.

Fig. 1 shows the integration of Active Search into VPS. Only 2D-to-3D matches trigger Active Search to avoid unstable configurations for pose estimation caused by finding matches only in a small part of the query image. Correspondence search is terminated once a total of N_t 2D-3D matches are found.

A common prioritization framework. As with 2D-to-3D search, we formulate 3D-to-2D matching as finding the nearest neighboring descriptors inside a set of visual words. Again, we can determine the search costs for each 3D point before performing matching by storing the number of features assigned to each word in the coarse vocabulary. As a result, we can use the same definition of priorities, i.e., an ordering based on the required number of descriptor comparisons, for both search directions. As illustrated in Fig. 4, there are three possible strategies that combine features and points into a common prioritization queue.

The *direct prioritization strategy*, used by [24], first evaluates all candidates for 3D-to-2D matching before resuming 2D-to-3D search. As for VPS, we consider the candidate points in ascending order of search costs. While the search costs of the query features increase with the size of the model, the search costs for a 3D point do not. Thus, the *direct* strategy can accelerate our approach for large-scale localization by finding many 3D-to-2D matches surrounding a 2D-to-3D match. Yet, this leads to detecting many matches in small regions of the query image, resulting in unstable configurations for camera pose estimation. [24] try to avoid this problem by selecting candidate points that are as far away as possible from the already matching points. The N_{3D} nearest neighbors of a matching 3D point selected by Active Search are by definition close to each other in 3D. The heuristic from [24] will thus not be effective in our scenario and is not applied.

The *afterwards* strategy, used by [26], first evaluates all query features and only performs 3D-to-2D search if less than N_t 2D-to-3D matches have been found. We thus perform the nearest neighbor search in 3D only after considering all query features. Again, we consider the candidate points in ascending order of search costs. Compared to the first strategy, the *afterwards* scheme usually leads to a better spatial distribution of the matches in the query image, resulting in a better pose accuracy. Yet, having to first consider all query features can negatively impact the efficiency of our method for larger models, where 2D-to-3D search is significantly slower than 3D-to-2D matching. Smaller datasets often exhibit a higher false positive matching rate for 2D-to-3D search. For such models, the *afterwards* strategy might be less effective than the *direct* scheme as the correspondence search might terminate before we attempt 3D-to-2D matching.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2611662, IEEE Transactions on Pattern Analysis and Machine Intelligence

As a compromise between the two strategies, we propose a *combined* strategy: Once we have found a set of candidate points, we insert them into the prioritization queue already containing the query features based on the search costs of the 3D points. As a result, the combined strategy will always prefer the search direction that is cheaper to evaluate. We will show in Sec. 6 that it achieves a localization accuracy similar to the *afterwards* strategy in practice. Notice that the *combined* strategy will degenerate to the *direct* strategy for very large models. This problem could be solved by using a more compact and thus sparser 3D model [26], but we do not investigate this approach.

The scalability of Active Search. Active Search not only allows us to recover correspondences that are lost to VPS due to quantization artifacts, but also enables our method to better handle larger datasets. As we consider larger and larger models, it becomes more likely to find scene points with very similar descriptors in unrelated parts of the model. Thus, the ratio test used during 2D-to-3D matching will reject more and more correct matches as too ambiguous since the descriptor space defined by the 3D points becomes denser [40]. In contrast, the density of the descriptor space defined by the query features does not depend on the 3D model and we can thus again recover such lost matches through 3D-to-2D search.

Computational complexity. In the beginning of this section, we motivated the use of Active Search by claiming that it is computationally more efficient than *soft assignments*. In the following, we prove this claim.

Let $|\mathcal{P}|$ and $|\mathcal{F}|$ be the number of 3D points and query features, respectively. We assume that the fine vocabulary used by VPS has an arbitrary but constant size. When using soft assignments, we need to consider $c-1$ additional words per feature. On average, each word contains $\mathcal{O}(|\mathcal{P}|)$ many 3D points. On average, the additional costs induced by soft assignments thus grow linearly in the size of the point cloud.

Every application of Active Search results in N_{3D} candidates for 3D-to-2D matching and Active Search is triggered at most N_t times before we terminate the correspondence search. Since both N_t and N_{3D} are constants, only a constant number of 3D points is considered for 3D-to-2D search. When matching a single point against the image, we need to compare its descriptors against at most $|\mathcal{F}|$ query features. In practice, $|\mathcal{F}|$ will be bounded by some constant independent of the model complexity $|\mathcal{P}|$, i.e., 3D-to-2D search requires only constant effort. Following a similar argument, updating the prioritization scheme also requires only constant time. As a result, the additional effort induced by Active Search is dominated by finding a constant number of nearest neighbors in 3D. This can be done in $\mathcal{O}(\log(|\mathcal{P}|))$ using a kd-tree. This shows that Active Search is computationally more efficient that using soft assignments.

Technical details. We build a vocabulary tree with branching factor 10 on top of the fine vocabulary used for VPS. We choose the level in the tree for 3D-to-2D matching based on the number of features found in the query image. We use level 2 (100 words) if $|\mathcal{F}| <$ 5k and level 3 otherwise. In contrast to the query features, each 3D point might be represented by multiple descriptors. We thus might need to search through multiple words. Let $D(p)$ be the set of descriptors of the 3D point p and let $\omega_l(D(p)) = \bigcup_{\mathbf{d}\in D(p)}\{\omega_l(\mathbf{d})\}$ be the set of corresponding visual words on level l in the tree. We define the distance between the descriptor $\mathbf{d}_f$ of a 2D feature f and $D(p)$ as

$$\text{dist}(\mathbf{d}_f, D(p)) = \min_{\mathbf{d}\in D(p), \omega_l(\mathbf{d})=\omega_l(\mathbf{d}_f)} \|\mathbf{d}_f - \mathbf{d}\|_2 \,. \quad (5)$$

During 3D-to-2D matching, we thus search for the two features f_1, f_2, $f_1 \neq f_2$, with minimal distances to $D(p)$ and accept a 3D-to-2D match between f_1 and p if $\text{dist}(\mathbf{d}_{f_1}, D(p)) < 0.6 \cdot \text{dist}(\mathbf{d}_{f_2}, D(p))$. If we have already found a 3D-to-2D match (p', f_1), we replace it with (p, f_1) if $\text{dist}(\mathbf{d}_{f_1}, D(p)) < \text{dist}(\mathbf{d}_{f_1}, D(p'))$. We do not replace 2D-to-3D matches with 3D-to-2D matches as the former are inherently more trustworthy. Since we use linear search to find the two nearest neighboring features, the search costs used to prioritize a point p are given by $\sum_{\mathbf{d}\in D(p)} n(\omega_l(\mathbf{d}))$, where $n(\omega_l)$ are the number of 2D features assigned to word ω on level l.

