This ICCV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Cascaded Parallel Filtering for Memory-Efficient Image-Based Localization

Wentao Cheng[1,2], Weisi Lin[1], Kan Chen[2] and Xinfeng Zhang[3]

[1]Nanyang Technological University, Singapore [2]Fraunhofer Singapore, Singapore
[3]University of Chinese Academy of Sciences, China

wcheng005@e.ntu.edu.sg, wslin@ntu.edu.sg, chen.kan@fraunhofer.sg, zhangxinf07@gmail.com

Abstract

Image-based localization (IBL) aims to estimate the 6DOF camera pose for a given query image. The camera pose can be computed from 2D-3D matches between a query image and Structure-from-Motion (SfM) models. Despite recent advances in IBL, it remains difficult to simultaneously resolve the memory consumption and match ambiguity problems of large SfM models. In this work, we propose a cascaded parallel filtering method that leverages the feature, visibility and geometry information to filter wrong matches under binary feature representation. The core idea is that we divide the challenging filtering task into two parallel tasks before deriving an auxiliary camera pose for final filtering. One task focuses on preserving potentially correct matches, while another focuses on obtaining high quality matches to facilitate subsequent more powerful filtering. Moreover, our proposed method improves the localization accuracy by introducing a quality-aware spatial reconfiguration method and a principal focal length enhanced pose estimation method. Experimental results on real-world datasets demonstrate that our method achieves very competitive localization performances in a memory-efficient manner.

1. Introduction

Image-based localization (IBL), *i.e.* computing the 6DOF camera pose for a query image, is a fundamental problem in many computer vision tasks. For example, IBL plays a key role in incremental Structure-from-Motion (SfM) reconstruction [13,36], visual place recognition [29], and visual navigation for autonomous vehicles [33]. IBL has witnessed tremendous advancement by means of deep learning [18, 19] and image retrieval techniques [1, 2, 34]. However, structure-based IBL [6,21,23,31,37,38,41] by directly establishing 2D-3D matches between a query image and SfM models is still the most prevailing strategy. Recent state-of-the-art methods handle the match ambiguity under high-dimensional feature representation with semantic consistency [38]. However, it remains challenging and crucial to solve this problem under compact feature representation.

A large SfM model requires prohibitive memory consumption to store tens of millions of descriptors. Meanwhile, match filtering becomes difficult, as it may contain many nearly identical descriptors. Particularly, the feature (*e.g.* visual similarity), visibility (*e.g.* point-image relationship), and geometry (*e.g.* camera pose) information in IBL leads to three interesting questions: Is it possible to improve the discriminative power of each information? How to unify them so that each can play its proper role, *i.e.* use its discriminative power to a tee? When is the appropriate phase to engage one specific information in an IBL pipeline? The accuracy is also a key issue for IBL especially in autonomous driving applications. The camera pose can be estimated by using a minimal pose solver [5] in RANSAC [12]. To achieve high accuracy, degenerate pose hypotheses should be prevented from being sampled or selected.

In this paper, we propose a cascaded parallel filtering method with respect to a binary feature representation via Hamming Embedding [15]. Using this binary feature representation, we can largely reduce the memory consumption. Meanwhile, it will introduce more ambiguities than high-dimensional feature representation, making match filtering notoriously harder. To break this dilemma, our proposed method filters wrong matches in a cascaded manner by sequentially leveraging the intrinsic feature, visibility, and geometry information. When engaging one type of information, we use a relaxed criterion to reject matches and retain a match pool that focuses on preserving correct matches. In parallel, we use a strict criterion to obtain high confident matches, which facilitate subsequent filtering steps. In feature-wise filtering, we reformulate a traditional match scoring function [16] with a bilateral Hamming ratio test to better evaluate the distinctiveness of matches. In visibility-

Method	Feature Type	Compactness	Match Filtering			Prior-free	SR
			Feature-wise	Visibility-wise	Geometry-wise		
AS [31]	SIFT	✗	Strict	✓	✗	✓	✗
WPE [21]	SIFT	✗	Relaxed	✓	✗	✓	✗
CSL [37]	SIFT	✗	Relaxed	✗	✓	✗*	✗
CPV [41]	SIFT	✗	Relaxed	✗	✓	✗*	✗
Hyperpoints [29]	SIFT	✓	Relaxed	✓	✓	✓	In RPE
EGM [23]	SIFT+Binary	✗	Relaxed	✓	✗	✓	✗
TC [6]	SIFT	✗	Relaxed	✗	✓	✓	✗
SMC [38]	SIFT	✗	Relaxed	✓	✓	✗*	✗
Our method	Binary	✓	Relaxed	✓	✓	✓	Before RPE

Table 1: Comparison between our method and other structure-based IBL methods. ✗* means that the vertical direction of camera is known in advance, SR represents Spatial Reconfiguration and RPE represents RANSAC-based Pose Estimation.

wise filtering, we explore the point-image relationship to filter wrong matches by retrieving relevant database images. Moreover, we propose a two-step match selection method by exploring the point-point relationship, which allows us to obtain substantial 2D-3D matches for computing an auxiliary camera pose. In geometry-wise filtering, we apply this auxiliary camera pose on the retained match pool to reject wrong matches by means of re-projection error.

