Pose Correction for Highly Accurate Visual Localization in Large-scale Indoor Spaces

Janghun Hyeon[1*] Joohyung Kim[1*] Nakju Doh[1,2]
[1] Korea University, [2] TeeLabs
Seoul, Republic of Korea
janghun0414@gmail.com kjh069@gmail.com nakju@korea.ac.kr

Abstract

Indoor visual localization is significant for various applications such as autonomous robots, augmented reality, and mixed reality. Recent advances in visual localization have demonstrated their feasibility in large-scale indoor spaces through coarse-to-fine methods that typically employ three steps: image retrieval, pose estimation, and pose selection. However, further research is needed to improve the accuracy of large-scale indoor visual localization. We demonstrate that the limitations in the previous methods can be attributed to the sparsity of image positions in the database, which causes view-differences between a query and a retrieved image from the database. In this paper, to address this problem, we propose a novel module, named pose correction, that enables re-estimation of the pose with local feature matching in a similar view by reorganizing the local features. This module enhances the accuracy of the initially estimated pose and assigns more reliable ranks. Furthermore, the proposed method achieves a new state-of-the-art performance with an accuracy of more than 90 % within 1.0 m in the challenging indoor benchmark dataset InLoc for the first time. [1]

1. Introduction

Indoor visual localization is a common solution for indoor applications such as autonomous robots, augmented reality, and mixed reality [8, 19, 32, 35]. However, even though recent advances in visual localization have demonstrated remarkable performances in urban environments and small indoor spaces [4, 5, 6, 7, 22, 23, 28], long-term visual localization in large-scale indoor spaces remains challenging due to similar places, repetitive patterns, featureless scenes, occluded scenes, and highly dynamic features [54].

Recently, it was reported that visual localization can be successfully scaled-up in indoor spaces using InLoc [54] and HFNet [42]. These works employ a hierarchical (coarse-to-fine) structure in which the algorithm retrieves several candidates using the lightest feature and subsequently estimates the poses of the selected few with more intensive features. The black boxes in Figure 1 describe the hierarchical model constituted by

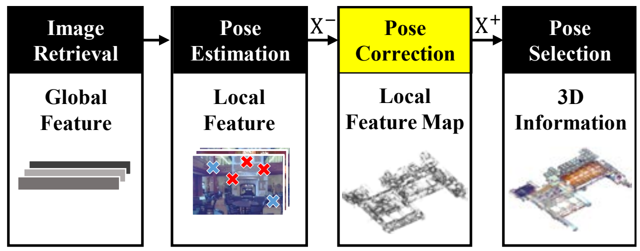


Figure 1. The hierarchical model, including pose correction. The pose correction step updates the initially estimated pose X^- to X^+. The main information used in each step is shown below the box.

- *Image retrieval*: retrieve many candidates with indirect features such as NetVLAD [1], GeM [37], AP-GeM [38], and i-GeM [19].
- *Pose estimation*: estimate candidates' pose with direct features such as SuperPoint [10] and D2Net [13].
- *Pose selection*: select the final pose with given 3D information such as pose verification (PV) [54, 55] and covisibility clustering [42].

These frameworks are de facto standards because many successful studies have inherited these structures [13, 14, 17, 19, 40, 41, 43, 50, 51, 55]. However, we argue that there is further scope for improvement because the accuracy of recent state-of-the-art methods [14, 17, 43] is approximately 80 % within 1.0 m in large-scale indoor spaces [54], where it often reaches over 90 % in outdoor benchmark datasets [3, 45, 47]. We determine that the sparsity of image positions in the database is the reason for the performance gap

*Equally contributed to this work.
[1]Code available at http://github.com/JanghunHyeon/PCLoc

because it is difficult to construct the database densely in large-scale indoor spaces [54]. The sparsity causes view-difference between a query image and a retrieved image. For example, when a query pose is far from the database image pose, the common view in both images tends to be small. Thus, the pose estimation module in existing methods yields inaccurate output as the local features that appear in both the query and the database image are not sufficient for accurate estimation.

In this work, to circumvent the sparsity issue and improve the accuracy, we propose a novel module called *"Pose Correction,"* as shown in the yellow box in Figure 1, which reorganizes local features that can be observed from the estimated pose (X^-). Note that this approach has an effect similar to that of estimating the query pose using an image located near the query pose. Figure 2(a) depicts an example where the query and database poses are far from each other. Owing to the view-difference, only a few features match between the query and database images, as shown in Figure 2(b). However, if we reconstruct the local features that can be observed in X^- and associate the two sets of features, more inliers appear, which circumvent the sparsity problem and resolve the view-difference problem as in Figure 2(c). This yields an updated pose (X^+) whose accuracy is superior to X^-. From the given candidates, once all X^+ candidates have been re-estimated, it is natural to reset their ranks in the order of the reliability of the matching. We evaluate the reliability using the number of inliers between the query and X^+ and provide more reliable candidates to the pose selection module.

In addition, we propose an extended pose correction that utilizes the properties of the pose correction step and also reduces redundant features that might be used during the pose update. We also modify the PV proposed in [54] such that the accuracy can be enhanced as far as possible.

Experiments were conducted on the most well-known indoor benchmark dataset, InLoc [54]. We validated our proposed method by comparing it with existing state-of-the-art methods [13, 14, 17, 40, 42, 43, 54, 55]. Further, we evaluated our method on an M-site dataset [19] to confirm the relevance of our results. Our proposed method performed significantly better and achieved state-of-the-art results for large-scale indoor visual localization. Moreover, we also conducted ablation studies to demonstrate the superiority of the extended pose correction and the effect of the iterating pose correction.