5 Visibility Filtering

In Sec. 4, we have made the assumption that 3D points spatially close to a matching point p are likely to lead to valid 3D-to-2D correspondences. Yet, spatial closeness does not necessarily imply the co-visibility of two 3D points. This is illustrated in Fig. 5(a): The red 3D point is among the nearest neighbors of the blue point, yet no image observing the blue point will ever observe the red point due to the geometry of the scene (dashed lines). In the following, we show how to exploit the co-visibility information encoded in the SfM model of the scene to filter out such neighboring points. In addition, we demonstrate that the co-visibility information can also be used to remove wrong matches before applying RANSAC.

The Visibility Graph. We use the co-visibility information readily available from the SfM process that is encoded in the bipartite *Visibility Graph* [2]

$$\mathcal{G} = (\mathcal{P}_\mathcal{G} \cup \mathcal{C}_\mathcal{G}, E) \,. \quad (6)$$

Each vertex $p_\mathcal{G} \in \mathcal{P}_\mathcal{G}$ corresponds to a 3D point $p \in \mathcal{P}$ in the reconstruction while each node $c_\mathcal{G} \in \mathcal{C}_\mathcal{G}$ represents one camera / database image $c \in \mathcal{C}$ in the model. As illustrated in Fig. 5(b), $\mathcal{G}$ contains an edge $e = \{p_\mathcal{G}, c_\mathcal{G}\}$ between a point node $p_\mathcal{G} \in \mathcal{P}_\mathcal{G}$ and a camera vertex $c_\mathcal{G} \in \mathcal{C}_\mathcal{G}$ iff the corresponding camera c observes the 3D point p. The set of cameras observing a point p is thus given as

$$\mathcal{C}_\mathcal{G}(p) = \{c_\mathcal{G} \in \mathcal{C}_\mathcal{G} | \{p_\mathcal{G}, c_\mathcal{G}\} \in E\} \,. \quad (7)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2611662, IEEE Transactions on Pattern Analysis and Machine Intelligence

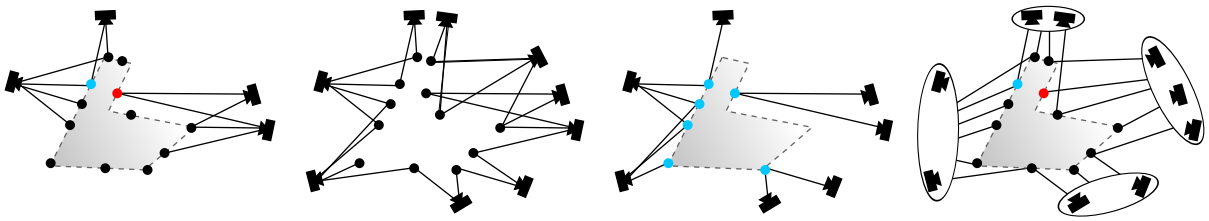

Fig. 5. (a) The red point is amongst the nearest neighbors of the blue point. Yet, there is no database image observing both points. (b) The bipartite *Visibility Graph* $\mathcal{G}$ defined by the SfM reconstruction. (c) The 3D points (blue) contained in a set $\mathcal{M}$ of 2D-3D matches define a subgraph $\mathcal{G}(\mathcal{M})$ of the Visibility Graph. (d) The new Visibility Graph $\mathcal{G}'_{\mathcal{S}}$ obtained by clustering the $k = 2$ nearest cameras and solving the set cover problem.

Two points p and p' are co-visible iff they are seen together in a camera c, i.e., iff $\mathcal{C}_{\mathcal{G}}(p) \cap \mathcal{C}_{\mathcal{G}}(p') \neq \emptyset$.

A point filter for unlikely matching candidates. Let p be the 3D point contained in a 2D-to-3D match and let $\mathcal{P}_{N_{3D}}(p)$ be the set of the N_{3D} nearest neighbors of p in 3D. As shown in Fig. 5(a), spatial proximity in 3D does not necessarily imply co-visibility. It is unlikely to observe a point $p' \in \mathcal{P}_{N_{3D}}(p)$ in the query image if p' and p are not co-visible. Thus, we only consider co-visible points from $\mathcal{P}_{N_{3D}}(p)$ for 3D-to-2D search. Our *point filter* thus removes a point $p' \in \mathcal{P}_{N_{3D}}(p)$ from consideration as a 3D-to-2D matching candidate if

$$\mathcal{C}_{\mathcal{G}}(p') \cap \mathcal{C}_{\mathcal{G}}(p) = \emptyset \ . \tag{8}$$

RANSAC pre-filter. Let $\mathcal{M} = \{(f, p) \mid f \in \mathcal{F}, p \in \mathcal{P}\}$ be a set of 2D-3D matches and let $\mathcal{P}_{\mathcal{G}}(\mathcal{M}) = \{p_{\mathcal{G}} \in \mathcal{P}_{\mathcal{G}} \mid (f, p) \in \mathcal{M}\}$ denote the set of nodes corresponding to the matching points. $\mathcal{P}_{\mathcal{G}}(\mathcal{M})$ induces a subgraph

$$\mathcal{G}(\mathcal{M}) = (\mathcal{P}_{\mathcal{G}}(\mathcal{M}) \cup \bigcup_{p_{\mathcal{G}} \in \mathcal{P}_{\mathcal{G}}(\mathcal{M})} \mathcal{C}_{\mathcal{G}}(p), E(\mathcal{M})) \tag{9}$$

consisting only of the matching points and their cameras (cf. Fig. 5(c)). In general, $\mathcal{G}(\mathcal{M})$ consists of multiple connected components. Assuming that the visibility graph perfectly encodes the co-visibility relations between the 3D points, all correct matches will be contained in the same connected component. All other components are induced by wrong correspondences. Instead of applying RANSAC-based pose estimation on all matches, our *RANSAC pre-filter* thus first identifies all connected components and then filters out all matches not contained in the largest component in $\mathcal{G}(\mathcal{M})$. One potential concern of this approach is that wrong correspondences might cluster in the visibility graph and thus form the largest connected component. In practice, we did not observe such behavior since the ratio test is very effective in removing ambiguities caused by repeating structures. Applying our *RANSAC pre-filter* thus enables us to remove many wrong matches, which in turn accelerates the RANSAC-based pose estimation stage.