Our method also aims to improve the localization accuracy based on two key observations. The first observation is that, correct matches that appear in sparse regions, are essential to establish a non-degenerate camera pose hypothesis. Due to the scarcity of such matches, they are usually neglected in camera pose estimation. Consequently, we propose a quality-aware spatial reconfiguration method to increase the possibility of sampling such matches in RANSAC-based pose estimation. The second observation is that, several top ranked camera pose hypotheses that have similar and realistic focal length values, are more robust than the camera pose hypothesis with the largest number of inliers. Based on this, we shift the focus to find a principal focal length value so that we can obtain a more accurate camera pose accordingly. The evaluation on benchmark datasets demonstrates that our method gets promising localization results with significantly lower memory consumption comparing with state-of-the-art methods. The source code of our method is available at https://github.com/wentaocheng-cv/cpf_localization

Related work. In recent years, numerous structure-based IBL approaches [6–9, 14, 21–23, 29–32, 35, 37, 39, 41] have been proposed. Table 1 shows an overview of state-of-the-art structure-based IBL approaches. Feature-wise filtering that mainly relies on the widely used SIFT ratio test [24] is a fundamental strategy in IBL. Efficient 2D-3D feature matching methods [9, 31] require a strict feature-wise filtering criterion to generate highly confident seed matches. The matches that are co-visible frequently with seed matches are prioritized to accelerate the matching process. Recent works [6, 21, 28, 37, 38, 41] commonly relax the feature-wise filtering criterion to preserve more correct matches and shift the filtering task to visibility or geometry tools. Li *et al*. introduce a RANSAC sampling strategy by prioritizing samples with frequent co-visibility [21]. Liu *et al*. propose a ranking algorithm by globally exploiting the visibility information on a Markov network [23]. Top ranked matches are then filtered through traditional SIFT ratio test. Camposeco *et al*. propose a geometric outlier filtering approach, in which a novel 2-point solver is able to compute an approximate camera position [6]. Assuming that the gravity direction and an approximate estimation of camera height are known, both Zeisl *et al*. and Svarm *et al*. present geometric outlier filtering approaches to handle extremely large outlier ratios [6, 37]. Toft *et al*. derive an outlier filtering method by combining the known gravity direction prior and semantic information [38].

In order to reduce the memory consumption of large SfM models, point cloud simplification approaches [7, 8, 22, 25] select a subset of representative 3D points by formulating a set cover problem. However, the reduction of points usually decreases the localization effectiveness and accuracy. Learning-based approaches implicitly compress the SfM model by training a CNN model to regress the camera pose [18, 19, 40] or scene coordinates [4]. Yet, when facing large SfM models, these approaches either have low accuracy [18, 19] or encounter a complete training failure [4]. Sattler *et al*. quantize the model descriptors into a 16M fine visual vocabulary to reduce memory consumption [29]. To handle the ill-conditioned spatial distribution, they improve the effective inlier count algorithm [14] and apply it in the RANSAC verification stage. In contrast, our proposed quality-aware spatial reconfiguration method is employed before RANSAC-based pose estimation, which allows us to obtain more non-degenerate pose hypotheses with the same number of RANSAC iterations.

2. Proposed Method

Fig. 1 shows the structured-based IBL pipeline using our method. In this section, we describe each step in detail.

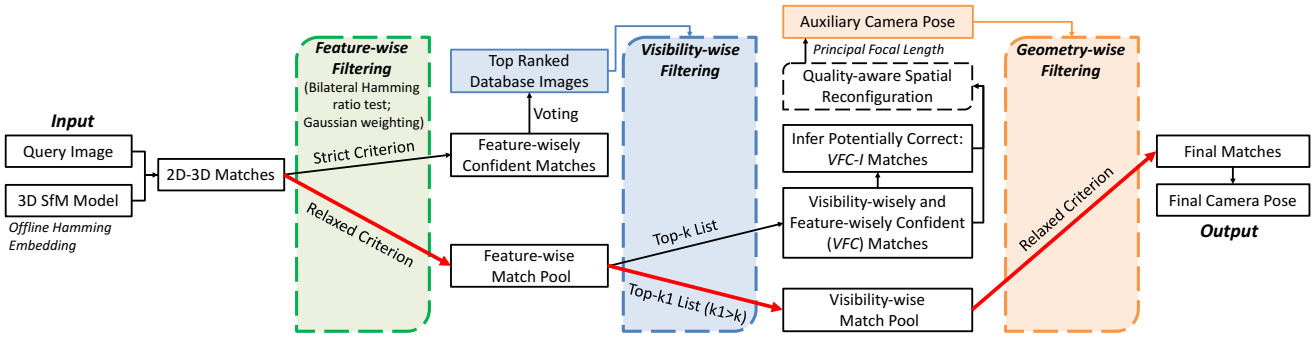

Figure 1: Overview of the localization pipeline using our cascaded parallel filtering method. A cascade of feature-, visibility- and geometry-wise filtering steps is illustrated from left to right. Two parallel tasks are applied in both the feature-wise (green) and visibility-wise (blue) filtering steps. One task with strict criterion aims for facilitating subsequent steps. Another task with relaxed criterion (in red arrow) aims for preserving correct matches.

2.1. Feature-wise Match Filtering

First, we introduce the feature-wise match filtering step. The goal of this step is twofold: 1) to retain a feature-wise match pool by rejecting obviously wrong matches, 2) to obtain a set of feature-wisely confident matches to facilitate subsequent filtering steps.

Data pre-processing. Let $\mathcal{P}$ be the 3D points in a SfM model. Each 3D point is associated with a set of SIFT descriptors. A general or specific visual vocabulary should firstly be trained using clustering techniques. In the offline stage, the descriptors of a 3D point are assigned to their closest visual words through nearest neighbor search. For efficiency, we follow [31] by representing the SIFT descriptors of a 3D point as an integer mean descriptor per visual word. Subsequently, each integer mean descriptor is converted into a compact binary signature containing B bits using Hamming Embedding [15]. Given a query image, a set of SIFT descriptors are extracted, denoted as $\mathcal{Q}$. For each descriptor $q \in \mathcal{Q}$, we first assign it to its closest visual word. Using Hamming Embedding, we also obtain the binary signature for descriptor q, denoted as s_q. For each 3D point $p \in \mathcal{P}$, if one of its associated integer mean descriptors is quantized into the same visual word with query descriptor q, a 2D-3D match can be established as $m = \{q \leftrightarrow p\}$. The Hamming distance of m can be measured as $h(s_q, s_p)$.

Bilateral Hamming ratio test. To evaluate the distinctiveness of the resultant 2D-3D matches, previous works mainly focus on the SfM model side by using a fixed Hamming distance threshold [35], Gaussian weighting [17], or density estimation [3]. Few attentions have been paid on filtering on the query image side, where the corresponding feature space is easier to distinguish correct matches due to its sparsity. Inspired from the variable radius search in [41], we propose a bilateral Hamming ratio test that operates on both the query image and the SfM model.

