Cascaded Parallel Filtering for Memory-Efficient Image-Based Localization

Wentao Cheng¹,², Weisi Lin¹, Kan Chen² and Xinfeng Zhang³
¹Nanyang Technological University, Singapore ²Fraunhofer Singapore, Singapore ³University 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-
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 methods. 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.

Table 1: Comparison between our method and other structure-based IBL methods.

<table>
<thead>
<tr>
<th>Method</th>
<th>Feature Type</th>
<th>Compactness</th>
<th>Match Filtering</th>
<th></th>
<th>Prior-free</th>
<th>SR</th>
</tr>
</thead>
<tbody>
<tr>
<td>AS [31]</td>
<td>SIFT</td>
<td>×</td>
<td>Strict</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WPE [21]</td>
<td>SIFT</td>
<td>×</td>
<td>Relaxed</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>CSL [37]</td>
<td>SIFT</td>
<td>×</td>
<td>Relaxed</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>CPV [41]</td>
<td>SIFT</td>
<td>×</td>
<td>Relaxed</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Hyperpoints [29]</td>
<td>SIFT</td>
<td>✓</td>
<td>Relaxed</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>EGM [23]</td>
<td>SIFT+Binary</td>
<td>×</td>
<td>Relaxed</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>TC [6]</td>
<td>SIFT</td>
<td>×</td>
<td>Relaxed</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>SMC [38]</td>
<td>SIFT</td>
<td>×</td>
<td>Relaxed</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Our method</td>
<td>Binary</td>
<td>✓</td>
<td>Relaxed</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>

Before RPE

Before RPE

After RPE

After RPE

Relaxed

Strict
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 $\tau$. Therefore, for a match $m = \langle q \leftrightarrow p \rangle$, the set of 3D points that can form a match with query descriptor $q$ can be defined as $P(q) = \{ p \in P | h(s_q, s_p) \leq \tau \}$. Similarly, the set of query descriptors that can form a match with a 3D point $p$ can be represented as $Q(p) = \{ q \in Q | h(s_q, s_p) \leq \tau \}$. Our core idea is that a match should be distinct if its corresponding Hamming distance is significantly lower than the average Hamming distance in $P(q)$ and $Q(p)$. To evaluate a match within the feature space of a query image, we apply an image side Hamming ratio test as follows:

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

where one $|Q(p)|$ in $|Q(p)|^2$ 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 $\varphi$. We observe that setting $\varphi$ 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:

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

Since the term $|P(q)|$ 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:

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

### Aggregating Gaussian weighting function.

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

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

where \( h \) is the Hamming distance of a match, and \( \sigma \) is usually set to one quarter of the binary feature dimension [3].

By aggregating the Gaussian weighting function, the score for a match \( m \) can therefore be computed as follows:

\[
E(m) = T(m)w(h(m)).
\] (5)

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

### 2.2. Visibility-wisely Match Filtering

Given the match sets \( M \) and \( M_{FC} \), 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 \( M \) 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 \( M_{FC} = \{ m = \{ q \leftrightarrow p \} \mid m \in M_{FC}, (p, d) \in \mathcal{E} \} \) be the FC matches that vote for database image \( d \). We enforce that a match for database image \( d \) can be added to \( M_{FC}^d \) only if its corresponding query descriptor has not appeared in \( M_{FC}^d \) 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 \( \mathcal{P}^d = \{ p \mid (p, d) \in \mathcal{E} \} \) be the set of 3D points that are observed by the database image \( d \), the voting score can be defined as follows:

\[
S(d) = \frac{\sum_{m \in M_{FC}^d} E(m)}{\sqrt{|\mathcal{P}^d|}}.
\] (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 \( m \in M \), we select it into the set \( M_{d(k)} \) 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 \( M_{d(k)} \). Similarly, we apply a relaxed criterion by using a larger \( k \) to select another set of matches \( M_{d(k,1)} \) which may contain more correct matches but also are more noisy than \( M_{d(k)} \). \( M_{d(k,1)} \) 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 \( M_{d(k)} \) 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:

\[
M_{VFC}^{d(k)} = \{ m \mid m \in M_{d(k)} \land E(m) \geq \alpha \}.
\] (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 \( M_{VNFC}^{d(k)} = M_{d(k)} \setminus M_{VFC}^{d(k)} \). 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
Visibility-wise Match Filtering