Camera sets. The database images used for SfM reconstruction represent a discrete approximation to the set of all possible viewpoints. Thus, the Visibility Graph only approximates the true co-visibility relation between the points in the model. As a result, the two filtering steps proposed above might be too aggressive and filter out correctly matching points and correct matches due to weak connections in $\mathcal{G}$.

One reason for the approximate nature of the Visibility Graph is that each database image is often only matched against a limited number of other images in order to accelerate SfM [1]. Consequently, spatially close images with visual overlap might not share common 3D points (cf. top two cameras in Fig. 5(b)). In order to mitigate this effect, we propose to cluster similar cameras together. For each camera c in the reconstruction, we find the k cameras with closest camera centers. The set $\text{sim}(c)$ then comprises of c and the subset of nearest neighbors whose viewing directions, defined as the viewing ray along the principal axis of the camera, differs from c's direction by at most $60°$. Let $\mathcal{S} = \{\text{sim}(c)\}$ be the resulting set of camera clusters. Using the camera sets instead of the original images results in a new Visibility Graph

$$\mathcal{G}_{\mathcal{S}} = (\mathcal{P}_{\mathcal{G}} \cup \mathcal{S}, E_{\mathcal{S}}) \ . \tag{10}$$

$E_{\mathcal{S}}$ contains an edge $\{p_{\mathcal{G}}, \text{sim}(c)\}$ iff the original graph $\mathcal{G}$ contains an edge between $p_{\mathcal{G}}$ and some camera node $c'_{\mathcal{G}} \in \text{sim}(c)$. Compared to $\mathcal{G}$, $\mathcal{G}_{\mathcal{S}}$ contains the same number of vertices but can contain more edges. The run-time efficiency of the proposed filtering steps directly depends on the edge degree of the nodes in the visibility graph. We aim to reduce the run-time cost of the steps by selecting a minimal subset $\mathcal{S}' \subseteq \mathcal{S}$ such that every camera in the reconstruction is contained in at least one camera set $\text{sim}(c) \in \mathcal{S}'$. Similar to the point selection approach from [2], we greedily solve this set cover problem by iteratively selecting the camera set $\text{sim}(c)$ that contains the largest number of cameras not included in any of the sets selected so far. The resulting set $\mathcal{S}'$ again defines a new Visibility Graph $\mathcal{G}'_{\mathcal{S}}$ (cf. Fig. 5(d)). When using camera sets, the filtering steps are performed using $\mathcal{G}'_{\mathcal{S}}$ instead of $\mathcal{G}$.

6 Experimental Evaluation

In this section, we evaluate the performance of our proposed method on real-world data. We first eval-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2611662, IEEE Transactions on Pattern Analysis and Machine Intelligence

TABLE 1
The datasets used for evaluation.

Dataset	# DB Imgs.	# 3D Points	# Descriptors	# Query Imgs.	Mean # Query Feat.
Aachen [29]	3,047	1.54M	7.28M	369	8,648.66
Dubrovnik [2]	6,044	1.89M	9.61M	800	9,678.14
Landmarks 1k [26]	204,626	38.19M	177.82M	10k	8,378.7
Rome [2]	15,179	4.07M	21.52M	1k	7,279.91
Vienna [21]	1,324	1.12M	4.85M	266	9,707.29
Vienna (ext.) [21]	2,267	2.15M	9.03M	686	13,539.9

uate the parameters of both VPS and Active Search to derive a parameter setting that works well over a large variety of scenes. Next, we compare the resulting framework to current state-of-the-art methods. We then evaluate different camera pose solvers with the matches computed by our method. Finally, we analyze the scalability of the proposed framework.

Datasets. The Dubrovnik [2], Landmarks 1k [26], and Rome [2] datasets were reconstructed from photos obtained from Flickr and depict the old city of Dubrovnik in Crotia, the 1k most photographed landmarks found on Flickr, and 69 individual landmarks in Rome, respectively. For each dataset, query images were obtained by removing images from the reconstruction. We use the reconstructed poses as ground truth to determine the localization accuracy of our methods on the Dubrovnik dataset, where distances can be measured in meters. 3D points visible in only one remaining camera were removed as well to yield the models used for the experiments [2].

The Vienna dataset depicts three landmark scenes in the city of Vienna, Austria and was reconstructed from images taken with a single calibrated camera at regular intervals [21]. We also use an extended version of the dataset that contains three additional landmarks. For both versions, query images were obtained from Panoramio. Thus, the query images exhibit a larger difference in viewpoints and illumination conditions than on the other datasets. The Aachen dataset depicts the old inner city of Aachen, Germany and was reconstructed from images taken with a small set of cameras [29]. Query images were captured with a mobile phone over a time span of about two years.

All query images have a maximum size of 1600 x 1600 pixels and come with pre-computed SIFT descriptors [25]. Tab. 1 provides details on the datasets.

Experimental setup. We report the localization efficiency, measured as the mean time required to localize an image, the localization effectiveness, i.e., the number of localized images, and the localization accuracy, measured as the distance between the estimated camera center and the ground truth position for the camera. We follow common practice [2], [24], [26] and consider a query image as localized or registered if the best pose estimated by RANSAC has at least 12 inliers. The timing results do not include feature extraction. If not specified otherwise, we repeat every experiment 10 times to account for RANSAC's random nature. We thus report the mean number of localized images and localization times. We use the standard 6-point DLT (p6p) algorithm to compute the projection matrix from six 2D-3D matches [39] inside a SPRT-RANSAC [41] loop, followed by linear least squares optimization on all detected inliers. We use a reprojection error of $\sqrt{10}$ pixels to distinguish between inliers and outliers.

For most experiments, we use a visual vocabulary containing 100k words trained using approximate k-means clustering [11]. For Active Search, we build a vocabulary tree on top of this dictionary using a modified implementation from the FLANN library [38] that visits only a single leaf node and does not perform back-tracking. For VPS, we use a kd-tree visiting 10 leaves to compute the assignments [7].

The experiments on Landmark 1k were performed on a PC with an Intel Xeon X5650 CPU with 2.67GHz and 64 GB RAM. A PC with an Intel i7-920 CPU with 2.79GHz and 12 GB of RAM was used for all other datasets. A single CPU thread is used.