In order to prevent correct matches from being rejected in this step, we apply a coarse filtering scheme by using a large Hamming distance threshold τ. Therefore, for a match $m = \{q \leftrightarrow p\}$, the set of 3D points that can form a match with query descriptor q can be defined as $\mathcal{P}(q) = \{p \in \mathcal{P} | h(s_q, s_p) \leq \tau\}$. Similarly, the set of query descriptors that can form a match with 3D point p can be represented as $\mathcal{Q}(p) = \{q \in \mathcal{Q} | h(s_q, s_p) \leq \tau\}$. Our core idea is that a match should be distinctive if its corresponding Hamming distance is significantly lower than the average Hamming distance in $\mathcal{P}(q)$ and $\mathcal{Q}(p)$. To evaluate a match within the feature space of a query image, we apply an image side Hamming ratio test as follows:

$$t(m) = \frac{\sum_{j \in \mathcal{Q}(p)} h(s_j, s_p)}{h(s_q, s_p) |\mathcal{Q}(p)|^2}, \tag{1}$$

where one $|\mathcal{Q}(p)|$ in $|\mathcal{Q}(p)|^2$ is used to compute the average Hamming distance, and another is to penalize a match whose corresponding 3D point establish multiple matches. It is safe to reject a match when it is obviously ambiguous in the feature space of a query image. Therefore, we reject matches if their corresponding image side ratio test scores are smaller than a threshold φ. We observe that setting φ to 0.3 works well in practice.

Similarly, to evaluate the distinctiveness of a match within the feature space of a SfM model, we apply the model side Hamming ratio test as follows:

$$t'(m) = \frac{\sum_{j \in \mathcal{P}(q)} h(s_q, s_j)}{h(s_q, s_p) |\mathcal{P}(q)|}. \tag{2}$$

Since the term $|\mathcal{P}(q)|$ may vary dramatically with using different size of visual vocabularies, here we don't use it to penalize a match whose corresponding query descriptor can establish multiple matches with different 3D points. In addition, a large SfM model usually contains orders of magnitude more descriptors than an image. This makes the model side Hamming ratio test prone to reject correct matches by

directly setting a hard threshold. Therefore, we only apply $t'(m)$ as a soft scoring function to evaluate a match. The final bilateral Hamming ratio test can be defined as follows:

$$T(m) = \begin{cases} t'(m), & t(m) \geq \varphi \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Aggregating Gaussian weighting function. In order to strengthen the feature distinctiveness, we propose an adapted version of Gaussian weighting function [16] as follows:

$$w(h) = \begin{cases} (\frac{\sigma}{h})^2 e^{-(\frac{h}{\sigma})^2}, & 0.5\sigma < h \leq \tau \\ 4e^{-0.25}, & 0 < h \leq 0.5\sigma \\ 0, & \text{otherwise,} \end{cases} \tag{4}$$

where h is the Hamming distance of a match, and σ is usually set to one quarter of the binary feature dimension [3]. By aggregating the Gaussian weighting function, the score for a match m therefore can be computed as follows:

$$E(m) = T(m)w(h(m)). \tag{5}$$

Overall, we can retain a feature-wise match pool $\mathcal{M} = \{m|E(m) > 0\}$, which focuses on preserving correct matches. We also obtain a set of Feature-wisely Confident (*FC*) matches $\mathcal{M}_{FC} = \{m|E(m) \geq \alpha\}, \alpha > 0$.

2.2. Visibility-wise Match Filtering

Given the match sets $\mathcal{M}$ and $\mathcal{M}_{FC}$, we describe how to leverage the visibility information in a SfM model to further filter wrong matches. In particular, we aim to achieve two purposes at this stage: 1) to reject wrong matches in $\mathcal{M}$ to retain a visibility-wise match pool that well preserves correct matches, 2) to select a set of high quality matches that are substantial to derive an auxiliary camera pose for later geometry-wise filtering. The visibility information encoded in a SfM model can be represented as a bipartite visibility graph $\mathcal{G} = \{\mathcal{P}, \mathcal{D}, \mathcal{E}\}$. Each node $p \in \mathcal{P}$ represents a 3D point, and each node $d \in \mathcal{D}$ represents a database image. An edge $(p, d) \in \mathcal{E}$ exists if point p is observed in database image d. Intuitively, correct matches usually cluster in the database images that are relevant to a given query image. Thus, the problem of match filtering can be transferred as a problem of finding relevant database images.

Voting with *FC* matches. Using the visibility graph $\mathcal{G}$, a 2D-3D match $m = \{q \leftrightarrow p\}$ can cast a vote to each database image that observes point p. In order to prevent ambiguous matches from interfering the voting procedure, we only use *FC* matches to vote database images. Inspired from [29], we also enforce a locally unique voting scheme. Let $\mathcal{M}_{FC}^d = \{m = \{q \leftrightarrow p\} | m \in \mathcal{M}_{FC}, (p, d) \in \mathcal{E}\}$ be the *FC* matches that vote for database image d. We enforce that a match for database image d can be added to $\mathcal{M}_{FC}^d$ only if its corresponding query descriptor has not appeared in $\mathcal{M}_{FC}^d$ before. In addition, we only consider database images that receive at least three votes to ensure high relevancy to the query image. After accumulating the match scores for a database image, we adopt a term frequency weight in order to penalize database images that observe a large number of 3D points. Let $\mathcal{P}^d = \{p|(p, d) \in \mathcal{E}\}$ be the set of 3D points that are observed by the database image d, the voting score can be defined as follows:

$$\mathcal{S}(d) = \frac{\sum_{m \in \mathcal{M}_{FC}^d} E(m)}{\sqrt{|\mathcal{P}^d|}}. \tag{6}$$

A larger voting score inherently indicates that the corresponding database image is more relevant to a given query image, hence more likely to find correct matches. We first retrieve top-k ranked database images $d(k)$ with the largest voting scores. For a match $m \in \mathcal{M}$, we select it into the set $\mathcal{M}^{d(k)}$ if its corresponding 3D point is observed in at least one of the images in $d(k)$. Note that only visibility information is considered and we preserve both *FC* and *non-FC* matches in $\mathcal{M}^{d(k)}$. Similarly, we apply a relaxed criterion by using a larger k_1 to select another set of matches $\mathcal{M}^{d(k_1)}$, which may contain more correct matches but also are more noisy than $\mathcal{M}^{d(k)}$. $\mathcal{M}^{d(k_1)}$ will serve a visibility-wise match pool and later be filtered in Section 2.3.