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

Require: \( \mathcal{M}_{VFC} \leftarrow 0, \mathcal{M}_{VFC-I} \leftarrow \emptyset, \mathcal{M}^{d(k)}_{VFC-I} \leftarrow \emptyset \)

1: /* explore point image visibility */
2: Apply image voting with FC matches using Eq. 6
3: Retrieve top \( k \) and \( k_1 \) database images \( d(k) \) and \( d(k_1) \)
4: Select all matches in \( d(k_1) \) as \( \mathcal{M}^{d(k_1)}_{VFC} \) for visibility-wise match pool
5: Separate VFC matches \( \mathcal{M}_{VFC} \) and VNFC matches \( \mathcal{M}_{VNFC} \) using Eq. 7
6: /* explore point-image visibility */
7: for all \( d \in d(k) \) do
8: Compute the number of VFC matches \( \omega_{VFC}^{d(k)} \)
9: Compute the number of VNFC matches \( \omega_{VNFC}^{d(k)} \)
10: for all \( m \in \mathcal{M}_{VNFC} \) do
11: Compute the updated match score \( E'(m) \) using Eq. 8
12: for all \( m \in \mathcal{M}_{VNFC} \) do
13: if \( E'(m) \geq \alpha \) then
14: \( \mathcal{M}^{d(k)}_{VFC-I} \leftarrow \mathcal{M}^{d(k)}_{VFC-I} \cup \{ m \} \)
15: return \( \mathcal{M}^{d(k)}_{VFC} \cup \mathcal{M}^{d(k)}_{VFC-I} \) and \( \mathcal{M}^{d(k)}_{VNFC} \)

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 \( \omega_{VFC}^{d(k)} \) and \( \omega_{VNFC}^{d(k)} \) 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:

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

The larger the updated match score, the more likely that corresponding VNFC match is correct. Using the previously match score threshold \( \alpha \), 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:

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

Therefore, the matches that we select from \( \mathcal{M}^{d(k)}_{VFC-I} \) 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 \( \mathcal{M}^{d(k)}_{VFC} \).

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 \( B \). The VFC and VFC-I matches are then quantized into \( B \) 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 \( b \in B \). 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:

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

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 \( R_b \cdot N \).

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 \( \beta N_{VFC} \) VFC-I matches can be selected. In this work, we set \( \beta \) 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
its corresponding focal length as principal focal length selected pose hypothesis as an auxiliary camera pose, and dian value among the top pose hypotheses. We define the to select the pose hypothesis whose focal length is the me-

ants [10] that vote for optimal parameter values, we propose 
able camera pose estimation. Inspired from RANSAC vari-

largest number of inliers, provide us a more stable and reli-

values. These focal length values, instead of the one with 

of the top hypotheses have numerically close focal length 

proved localization accuracy. We apply the auxiliary 

recover potentially correct matches back can further im-

Filtering with auxiliary camera pose. The computed aux-

iliary camera pose exhibits sufficient accuracy. Using it to 

cover potentially correct matches back can further im-

prove the localization accuracy. We apply the auxiliary 

camera pose on the visibility-wise match pool $\mathcal{M}^{d(k_i)}$ to 

realize the geometry-wise filtering. We define a relaxed re-

projection error threshold $\theta$ in case rejecting potentially cor-

rect matches. As such, a match can be selected as a poten-
tially correct match if the re-projection error with respect to 
the auxiliary camera pose is below $\theta$. In this work, we 
choose a threshold of 10 pixels.