6.1 Vocabulary-based Prioritized Search

In this part, we evaluate the impact of the parameters of VPS, namely the minimal assumed inlier ratio R used to accelerate RANSAC, the threshold N_t used for early termination of the correspondence search, and the used visual vocabulary. As detailed in Sec. 3, we only use the *all descriptors* and the *integer mean per visual word* (*int. mean / vw*) point representations.

For all experiments, we use the Dubrovnik, Rome, and Vienna datasets since they form a standard benchmark for image-based localization [2], [24], [26], [33].

Faster rejection times. As can be seen in Fig. 6, the mean time required to decide that an image cannot be localized is significantly higher than the mean localization times show in Fig. 2. In order to limit the number of RANSAC iterations and thus improve the rejection times, VPS assumes that the inlier ratio is at least $\max(R, 12/|\mathcal{M}|)$, where $|\mathcal{M}|$ is the number of found matches. Fig. 6 details the impact of the parameter R on the mean rejection times and the number of rejected images for $N_t = \infty$. We can safely set $R{=}0.2$ without significantly decreasing the localization effectiveness. We thus fix $R = 0.2$ for all further experiments.

Faster localization times. VPS terminates the correspondence search after finding N_t matches in order to avoid matching all query features. Tab. 2 details the impact of the parameter N_t on both localization efficiency and effectiveness. $N_t{=}100$ offers the fastest localization times while still achieving an effectiveness similar to using all query features ($N_t = \infty$). Notice the increase in RANSAC run-times from $N_t = 50$ to $N_t = 100$ on the Vienna dataset. Since the Vienna dataset is the smallest of the three models, its 3D points induce the sparsest descriptor space, resulting

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2611662, IEEE Transactions on Pattern Analysis and Machine Intelligence

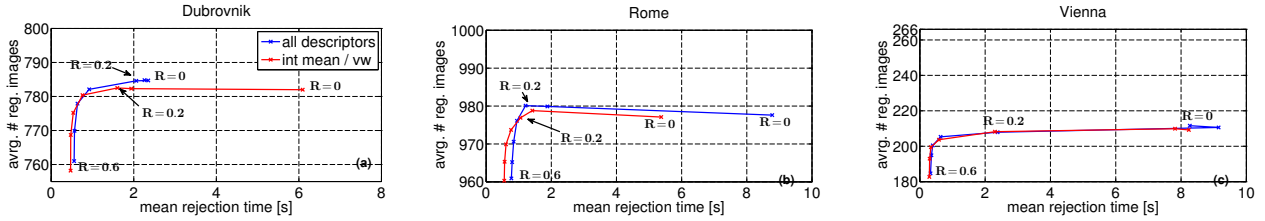

Fig. 6. The impact of enforcing a minimal inlier ratio of R on VPS's rejection times and effectiveness for (a) Dubrovnik, (b) Rome, and (c) Vienna. We used $N_t = \infty$ and values from $\{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6\}$ for R to create the curves. $R = 0.2$ reduces the rejection times considerably with only little impact on the effectiveness.

TABLE 2
Influence of the parameter N_t on the localization performance of VPS. Terminating the search after finding $N_t = 100$ matches significantly reduces the search times without sacrificing localization effectiveness.

		Dubrovnik				Rome				Vienna			
		# reg.	reg. times [s]			# reg.	reg. times [s]			# reg.	reg. times [s]		
	N_t	images	lin. search	RANSAC	total	images	lin. search	RANSAC	total	images	lin. search	RANSAC	total
all desc.	50	778.90	0.04	0.05	0.23	972.00	0.06	0.02	0.18	200.40	0.02	0.13	0.28
	100	**783.90**	0.10	0.08	**0.31**	**976.90**	0.15	0.05	**0.29**	**207.70**	0.06	0.30	**0.50**
	150	783.90	0.16	0.08	0.36	977.80	0.23	0.06	0.39	208.20	0.09	0.30	0.52
	200	784.40	0.20	0.08	0.40	979.20	0.30	0.07	0.46	208.80	0.11	0.29	0.54
	∞	784.60	0.47	0.08	0.68	980.10	0.81	0.07	0.98	207.90	0.24	0.27	0.65
int. mean	50	775.80	0.03	0.05	0.21	971.30	0.05	0.02	0.16	199.10	0.02	0.10	0.26
	100	**782.00**	0.08	0.08	**0.28**	**974.60**	0.11	0.05	**0.25**	**206.90**	0.05	0.28	**0.46**
	150	781.80	0.12	0.08	0.32	976.50	0.17	0.06	0.33	207.90	0.07	0.29	0.50
	200	782.50	0.15	0.08	0.35	976.90	0.22	0.07	0.38	208.20	0.08	0.30	0.52
	∞	782.50	0.34	0.08	0.54	976.90	0.57	0.07	0.74	208.20	0.17	0.28	0.59

in a higher false positive matching rate. Both the *all descriptor* and *integer mean per word* strategies achieve a very similar effectiveness, while the latter offers faster localization times and reduced memory requirements. For all following experiments, we thus only use the *integer mean per word* strategy and fix $N_t = 100$.

Impact of the visual vocabulary. So far, we have always used a generic vocabulary of 100k words trained on images from the Aachen model and an unrelated dataset. To determine the impact of the size of the vocabulary, we trained two new vocabularies consisting of 10k and 1M words using the same training images. As can be seen from Tab. 3, using the smaller vocabulary increases the effectiveness as it induces fewer quantization artifacts. At the same time, the search times increase since more points are stored for each word. The larger vocabulary offers faster search times but also has a reduced effectiveness due to producing more quantization artifacts and more false positive matches. Using 100k words offers the best compromise between efficiency and effectiveness.

VPS computes the visual word assignments by visiting at most 10 leaves when searching through the kd-tree. Since this approach considers less than 0.01% of all 100k words, it is unlikely to actually find the nearest word for each features. Yet, VPS still works well as long as enough related features and points are assigned to the same word. As a consequence, training a specific vocabulary from the point descriptors of each dataset has only a slight impact on the performance of VPS (cf. Tab. 3). We thus use the same generic vocabulary for all datasets.

TABLE 3
Comparing the use of generic vocabularies (G) of different sizes and dataset specific vocabularies (S).