Two-step match selection. Naturally, we can define the matches in $\mathcal{M}^{d(k)}$ as *Visibility-wisely Confident (VC)* matches. Due to the existence of feature-wisely ambiguous matches, *VC* matches may contain a large portion of outliers, making them difficult to be directly applied in camera pose estimation. We propose a two-step match selection method to filter *VC* matches. In the first step, we select the *FC* from *VC* matches as *Visibility-wisely and Feature-wisely Confident (VFC)* matches that can be defined as follows:

$$\mathcal{M}_{VFC}^{d(k)} = \left\{ m | m \in \mathcal{M}^{d(k)} \wedge E(m) \geq \alpha \right\}. \tag{7}$$

The *VFC* matches exhibit high confidence to be correct since they not only are observed in top ranked database images, but also are highly distinctive in feature space. The major difficulty is how to distinguish correct matches from the rest *Visibility-wisely but Not Feature-wisely Confident (VNFC)* matches that can be defined as $\mathcal{M}_{VNFC}^{d(k)} = \mathcal{M}^{d(k)} \setminus \mathcal{M}_{VFC}^{d(k)}$.

During the image voting procedure, we leverage the point-image relationship in the bipartite visibility graph $\mathcal{G}$. Now we use the point-point relationship in $\mathcal{G}$ to help us filter the *VNFC* matches. Intuitively, if a 3D point of one *VNFC* match exhibits a strong co-visibility relationship with 3D points of *VFC* matches in top ranked database images, it should be regarded as a potentially correct match. To this

Algorithm 1 Visibility-wise Match Filtering

Require: Matches $\mathcal{M}$ with feature-wise match scores $E(m)$, match score threshold α

Require: $\mathcal{M}_{VFC}^{d(k)} \leftarrow \emptyset$, $\mathcal{M}_{VFC\text{-}I}^{d(k)} \leftarrow \emptyset$,$\mathcal{M}^{d(k_1)} \leftarrow \emptyset$

/* explore **point-image** visibility */
Apply image voting with *FC* matches using Eq. 6
Retrieve top k and k_1 database images $d(k)$ and $d(k_1)$
Select all matches in $d(k_1)$ as $\mathcal{M}^{d(k_1)}$ for visibility-wise match pool
Separate *VFC* matches $\mathcal{M}_{VFC}^{d(k)}$ and *VNFC* matches $\mathcal{M}_{VNFC}^{d(k)}$ using Eq. 7
/* explore **point-point** visibility */
for all $d \in d(k)$ **do**
    Compute the number of *VFC* matches ω_{VFC}^{d}
    Compute the number of *VNFC* matches ω_{VNFC}^{d}
    for all $m \in \mathcal{M}_{VNFC}^{d}$ **do**
        Compute the updated match score $E^{'}(m)$ using Eq. 8
for all $m \in \mathcal{M}_{VNFC}^{d(k)}$ **do**
    if $E^{'}(m) \geq \alpha$ **then**
        $\mathcal{M}_{VFC\text{-}I}^{d(k)} \leftarrow \mathcal{M}_{VFC\text{-}I}^{d(k)} \cup \{m\}$
return $\mathcal{M}_{VFC}^{d(k)} \cup \mathcal{M}_{VFC\text{-}I}^{d(k)}$ and $\mathcal{M}^{d(k_1)}$

end, we engage the second step match selection to infer potentially correct matches from *VNFC* matches. For each database image $d \in d(k)$, we first count the number of *VFC* matches and *VNFC* matches, which we call ω_{VFC}^{d} and ω_{VNFC}^{d} respectively. If *VFC* matches occupy a larger portion compared with *VNFC* matches in one database image, each *VNFC* match should receive stronger promotion from *VFC* matches respectively. Therefore, for an *VNFC* match, we compute its updated match score as follows:

$$E^{'}(m) = E(m) + \sum_{d \in d(k)} \frac{\alpha}{2} \ln(1 + \frac{\omega_{VFC}^{d}}{\omega_{VNFC}^{d}}). \quad (8)$$

The larger the updated match score, the more likely that corresponding *VNFC* match is correct. Using the previous match score threshold α, we can select a set of potentially correct matches from *VNFC* matches. Since these potentially correct matches are mainly inferred by exploring the visibility information with *VFC* matches, we call them *VFC-I* matches and they can be defined as follows:

$$\mathcal{M}_{VFC-I}^{d(k)} = \left\{ m | m \in \mathcal{M}_{VNFC}^{d(k)} \wedge E^{'}(m) \geq \alpha \right\}. \quad (9)$$

Therefore, the matches that we select from $\mathcal{M}^{d(k)}$ are the union of *VFC* and *VFC-I* matches. Algorithm 1 illustrates the process of visibility-wise match filtering.

2.3. Geometry-wise Match Filtering

In this section, we describe how to use the obtained *VFC* and *VFC-I* matches to compute an auxiliary camera pose, which facilitates geometry-wise match filtering for the visibility-wise match pool $\mathcal{M}^{d(k_1)}$.

Quality-aware spatial reconfiguration. A common way to estimate the camera pose is to use pose solvers inside RANSAC loops. The quality of input 2D-3D matches, *i.e.* the inlier ratio, is an essential factor for robust and efficient camera pose estimation. It is also important to ensure that the input matches have a uniform spatial distribution, especially when the majority of input matches cluster in a highly textured region as shown in Fig. 2. Correct matches, rare but critical, in poorly textured regions are unlikely to be sampled in the RANSAC hypothesis stage. This will significantly reduce the localization accuracy due to the difficulty of obtaining a non-degenerate pose hypothesis.