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 re-
lated 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 cen-
ter position and the ground truth camera center position of 
query image. The RobotCar Seasons [33] dataset was re-
constructed 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 im-
ages 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 Sea-
sons and Aachen Day-Night datasets, we follow the eval-
uation metric in [33] and report the percentage of query im-
ages localized within $U\text{m}$ and $V\text{m}$ from ground truth camera poses. To evaluate under different levels of localization ac-
curacy, we use the three accuracy intervals defined in [33] 
as follows: High-precision $(0.25\text{m}, 2\text{deg})$, Medium-precision 
$(0.5\text{m}, 5\text{deg})$ and Coarse-precision $(5\text{m}, 10\text{deg})$. For the large-
scale SF-0 dataset [21], we use the evaluation package pro-
vided by [34] which contains reference camera poses for 442 query images.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Database Images</th>
<th>3D Points</th>
<th>Query Images</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dubrovnik [22]</td>
<td>6,044</td>
<td>1.89M</td>
<td>800</td>
</tr>
<tr>
<td>RobotCar Seasons [33]</td>
<td>20,862</td>
<td>6.77M</td>
<td>11,934</td>
</tr>
<tr>
<td>Aachen Day-Night [33]</td>
<td>4,328</td>
<td>1.65M</td>
<td>922</td>
</tr>
<tr>
<td>SF-0 [21, 34]</td>
<td>610,773</td>
<td>30M</td>
<td>442</td>
</tr>
</tbody>
</table>
Table 3: The comparison between our method and state-of-the-art methods on the Dubrovnik dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>Error Quartiles [m]</th>
<th>Localized images</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>25%</td>
<td>50%</td>
</tr>
<tr>
<td>EGM</td>
<td>0.24</td>
<td>0.70</td>
</tr>
<tr>
<td>TC</td>
<td>0.22</td>
<td>1.07</td>
</tr>
<tr>
<td>AS</td>
<td>0.40</td>
<td>1.40</td>
</tr>
<tr>
<td>Our method</td>
<td>0.22</td>
<td>0.64</td>
</tr>
</tbody>
</table>

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 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, $\tau$ to 32 and $\alpha$ 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. red and blue represent the best and second-best methods, and the asterisk symbol represents using knowledge about the gravity direction.

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.

<table>
<thead>
<tr>
<th></th>
<th>RobotCar Seasons</th>
<th>Aachen Day-Night</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>All Day</td>
<td>All Night</td>
</tr>
<tr>
<td></td>
<td>.25 / 0.5 / 5.0</td>
<td>.25 / 0.5 / 5.0</td>
</tr>
<tr>
<td></td>
<td>2 / 5 / 10</td>
<td>2 / 5 / 10</td>
</tr>
<tr>
<td>AS</td>
<td>35.6 / 67.9 / 90.4</td>
<td>0.9 / 2.1 / 4.3</td>
</tr>
<tr>
<td>DenseVLAD</td>
<td>7.7 / 31.3 / 91.2</td>
<td>1.0 / 4.5 / 22.7</td>
</tr>
<tr>
<td>NetVLAD</td>
<td>6.4 / 26.3 / 91.0</td>
<td>0.4 / 2.3 / 16.0</td>
</tr>
<tr>
<td>CSL*</td>
<td>45.3 / 73.5 / 90.1</td>
<td>0.6 / 2.6 / 7.2</td>
</tr>
<tr>
<td>SMC*</td>
<td>50.6 / 79.8 / 95.1</td>
<td>7.6 / 21.5 / 45.4</td>
</tr>
<tr>
<td>Our method</td>
<td>48.0 / 78.0 / 94.2</td>
<td>3.4 / 9.5 / 17.0</td>
</tr>
<tr>
<td></td>
<td>76.7 / 88.6 / 95.8</td>
<td>25.5 / 38.8 / 54.1</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Method</th>
<th>Memory Consumption</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Dubrovnik</td>
</tr>
<tr>
<td>AS</td>
<td>0.75</td>
</tr>
<tr>
<td>Our method</td>
<td>0.14</td>
</tr>
</tbody>
</table>

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. 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


[33] 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


[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