VPS, integer mean per word, $N_t = 100$, $R = 0.2$					
	#reg.	reg. time [s]			rej.
	images	lin. search	RANSAC	total	time [s]
Dubrovnik (10k) (G)	786.70	0.31	0.05	0.40	1.46
Dubrovnik (100k) (G)	782.00	0.08	0.08	0.28	1.70
Dubrovnik (100k) (S)	781.50	0.06	0.06	0.24	0.93
Dubrovnik (1M) (G)	727.10	0.01	0.80	1.28	6.17
Rome (10k) (G)	976.70	0.47	0.04	0.55	2.64
Rome (100k) (G)	974.60	0.11	0.05	0.25	1.66
Rome (100k) (S)	970.80	0.08	0.05	0.23	1.42
Rome (1M) (G)	963.50	0.01	0.07	0.45	2.39
Vienna (10k) (G)	212.60	0.21	0.14	0.41	1.32
Vienna (100k) (G)	206.90	0.05	0.28	0.46	2.43
Vienna (100k) (S)	210.60	0.04	0.28	0.45	1.60
Vienna (1M) (G)	165.80	< 0.01	1.64	2.11	5.66

6.2 Active Search

VPS does not achieve the same effectiveness as tree-based search due to the quantization artifacts induced by using a visual vocabulary. As can be seen in Fig. 7, using the Active Search mechanism proposed in Sec. 4 enables our framework to close the gap in effectiveness between VPS and tree-based search, independently of the prioritization strategy and the choice of N_{3D}. While the additional computations required for Active Search increase the localization times, the resulting approach is still faster than the kd-tree method. As can be expected, higher values for N_{3D} result in a better effectiveness but also increase the overall run-times. The best effectiveness is obtained with the *direct* strategy, which directly evaluates all 3D-to-2D matching candidates before resuming 2D-to-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2611662, IEEE Transactions on Pattern Analysis and Machine Intelligence

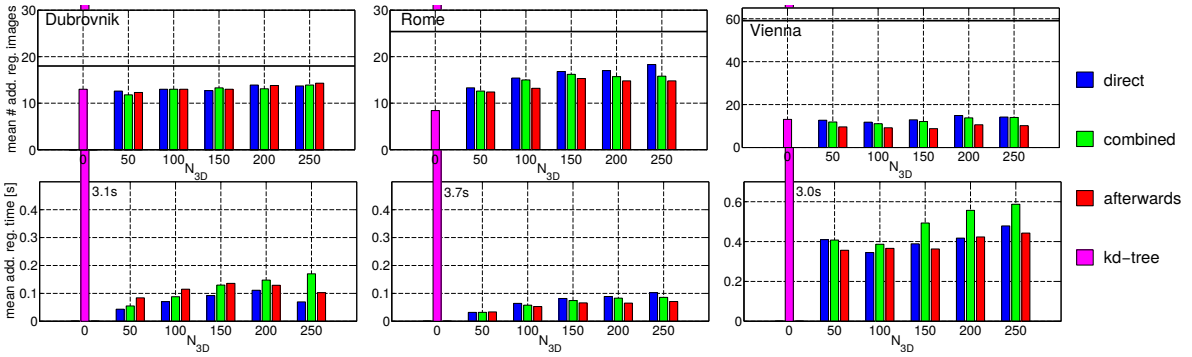

Fig. 7. Increase in (top row) the mean number of localized images and (bottom row) the mean localization times compared to VPS. The black lines denote the maximum possible increase in the number of images that can be localized. Active Search achieves a similar or better effectiveness than kd-tree search at faster localization times.

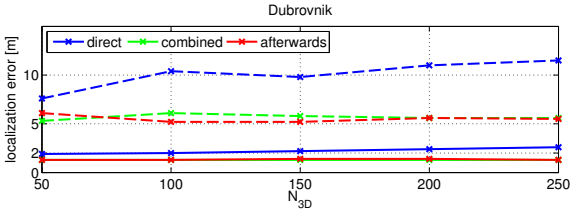

Fig. 8. The 50% (full) and 75% (dashed) quantiles of the localization errors for the three strategies.

3D search. Unfortunately, this strategy is prone to find matches in a small part of the query image, resulting in unstable configurations for pose estimation and thus a worse localization accuracy. Fig. 8 shows the localization errors of the three strategies, measured on the Dubrovnik dataset as the distance between the average estimated camera position and the ground truth position. While the localization error increases with N_{3D} for the *direct* strategy, it essentially remains constant for the other two strategies. The *combined* scheme achieves a better localization effectiveness than the *afterwards* strategy (cf. Fig. 7), since the latter only applies active search if not enough 2D-to-3D matches were found. We thus use the *combined* scheme for all subsequent experiments as it offers a good compromise between effectiveness and accuracy.

6.3 Visibility Filtering

As can be seen from Fig. 7, using Active Search increases the localization times. In order to avoid unnecessary computations, we proposed two filtering steps in Sec. 5. The *point filter* removes 3D points unlikely to yield matches from the list of candidates for 3D-to-2D search. The *RANSAC pre-filter* removes 2D-3D correspondences likely to be wrong before applying RANSAC-based camera pose estimation. Fig. 9 details the impact of both filters on the performance of our framework. As can be seen, both filtering operations significantly improve the efficiency of our approach. The *RANSAC pre-filter* yields the largest improvement on the Vienna dataset since this model exhibits the highest false positive matching rate. Consequently, the *pre-filter* has only a small impact on the Rome dataset, which exhibits a low false positive matching rate due to its denser descriptor space. Based on Fig. 9, we choose to use both filtering steps. Furthermore, we fix $N_{3D}=200$ to limit the loss in effectiveness.

Using camera sets. The co-visibility information obtained from the reconstruction process only approximates the true co-visibility relation between the points. Thus, applying the filters decreases the localization effectiveness. In Sec. 5, we proposed to cluster each camera in a model with its k most similar database images. Fig. 10 shows the effects of using such camera sets. As can be seen, the optimal choice of k is dataset dependent, but setting $k=10$ offers a good compromise between localization effectiveness and efficiency. We thus fix $k=10$, enabling our approach to essentially achieve the same effectiveness as without visibility filtering (black line in Fig. 10).