Our goal is to obtain a set of matches that simultaneously have a large inlier ratio and a uniform spatial distribution by selecting from *VFC* and *VFC-I* matches. To this end, we first divide the query image into 4 by 4 equally-sized bins, denoted as $\mathcal{B}$. The *VFC* and *VFC-I* matches are then quantized into $\mathcal{B}$ according to the image coordinates of their associated 2D query descriptors. To make the spatial distribution of selected matches more uniform, we apply a spatial reconfiguration method to penalize dense bins with more quantized matches and emphasize sparse bins with fewer quantized matches. Let N_b be the number of matches that are quantized into bin $b \in \mathcal{B}$. Let R_b be the proportion of matches that can be selected from bin b, the spatial reconfiguration can be realized by computing R_b as follows:

$$R_b = \frac{\sqrt{N_b}}{\sum_{i \in \mathcal{B}} \sqrt{N_i}}. \quad (10)$$

To achieve an efficient camera pose estimation, we limit that overall at most N matches can be selected. Accordingly, for each bin b, the match selection quota is $R_b N$.

We first select the *VFC* matches with larger match scores according to each bin's selection quota. After that, if there exist bins that still do not reach the selection quotas, we then select the *VFC-I* matches from these bins. Note that the *VFC-I* matches exhibit inferior quality than the *VFC* matches because of their confidence in only visibility. To ensure high quality of selected matches, the *VFC* matches should be dominant. Suppose the number of selected *VFC* matches is N_{VFC}, we restrict that at most βN_{VFC} *VFC-I* matches can be selected. In this work, we set β to 0.33.

Auxiliary camera pose with principal focal length. We then use the selected matches after quality-aware spatial reconfiguration to compute an auxiliary camera pose. Assuming a general scenario when the focal length of a given query image is unknown, we can adopt a 4-point pose solver

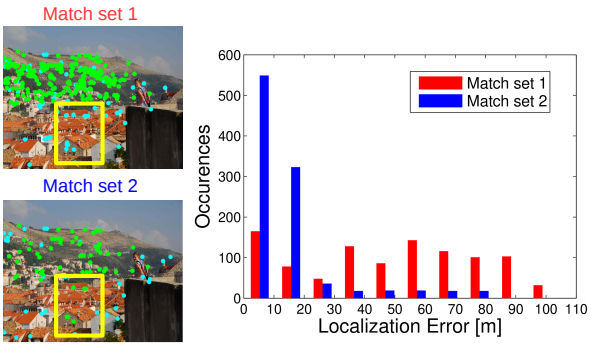

Figure 2: The influence of a uniform spatial distribution for matches. **Left top:** Original match set with 242 inliers shown in green and 64 outliers shown in cyan (inlier ratio is 0.79), matches are clustered in mountain area; **Left bottom:** a selection from original match set by applying spatial reconfiguration, this selection has 63 inliers and 31 outliers (inlier ratio is 0.67), matches are more uniformly distributed over the image; **Right:** Localization error statistics with these two match sets by running 1000 camera pose estimation trials. **Yellow box:** the inside correct but sparse matches are emphasized in match set 2.

(P4P) [5] to estimate the extrinsic calibration and the focal length. In RANSAC-based camera pose estimation, the estimated camera pose usually is the pose hypothesis that is supported by the largest number of inliers. However, we notice that this strategy becomes unreliable when few correct matches exist. In such case, a co-planar degenerated sample may result in that the estimated camera will lie far away from the scene with an unrealistic focal length. To tackle this unreliability problem, we propose a statistical verification scheme to find a reliable camera pose. Let ε be the largest number of inliers of a pose hypothesis after running a certain number of RANSAC+P4P loops. We store the top-10 pose hypotheses, whose corresponding inliers are more than 0.7ε. For a successful localization, we notice that most of the top hypotheses have numerically close focal length values. These focal length values, instead of the one with largest number of inliers, provide us a more stable and reliable camera pose estimation. Inspired from RANSAC variants [10] that vote for optimal parameter values, we propose to select the pose hypothesis whose focal length is the median value among the top pose hypotheses. We define the selected pose hypothesis as an auxiliary camera pose, and its corresponding focal length as principal focal length f.

Filtering with auxiliary camera pose. The computed auxiliary camera pose exhibits sufficient accuracy. Using it to recover potentially correct matches back can further improve the localization accuracy. We apply the auxiliary camera pose on the visibility-wise match pool $\mathcal{M}^{d(k_1)}$ to realize the geometry-wise filtering. We define a relaxed re-projection error threshold θ in case rejecting potentially correct matches. As such, a match can be selected as a potentially correct match if the re-projection error with respect to the auxiliary camera pose is below θ. In this work, we choose a threshold of 10 pixels.

Table 2: Summarization of the used datasets.

Dataset	Database Images	3D Points	Query Images
Dubrovnik [22]	6,044	1.89M	800
RobotCar Seasons [33]	20,862	6.77M	11,934
Aachen Day-Night [33]	4,328	1.65M	922
SF-0 [21, 34]	610,773	30M	442

Final camera pose estimation. The matches selected by the auxiliary camera pose exhibit both high quality and high quantity. In addition, we have also obtained a reliable focal length value f. Based on these, we can directly apply a 3-point pose solver (P3P) [20], which is much more efficient than 4-point pose solvers, to compute the final camera pose.

3. Experiments

3.1. Datasets and Evaluation Metrics

We evaluate our proposed method on four benchmark datasets as summarized in Table 2. For the Dubrovnik dataset, we adopt the same evaluation metric used in related works [6, 22, 23, 30, 31, 37, 41]. A query image is considered as successfully registered or localized if the best camera pose after RANSAC has at least 12 inliers. The localization accuracy on the Dubrovnik dataset can be measured as the distance between estimated camera center position and the ground truth camera center position of query image. The RobotCar Seasons [33] dataset was reconstructed from images that were captured with cameras mounted on an autonomous vehicle. This dataset covers a wide range of condition changes, *e.g.* weather, seasons, day-night, which make image-based localization on this dataset challenging. The ground truth camera poses of query images were obtained by aligning all 49 SfM sub-models to LIDAR point clouds. The query images of the Aachen Day-Night dataset consist of 824 images in day condition and 98 images in night condition. For the RobotCar Seasons and Aachen Day-Night datasets, we follow the evaluation metric in [33] and report the percentage of query images localized within Um and $V°$ from ground truth camera poses. To evaluate under different levels of localization accuracy, we use the three accuracy intervals defined in [33] as follows: High-precision $(0.25m, 2°)$, Medium-precision $(0.5m, 5°)$ and Coarse-precision $(5m, 10°)$. For the large-scale SF-0 dataset [21], we use the evaluation package provided by [34] which contains reference camera poses for 442 query images.

Table 3: The comparison between our method and state-of-the-art methods on the Dubrovnik dataset.

Method	Error Quartiles [m]			Localized
	25%	50%	75%	images
EGM	0.24	0.70	2.67	794
TC	**0.22**	1.07	2.99	**800**
AS	0.40	1.40	5.30	796
Our method	**0.22**	**0.64**	**2.16**	794

3.2. Implementation Details

For the Dubrovnik dataset, we use the same 10k general visual vocabulary trained by [31]. For the RobotCar Seasons and Aachen Day-Night datasets, we train a specific 10k visual vocabulary on all upright RootSIFT descriptors found in 1000 randomly selected database images in the reference SfM model. For the large-scale SF-0 dataset, we train a 50k specific visual vocabulary on all integer mean RootSIFT descriptors. For the Dubrovnik and RobotCar Seasons datasets, we set $B = 64$, $\tau = 19$ and $\alpha = 0.8$ for feature-wise filtering. In the visibility-wise filtering step, we set $k = 20$ and $k_1 = 100$. In the geometry-wise filtering step, we set $N = 100$. For the Aachen Day-Night dataset, we find that by keeping other parameters unchanged and setting $\tau = 16$ and $k_1 = 50$ can obtain sufficient correct 2D-3D matches. Due to dramatically different characteristics between large-scale SF-0 and the above three medium-scale datasets, we adjust B to 128, τ to 32 and α to 0.4 accordingly. For computing the auxiliary camera pose and the final camera pose, we run both 1000 RANSAC iterations. For a fair comparison on the Dubrovnik dataset, we use a threshold of 4 pixels for final pose estimation. For a fair comparison on the RobotCar Seasons and Aachen Day-Night dataset, we use a 3-point pose solver to compute the auxiliary camera pose and a threshold of 4 pixels for final pose estimation. All experiments were conducted with a single CPU thread on a PC with an Intel i7-6800K CPU with 3.40 GHz and 32 GB RAM.

3.3. Comparison with State-of-the-art

On the Dubrovnik dataset, we compare against three prior-free state-of-the-art approaches: Efficient Global Matching (EGM) [23], Active Search (AS) [31] and Toroidal Constraint (TC) [6]. On the other three datasets in which images are captured on the street, we include the comparison with approaches that use the knowledge about gravity direction. Concretely, we compare with City-scale Localization (CSL) [37], Camera Pose Voting (CPV) [41] and Semantic Match Consistency (SMC) [38]. For comprehensiveness, we also compare with two retrieval-based approaches, namely DenseVLAD [2] and NetVLAD [1].

Evaluation on medium-scale datasets. Table 3 shows the comparison on the Dubrovnik dataset. As can be seen, our method outperforms state-of-the-art methods in localization accuracy. In the meantime, we maintain a very competitive effectiveness, i.e., the number of successfully localized query images. Table 4 shows the percentage of query images localized within three pose accuracy intervals of our proposed method compared with state-of-the-art localization methods on the RobotCar Seasons and Aachen Day-Night datasets. Our method achieves the second best localization performance on the RobotCar Seasons dataset. Interestingly, our method significantly outperforms CSL that requires prior knowledge about the gravity direction. SMC relies on a neural network for semantic segmentation. Note that the training data used in SMC includes several manually labelled images from the original RobotCar dataset [26]. On the Aachen Day-Night dataset, our method achieves the best localization performance in most cases.

Table 4: The percentage of query images localized within three pose accuracy intervals of our proposed method compared with state-of-the-art localization methods on the RobotCar Seasons and Aachen Day-Night datasets. **red** and **blue** represent the **best** and **second-best** methods, and the asterisk symbol represents using knowledge about the gravity direction.

RobotCar Seasons		
	All Day	All Night
m	.25 / 0.5 / 5.0	.25 / 0.5 / 5.0
deg	2 / 5 / 10	2 / 5 / 10
AS	35.6 / 67.9 / 90.4	0.9 / 2.1 / 4.3
DenseVLAD	7.7 / 31.3 / 91.2	1.0 / 4.5 / **22.7**
NetVLAD	6.4 / 26.3 / 91.0	0.4 / 2.3 / 16.0
CSL*	45.3 / 73.5 / 90.1	0.6 / 2.6 / 7.2
SMC*	**50.6** / **79.8** / **95.1**	**7.6** / **21.5** / **45.4**
Our method	**48.0** / **78.0** / **94.2**	**3.4** / **9.5** / 17.0
Aachen Day-Night		
	Day	Night
m	.25 / 0.5 / 5.0	.25 / 0.5 / 5.0
deg	2 / 5 / 10	2 / 5 / 10
AS	**53.7** / **83.7** / **96.6**	19.4 / 30.6 / 43.9
DenseVLAD	0.0/ 0.1 / 22.8	0.0/ 2.0 / 14.3
NetVLAD	0.0 / 0.2 / 18.9	0.0 / 2.0 / 12.2
CSL*	52.3 / 80.0 / 94.3	**24.5** / **33.7** / **49.0**
SMC*	-	-
Our method	**76.7** / **88.6** / **95.8**	**25.5** / **38.8** / **54.1**

Memory consumption. We also investigate the memory consumption required by our method and other methods. Without losing generality, we only compare against AS which is the most memory-efficient in state-of-the-art structure-based localization methods. Table 5 shows the detailed comparison. Comparing with AS, our method requires significantly lower memory consumption. The reason for the memory reduction is that our method only needs to store a compact binary signature (8-bytes when $B = 64$) per visual word for each 3D point. While AS needs to store

Table 5: The memory consumption (in GB) comparison between our method and other state-of-the-art methods.