6.4 Comparison with State-of-the-Art

Tab. 4 compares our method with the current state-of-the-art for image-based localization. As in Sec. 2, P2F denotes the point-to-feature matching method from [2]. P2F+F2P is an extension that performs 2D-to-3D matching using a kd-tree if P2F fails [2]. *Vis. Prob.* is a prioritized 3D-to-2D matching approach based on co-visibility probabilities that starts from a single 2D-to-3D match [24]. The *Worldwise Pose Estimation* (WPE) method uses a mean descriptor per point and a higher threshold for the ratio test, handling the resulting higher outlier ratios by adapting RANSAC's sampling scheme [26]. WPE applies P2F if the pose cannot be estimated from 2D-to-3D matches alone. The *discriminative feature-to-point* (DF2P) matching method assumes that the position of the query image is roughly known, e.g., from GPS, and formulates descriptor matching as a classification problem [33]. The *accurate pose estimation* (APE) method uses knowledge about the gravity direction and the height of the camera to

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2611662, IEEE Transactions on Pattern Analysis and Machine Intelligence

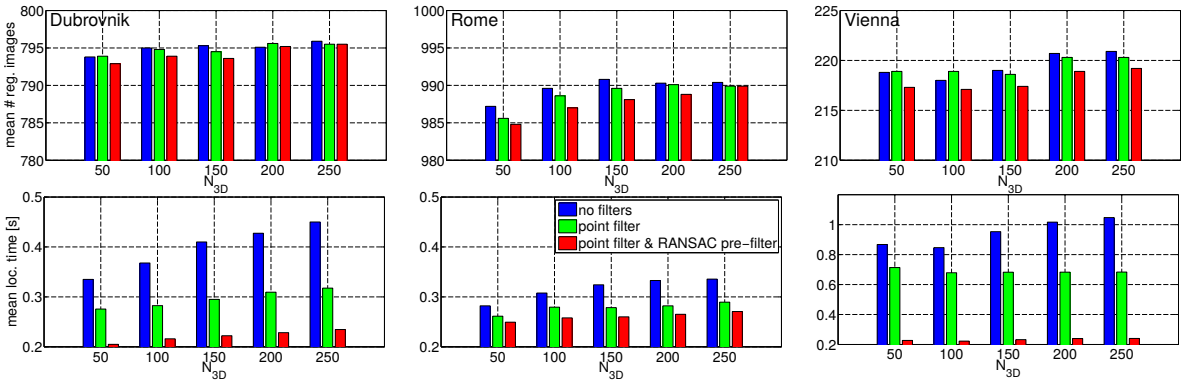

Fig. 9. The impact of applying the filtering operations proposed in Sec. 5. Both filtering steps significantly improve the efficiency of our framework at the cost of effectiveness. Active Search with the *combined* prioritization strategy was used for all experiments.

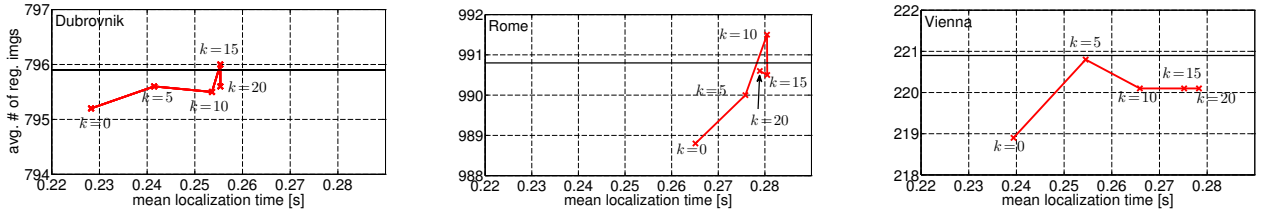

Fig. 10. Using camera sets based on clustering the k closest database images instead of the original images ($k=0$) enables us to achieve the same effectiveness as without the filtering steps (black line).

compute the pose [27]. Beside these direct matching techniques, we also compare our approach to three image retrieval-based methods: The *Voc. tree* approach generates synthetic views on a ground plane and uses the GPU to accelerate both retrieval and descriptor matching [21]. The *Hamming Voting* (HV) method uses compact binary descriptors to avoid casting votes for unrelated images during retrieval [29]. The *Graph-based Location Recognition* (GBLR) method divides the database images into places and learns classifiers to distinguish between them [30]. All three methods perform descriptor matching followed by pose estimation for the 10 top-ranked images.

As can be seen from Tab. 4, our approach achieves the fastest registration times and (with the exception of the Rome dataset) the fastest rejection times published so far. Only approaches that are at least one order of magnitude slower than our method achieve a better localization effectiveness. Our method essentially solves the Dubrovnik and Rome datasets as it fails to localize only 3 and 7 images in any of the 10 repetitions of the experiment, respectively. On the Rome and extended Vienna datasets, which are more challenging than the other datasets, our approach achieves a better effectiveness than kd-tree search. Notice that Tab. 4 reports the timings obtained with an optimized implementation of our method. This implementation optimizes the cache efficiency of our method and thus does not affect the validity of the parameter settings derived above.

6.5 Localization Accuracy

Tab. 5 reports the localization accuracy obtained with our approach. For each query image, we measure the localization error as the median of the distance between estimated and ground truth position over 10 repetitions. Besides the 6-point DLT algorithm (*p6p*) used previously, we evaluate different pose estimation strategies: *p3p / p6p* uses a 3-point algorithm (*p3p*) [4] if the focal length is stored in the EXIF tag of the query image and the *p6p* solver otherwise. The *p4pfr* solver [42] estimates the pose together with a single focal length and a radial distortion parameter from four matches. The *p4pfr-p/np* method [43] solves the same problem but handles planar and non-planer configurations separately. Since both solvers require 100ms or more, we only apply them on the inliers detected using the *p6p* solver. We denote these strategies as *p6p+p4pfr* and *p6p+p4pfr-p/np*, respectively.

Tab. 5 shows that our approach achieves a higher localization accuracy than methods relying on 3D-to-2D matching [2], [24], independent of the pose estimation strategy. This demonstrates that our approach is not only efficient and effective but also accurate. The *p6p+p4pfr* and *p6p+p4pfr-p/np* strategies provide more precise position estimates compared to the *p6p* method since the latter estimates the full internal calibration. APE [27] achieves a lower localization error using a deterministic pose estimation strategy, which could easily be combined with our matching approach.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2016.2611662, IEEE Transactions on Pattern Analysis and Machine Intelligence

TABLE 4
Compared to the current state-of-the-art, our approach achieves the fastest registration times published so far while achieving nearly the same efficiency as methods that are more than one order of magnitude slower.