Method	Memory Consumption		
	Dubrovnik	RobotCar	Aachen
AS	0.75	2.72	0.76
Our method	**0.14**	**0.52**	**0.14**

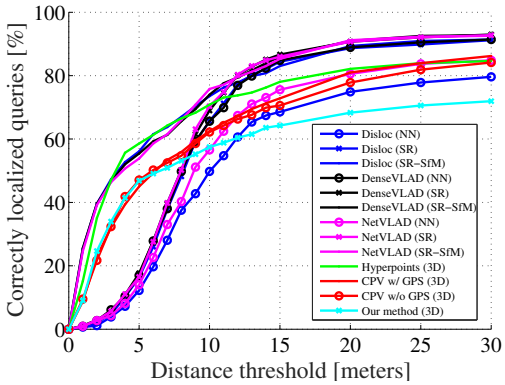

Figure 3: The experimental results on the SF-0 dataset.

an integer mean (128-bytes) of SIFT descriptors per visual word for each 3D point. Overall speaking, our method is memory-efficient and achieves very competitive localization performance on medium-scale datasets.

Evaluation on large-scale SF-0. Fig. 3 shows the results on the SF-0 dataset. We mainly compare with two structure-based methods: CPV and Hyperpoints [29]. Note that the SR-SfM scheme in Fig. 3 usually takes several minutes to process one query image. Comparing with CPV using full descriptors, our method achieves competitive results for thresholds of 5m or less. Yet, our method does not perform better than Hyperpoints, in which a fine vocabulary is used and more suitable for large-scale location recognition problems. In addition, using the GPS tags available in SF-0 would be beneficial to remedy the drawback of our method for coarse-level localization (5∼30m).

3.4. Ablation Study

We conduct an ablation study on the Dubrovnik dataset to evaluate the impact of key components in our method. The match score threshold in the two-step match selection method is heavily related to the the bilateral Hamming ratio test. For simplicity, we arguably evaluate these two components together. To this end, we first implement a baseline voting method that filters wrong matches established from binary signatures. In the baseline implementation, a match is evaluated by Eq. 4. Then, we select all matches from top-20 ranked database images for computing the auxiliary camera pose, and we select all matches from top-100 ranked database images to obtain the visibility-wise match pool. Other components in our method remain unchanged.

Table 6: The ablation study conducted on the Dubrovnik dataset.

Setting	Error quartiles [m]			Localized images
	25%	50%	75%	
Baseline Voting	0.25	0.69	2.19	778
w/o QSR	0.26	0.74	2.53	793
w/o PFL	0.31	0.80	2.70	**794**
Our full method	**0.22**	**0.64**	**2.16**	**794**

We test with multiple Hamming distance thresholds in Eq. 4, and the baseline implementation achieves the best performance when setting the threshold to 11. As shown in Table 6, our method can localize 16 more query images than the baseline implementation. This indicates that the combination of the bilateral Hamming ratio test and the two-step match selection method is beneficial for better filtering.

We also conduct an experiment to investigate the impact of the quality-aware spatial reconfiguration (QSR) method and the principal focal length estimation (PFL) in Section 2.3. We first disable QSR and select the same number of *VFC* and *VFC-I* matches as when QSR enabled. Note that the matches in QSR disabled are selected with the largest match scores. As shown in Table 6, QSR significantly improves the localization accuracy. This indicates that obtaining a set of uniformly distributed matches before RANSAC-based pose estimation is essential for accurate IBL. To examine the benefit of PFL, we conduct an experiment with traditional RANSAC scheme when computing the auxiliary camera pose, *i.e.* the best camera pose is the one with largest number of inliers. We can see that PFL also significantly improves the localization accuracy. This indicates that the auxiliary camera pose selected with PFL is more robust to apply geometry-wise match filtering.

4. Conclusion

In this paper, we have presented a cascaded parallel filtering method for memory-efficient image-based localization. Our method contains a cascade of feature-, visibility- and geometry-based filters, in which two parallel criteria are applied for preserving correct matches and obtaining high quality matches. The localization accuracy is improved by quality-aware spatial reconfiguration and principal focal length methods. Comprehensive experiments on challenging real-world datasets demonstrate the benefit of our method. Further improvements could be achieved by incorporating CNN-based feature descriptors [11] or hierarchical localization schemes [27].

Acknowledgements: This research was supported by Singapore Ministry of Education Tier-2 Fund MOE2016-T2-2-057(S) and the National Research Foundation, Prime Minister's Office, Singapore under its International Research Centres in Singapore Funding Initiative.

References

[1] Relja Arandjelović, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proc. CVPR*, 2016. 1, 7

[2] Relja Arandjelović and Andrew Zisserman. All about vlad. In *Proc. CVPR*, 2013. 1, 7

[3] Relja Arandjelović and Andrew Zisserman. Dislocation: Scalable descriptor distinctiveness for location recognition. In *Proc. ACCV*, 2014. 3, 4

[4] Eric Brachmann and Carsten Rother. Learning less is more-6d camera localization via 3d surface regression. In *Proc. CVPR*, 2018. 2

[5] Martin Bujnak, Zuzana Kukelova, and Tomas Pajdla. A general solution to the p4p problem for camera with unknown focal length. In *Proc. CVPR*, 2008. 1, 6

[6] Federico Camposeco, Torsten Sattler, Andrea Cohen, Andreas Geiger, and Marc Pollefeys. Toroidal constraints for two-point localization under high outlier ratios. In *Proc. CVPR*, 2017. 1, 2, 6, 7

[7] Song Cao and Noah Snavely. Minimal scene descriptions from structure from motion models. In *Proc. CVPR*, 2014. 2

[8] Wentao Cheng, Weisi Lin, Xinfeng Zhang, Michael Goesele, and Ming-Ting Sun. A data-driven point cloud simplification framework for city-scale image-based localization. *IEEE Transactions on Image Processing*, 26(1):262–275, 2017. 2