Method	Aachen			Dubrovnik			Rome			Vienna			Vienna (ext.)		
	# reg. imgs	reg. time [s]	rej. time [s]	# reg. imgs	reg. time [s]	rej. time [s]	# reg. imgs	reg. time [s]	rej. time [s]	# reg. imgs	reg. time [s]	rej. time [s]	# reg. imgs	reg. time [s]	rej. time [s]
Active Search	318.8	**0.12**	**0.11**	**796.1**	**0.16**	**0.24**	**990.5**	**0.16**	1.41	221.0	**0.17**	**0.29**	**485.1**	**0.26**	**0.39**
P2F [2]	-	-	-	753	0.73	2.70	921	0.91	2.93	204	0.55	1.96	-	-	-
P2F+F2P [2]	-	-	-	753	0.70	3.96	924	0.87	4.67	205	0.54	3.62	-	-	-
Vis. Prob. [24]	-	-	-	788	0.25	0.51	977	0.27	**0.61**	219	0.40	0.49	-	-	-
DF2P [33]	-	-	-	791.8	-	-	976.4	-	-	-	-	-	-	-	-
kd-tree [7]	317	2.46	1.94	**795**	3.4	14.45	983	3.97	6.27	220	3.44	2.72	474	4.61	4.33
APE [27]	-	-	-	**798**	5.06		-	-	-	-	-	-	-	-	-
WPE [26]	-	-	-	**800**	"few seconds"		**997**	"few seconds"		-	-	-	-	-	-
Voc. tree [21]	-	-	-	-	-	-	-	-	-	165	$\leq$ **0.27**		-	-	-
HV [29]	**327**	~ 3	~ 3	786	~ 3	~ 3	984	~ 3	~ 3	**227**	~ 3	~ 3	-	-	-
GBLR [30]	**329**	-	-	**796**	-	-	**997**	-	-	-	-	-	-	-	-

TABLE 5
Localization error on the Dubrovnik dataset.

Solver / Method		total # reg. images	Mean [m]	Quantiles [m] 25%	50%	75%	90%	95%
ours	p3p / p6p	795	**14.8**	0.5	1.5	4.9	20.1	46.4
ours	p6p	**797**	30.7	**0.5**	1.3	5.0	19.2	55.3
ours	p6p+p4pfr	790	22.2	**0.5**	1.4	4.8	16.6	55.5
ours	p4pfr-p/np	792	15.3	**0.4**	1.1	**3.6**	**15.2**	55.3
P2F [2]		753	18.3	7.5	9.3	13.4	$\sim$ 18.3	-
Vis. Prob. [24]		788	34.79	0.9	3.1	11.8	-	-
APE [27]		**798**	-	-	**0.6**	-	-	$\sim$ **18.3**

TABLE 6
Results for the Landmarks 1k dataset.

Method	# reg. imgs.	reg. times [s] lin. search	reg. times [s] total	total rej. time [s]
Active Search	9534	**0.38**	**0.48**	**1.15**
VPS	8547	0.72	0.89	1.38
WPE (2D-to-3D only) [26]	$\sim$9200	"few seconds"		
WPE [26]	**9895**	"few seconds"		
HV [29]	8932	~ 3		
Hyperpoints [32]	9401	"few seconds"		

6.6 The Scalability of Active Search

As can be seen in Fig. 9, our *RANSAC pre-filter* is less effective on larger datasets. This is due to the denser descriptor spaces defined by larger sets of 3D points, which lead Lowe's ratio test to reject more wrong matches. As we consider even larger dataset, we can expect the ratio test to also reject more and more correct matches. We verify this assumption experimentally: We split the Landmarks 1k dataset into sub-models containing a few landmarks each and match each query image against the sub-model containing its landmarks. In this case, VPS localizes 91.04% of the 10k query photos. In contrast, matching the query images against the full dataset reduces the percentage of localized image to 85.47%. The results from Tab. 6 show that using Active Search helps us to better handle this problem. The density of the descriptor spaces defined by the query features does not depend on the size of the 3D model. Thus, the 3D-to-2D matching steps performed by our approach enable us to recover matches that were otherwise rejected by the ratio test. WPE [26] uses a higher threshold for the ratio test and thus obtains a better effectiveness, but Active Search still localizes a comparable number of images. There are two interesting observations: Without 3D-to-2D matching, WPE actually performs worse than Active Search. This again shows the importance of searching in both directions. Second, even though the Landmarks 1k dataset contains about one order of magnitude more points than the Rome dataset, Active Search is on average only 3 times slower.

Tab. 6 shows that Active Search significantly outperforms the image retrieval-based approach from [29] on the Landmarks 1k dataset. In addition, we compare against the Hyperpoints method from [32], which performs 2D-to-3D matching via a fine vocabulary of 16M words [31] and employs visibility filtering to select subsets of matches for pose estimation. Again, our approach localizes more images in less time.

7 Conclusion

In this paper, we have proposed a framework for image-based localization based on prioritized matching that is both efficient and effective while estimating accurate camera poses. We have shown experimentally that it is crucial to combine 2D-to-3D and 3D-to-2D search with visibility information available from every SfM model to obtain state-of-the-art results. Our approach achieves the fastest run-times published so far. Interestingly, descriptor extraction (around 0.05-0.15s using a GPU) is the main bottleneck for many datasets when using our improved implementation.

Acknowledgments

The authors thank Noah Snavely for providing datasets and Tobias Weyand for valuable discussions.

References

[1] S. Agarwal, N. Snavely, I. Simon, S. Seitz, and R. Szeliski, "Building Rome in a Day," in *ICCV*, 2009.
[2] Y. Li, N. Snavely, and D. P. Huttenlocher, "Location Recognition using Prioritized Feature Matching," in *ECCV*, 2010.
[3] S. Middelberg, T. Sattler, O. Untzelmann, and L. Kobbelt, "Scalable 6-DOF Localization on Mobile Devices," in *ECCV*, 2014.