[9] Siddharth Choudhary and PJ Narayanan. Visibility probability structure from sfm datasets and applications. In *Proc. ECCV*, 2012. 2

[10] Ondřej Chum and Jiří Matas. Optimal randomized ransac. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(8):1472–1482, 2008. 6

[11] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-net: A trainable cnn for joint detection and description of local features. In *Proc. CVPR*, 2019. 8

[12] Martin A Fischler. Random sampling concensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981. 1

[13] Michael Goesele, Noah Snavely, Brian Curless, Hugues Hoppe, and Steven M Seitz. Multi-view stereo for community photo collections. In *Proc. ICCV*, 2007. 1

[14] Arnold Irschara, Christopher Zach, J-M Frahm, and Horst Bischof. From structure-from-motion point clouds to fast location recognition. In *Proc. CVPR*, 2009. 2

[15] Hervé Jégou, Matthijs Douze, and Cordelia Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *Proc. ECCV*, 2008. 1, 3

[16] Hervé Jégou, Matthijs Douze, and Cordelia Schmid. On the burstiness of visual elements. In *Proc. CVPR*, 2009. 1, 4

[17] Hervé Jégou, Florent Perronnin, Matthijs Douze, Jorge Sánchez, Patrick Perez, and Cordelia Schmid. Aggregating local image descriptors into compact codes. *IEEE transactions on pattern analysis and machine intelligence*, 34(9):1704–1716, 2012. 3

[18] Alex Kendall and Roberto Cipolla. Geometric loss functions for camera pose regression with deep learning. In *Proc. CVPR*, 2017. 1, 2

[19] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proc. ICCV*, 2015. 1, 2

[20] Laurent Kneip, Davide Scaramuzza, and Roland Siegwart. A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In *Proc. CVPR*, 2011. 6

[21] Yunpeng Li, Noah Snavely, Dan Huttenlocher, and Pascal Fua. Worldwide pose estimation using 3d point clouds. In *Proc. ECCV*. 2012. 1, 2, 6

[22] Yunpeng Li, Noah Snavely, and Daniel P Huttenlocher. Location recognition using prioritized feature matching. In *Proc. ECCV*. 2010. 2, 6

[23] Liu Liu, Hongdong Li, and Yuchao Dai. Efficient global 2d-3d matching for camera localization in a large-scale 3d map. In *Proc. ICCV*, 2017. 1, 2, 6, 7

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

[25] Simon Lynen, Torsten Sattler, Michael Bosse, Joel Hesch, and Roland Siegwart. Get out of my lab: Large-scale, real-time visual-inertial localization. In *Robotics: Science & Systems*, 2015. 2

[26] Will Maddern, Geoffrey Pascoe, Chris Linegar, and Paul Newman. 1 year, 1000 km: The oxford robotcar dataset. *The International Journal of Robotics Research*, 36(1):3–15, 2017. 7

[27] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *Proc. CVPR*, 2019. 8

[28] Paul-Edouard Sarlin, Frédéric Debraine, Marcin Dymczyk, Roland Siegwart, and Cesar Cadena. Leveraging deep visual descriptors for hierarchical efficient localization. In *Proc. CoRL*, 2018. 2

[29] Torsten Sattler, Michal Havlena, Filip Radenovic, Konrad Schindler, and Marc Pollefeys. Hyperpoints and fine vocabularies for large-scale location recognition. In *Proc. ICCV*, 2015. 1, 2, 4, 8

[30] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Fast image-based localization using direct 2d-to-3d matching. In *Proc. ICCV*, 2011. 2, 6

[31] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Improving image-based localization by active correspondence search. In *Proc. ECCV*. 2012. 1, 2, 3, 6, 7

[32] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Efficient & effective prioritized matching for large-scale image-based localization. *IEEE transactions on pattern analysis and machine intelligence*, 39(9):1744–1756, 2017. 2

[33] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, Fredrik Kahl, and Tomas Pajdla. Benchmarking 6dof outdoor visual localization in changing conditions. In *Proc. CVPR*, 2018. 1, 6

[34] Torsten Sattler, Akihiko Torii, Josef Sivic, Marc Pollefeys, Hajime Taira, Masatoshi Okutomi, and Tomas Pajdla. Are

large-scale 3d models really necessary for accurate visual localization? In *Proc. CVPR*, 2017. 1, 6

[35] Torsten Sattler, Tobias Weyand, Bastian Leibe, and Leif Kobbelt. Image retrieval for image-based localization revisited. In *Proc. BMVC*, 2012. 2, 3

[36] Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. In *ACM transactions on graphics (TOG)*, volume 25, pages 835–846. ACM, 2006. 1

[37] Linus Svärm, Olof Enqvist, Fredrik Kahl, and Magnus Oskarsson. City-scale localization for cameras with known vertical direction. *IEEE transactions on pattern analysis and machine intelligence*, 39(7):1455–1461, 2017. 1, 2, 6, 7

[38] Carl Toft, Erik Stenborg, Lars Hammarstrand, Lucas Brynte, Marc Pollefeys, Torsten Sattler, and Fredrik Kahl. Semantic match consistency for long-term visual localization. In *Proc. ECCV*, 2018. 1, 2, 7

[39] Ngoc-Trung Tran, Dang-Khoa Le Tan, Anh-Dzung Doan, Thanh-Toan Do, Tuan-Anh Bui, Mengxuan Tan, and Ngai-Man Cheung. On-device scalable image-based localization via prioritized cascade search and fast one-many ransac. *IEEE Transactions on Image Processing*, 28(4):1675–1690, 2019. 2

[40] Florian Walch, Caner Hazirbas, Laura Leal-Taixe, Torsten Sattler, Sebastian Hilsenbeck, and Daniel Cremers. Image-based localization using lstms for structured feature correlation. In *Proc. ICCV*, 2017. 2

[41] Bernhard Zeisl, Torsten Sattler, and Marc Pollefeys. Camera pose voting for large-scale image-based localization. In *Proc. ICCV*, 2015. 1, 2, 3, 6, 7