[4] M. Fischler and R. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Comm. ACM*, vol. 24, no. 6, pp. 381–395, 1981.
[5] N. Snavely, S. Seitz, and R. Szeliski, "Photo Tourism: Exploring Photo Collections in 3D," in *SIGGRAPH*, 2006.
[6] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Lost in quantization: Improving particular object retrieval in large scale image databases," in *CVPR*, 2008.
[7] T. Sattler, B. Leibe, and L. Kobbelt, "Fast Image-Based Localization using Direct 2D-to-3D Matching," in *ICCV*, 2011.
[8] ——, "Improving Image-Based Localization by Active Correspondence Search," in *ECCV*, 2012.
[9] J. Sivic and A. Zisserman, "Video Google: A Text Retrieval Approach to Object Matching in Videos," in *ICCV*, 2003.
[10] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *CVPR*, 2006.
[11] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object Retrieval with Large Vocabularies and Fast Spatial Matching," in *CVPR*, 2007.
[12] J. Hays and A. A. Efros, "im2gps: estimating geographic information from a single image," in *CVPR*, 2008.
[13] D. Robertson and R. Cipolla, "An image-based system for urban navigation," in *BMVC*, 2004.
[14] G. Schindler, M. Brown, and R. Szeliski, "City-Scale Location Recognition," in *CVPR*, 2007.
[15] J. Knopp, J. Sivic, and T. Pajdla, "Avoding Confusing Features in Place Recognition," in *ECCV*, 2010.
[16] A. Torii, J. Sivic, T. Pajdla, and M. Okutomi, "Visual Place Recognition with Repetitive Structures," in *CVPR*, 2013.
[17] D. Chen, G. Baatz, K. Köser, S. Tsai, R. Vedantham, T. Pylvänäinen, K. Roimela, X. Chen, J. Bach, M. Pollefeys, B. Girod, and R. Grzeszczuk, "City-scale landmark identification on mobile devices," in *CVPR*, 2011.
[18] A. R. Zamir and M. Shah, "Accurate Image Localization Based on Google Maps Street View," in *ECCV*, 2010.
[19] ——, "Image Geo-localization Based on Multiple Nearest Neighbor Feature Matching using Generalized Graphs," in *PAMI*, 2014.
[20] W. Zhang and J. Kosecka, "Image based localization in urban environments," in *3DPVT*, 2006.
[21] A. Irschara, C. Zach, J.-M. Frahm, and H. Bischof, "From Structure-from-Motion Point Clouds to Fast Location Recognition," in *CVPR*, 2009.
[22] M. Aubry, B. Russel, and J. Sivic, "Painting-to-3D Model Alignment Via Discriminative Visual Elements," in *ACM Trans. Graphics*, 2013.
[23] D. Sibbing, T. Sattler, B. Leibe, and L. Kobbelt, "SIFT-Realistic Rendering," in *3DV*, 2013.
[24] S. Choudhary and P. J. Narayanan, "Visibility Probability Structure from SfM Datasets and Applications," in *ECCV*, 2012.
[25] D. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.
[26] Y. Li, N. Snavely, D. P. Huttenlocher, and P. Fua, "Worldwide Pose Estimation Using 3D Point Clouds," in *ECCV*, 2012.
[27] L. Svarm, O. Enqvist, M. Oskarsson, and F. Kahl, "Accurate Localization and Pose Estimation for Large 3D Models," in *CVPR*, 2014.
[28] B. Zeisl, T. Sattler, and M. Pollefeys, "Camera Pose Voting for Large-Scale Image-Based Localization," in *ICCV*, 2015.
[29] T. Sattler, T. Weyand, B. Leibe, and L. Kobbelt, "Image Retrieval for Image-Based Localization Revisited," in *BMVC*, 2012.
[30] S. Cao and N. Snavely, "Graph-Based Discriminative Learning for Location Recognition," *CVPR*, 2013.
[31] A. Mikulík, M. Perdoch, O. Chum, and J. Matas, "Learning vocabularies over a fine quantization," *IJCV*, vol. 103, no. 1, pp. 163–175, 2013.
[32] T. Sattler, M. Havlena, F. Radenović, K. Schindler, and M. Pollefeys, "Hyperpoints and Fine Vocabularies for Large-Scale Location Recognition," in *ICCV*, 2015.
[33] M. Donoser and D. Schmalstieg, "Discriminative Feature-to-Point Matching in Image-Based Locallization," in *CVPR*, 2014.
[34] P. F. Alcantarilla, K. Ni, L. M. Bergasa, and F. Dellaert, "Visibility Learning in Large-Scale Urban Environment," in *ICRA*, 2011.
[35] E. Eade and T. Drummond, "Scalable monocular slam," in *CVPR*, 2006.
[36] A. J. Davison, I. D. Reid, N. Molton, and O. Stasse, "MonoSLAM: Real-Time Single Camera SLAM," *PAMI*, vol. 29, no. 6, pp. 1052–1067, 2007.
[37] R. O. Castle, G. Klein, and D. W. Murray, "Video-rate localization in multiple maps for wearable augmented reality," in *ISWC*, 2008.
[38] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," in *VISAPP*, 2009.
[39] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge Univ. Press, 2004.
[40] T. Sattler, B. Leibe, and L. Kobbelt, "Towards fast image-based localization on a city-scale," in *Outdoor and Large-Scale Real-World Scene Analysis*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, vol. 7474, pp. 191–211.
[41] O. Chum and J. Matas, "Optimal Randomized RANSAC," *PAMI*, vol. 30, no. 8, pp. 1472–1482, 2008.
[42] M. Byröd, K. Josephson, and K. Åström, "Fast and Stable Polynomial Equation Solving and Its Application to Computer Vision," *IJCV*, vol. 84, no. 3, pp. 237–256, 2009.
[43] M. Bujnak, Z. Kukelova, and T. Pajdla, "New efficient solution to the absolute pose problem for camera with unknown focal length and radial distortion," in *ACCV*, 2010.

Torsten Sattler obtained his Diploma in Computer Science and his Ph.D. from RWTH Aachen University in 2008 and 2013, respectively. He is currently a postdoctoral researcher in the Computer Vision and Geometry Group at ETH Zürich. His research interest include image-based localization, image retrieval, robust (camera pose) estimation, and Structure-from-Motion. He has published papers at ICCV, ECCV, and CVPR and organized tutorials at CVPR'14 and CVPR'15.

Bastian Leibe is an associate professor at RWTH Aachen University. He holds an M.Sc. degree from Georgia Institute of Technology (1999), a Diploma degree from the University of Stuttgart (2001), and a PhD from ETH Zurich (2004), all three in Computer Science. His main research interests are in computer vision and machine learning, in particular for multi-object detection and tracking. Over the years, he received several awards for his research work, including the CVPR Best Paper Award in 2007, the ICRA Best Vision Paper Award in 2009 and 2014, the U.V. Helava Award in 2012, and an ERC Starting Grant in 2012.

Leif Kobbelt is a full professor and the head of the Institute for Computer Graphics and Multimedia at RWTH Aachen University. He received his diploma in 1992 and his Ph.D. in 1994 in Computer Science, both from the Karlsruhe Institute of Technology. His research interests include 3D reconstruction, efficient geometry processing and optimization, realistic realtime rendering and (mobile) multimedia applications. He was awarded with a number of academic prices including the Eurographics Outstanding Technical Contribution Award 2004, an ERC Advanced Grant 2013 and the Gottfried Wilhelm Leibniz Prize 2014. He is a Fellow of the Eurographics Association (2008) and a Distinguished Professor of RWTH Aachen University (2013).