The contributions of this work are as follows. 1) To the best of our knowledge, it is the first work to address the problem of the view-difference due to the sparsity in the database and to propose a novel module, i.e., pose correction, to resolve the problem. 2) We extend pose correction based on its natural properties and verify improvements in accuracy. 3) Additionally, we propose modified PV (MPV),

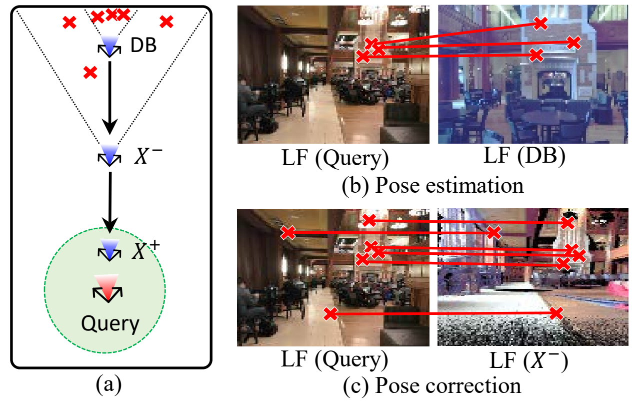


Figure 2. (a) Query and database poses are far from each other. The visible local features in the query image are represented with a red x. (b) Owing to the sparsity that causes view-difference, there are few feature matches between the query and the database images. (c) Local feature matching with the query and the reorganized features that can be observed in X^- yields a greater number of matches than the ones in (b) circumventing the sparsity problem. Here, the background image at X^- is rendered for visualization. LF denotes local feature.

which improves the performance further. 4) As a result, the proposed method outperforms recent works by a notable margin and achieves a new state-of-the-art in the public benchmark dataset.

2. Related Work

Many existing methods such as absolute or relative pose regression-based methods [7, 11, 22, 23, 28] and structure-based regression methods [4, 5, 6] have failed to estimate accurate poses in large-scale spaces [54].

Different approaches such as map-less approaches and structure-based approaches that use pre-defined 3D maps also have been studied for visual localization. Sattler *et al*. [46] proposed a map-less localization. The map-less method may reduce database size with the cost of run-time efficiency. In order to recover a camera pose, this method requires a large number of retrieved images for Structure-from-Motion (SfM) on the fly. However, SfM may fail in datasets such as InLoc dataset due to many reasons, including the small overlap between images [17].

Recent visual localization methods based on the coarse-to-fine model show feasibility in large-scale indoor spaces [17, 19, 42, 54]. These methods perform image retrieval [1, 37, 38] to predict the coarse position and restrict the search space for 2D-3D matching. Local feature matching [10, 13, 39] is then performed for each retrieved image (candidates) with the query image. These retrieved images are correlated 3D models represented by Structure-from-Motion point cloud [20, 29, 30, 47], LiDAR scans [34, 54] or mesh surfaces [12, 18, 19]. Thus, local feature matching

(2D-2D matching) enables 2D-3D matching through correlated 3D coordinates. The matched correspondences are then used to estimate the camera pose using the Perspective-n-Point (PnP) method [24, 25, 26] within a RANSAC loop [9, 16, 27]. Subsequently, the best pose is selected as the final pose.

Based on coarse-to-fine models, many techniques have been proposed to enhance localization performance. Some studies are focusing on retrieving better candidates [14, 17, 19] by adopting robust global descriptors [17, 19, 36, 38]. Additionally, several studies have attempted to increase the accuracy by extracting more robust local features [10, 13, 39] or feature matching [40, 41, 43]. Further, some studies use additional information such as semantic or depth information to select more reliable matched inliers [14, 50, 51] to conduct accurate pose estimation through accurate local feature matching of query and database images. Some studies are focusing on selecting the best candidates [15, 54, 55].

In short, recent studies aim to improve modules of the coarse-to-fine framework such as image retrieval [14, 17, 19], pose estimation [10, 13, 14, 39, 40, 41, 43, 50, 51], and pose selection [15, 54, 55]. In contrast, to the best of our knowledge, our study is the first work that proposes the pose correction module in the coarse-to-fine framework to address the limitation of the existing framework due to the view-difference problem in the large-scale indoor visual localization.

3. Visual Localization with Pose Correction

3.1. Baseline

InLoc [54] is a representative coarse-to-fine approach that uses three steps: image retrieval, pose estimation, and PV. We set the method as our baseline and build our pipeline upon it. First, we retrieve the top-K_1 closest images to a given query image from the database using NetVLAD [1], which converts an image into a global feature. Using NetVLAD, we predefine the global features of database images efficiently and use the nearest neighbor method to retrieve the K_1 best matching images.

The K_1 images are used for the next step, which is pose estimation. In this step, we extract local features (*i.e.* SuperPoint [10]) from the query image and a candidate image. Those features are matched using a robust feature matching algorithm based on a graph neural network, which is named SuperGlue [43]. With the given 3D information from the database and the correspondences, the query pose is estimated using a 2D-to-3D PnP algorithm [24] in a RANSAC loop [16]. Subsequently, we sort the final top-K_2 candidates out of K_1 candidates in the order of the number of RANSAC inliers. The main difference between InLoc [54] and our baseline is that InLoc uses dense features from certain layers of a convolutional neural network for matching, whereas we use sparse SuperPoint [10] features and the SuperGlue [43] matcher.

	# of Camera Location	# of DB Images	Area
7-Scenes [52]	26,000	26,000	31.5 m^3
12-Scenes [56]	240,002	240,002	521 m^3
M-Site [19]	720	25,920	12,557 m^2
InLoc [54]	277	9,972	25,287 m^2

Table 1. Sparsity difference between small-scale and large-scale indoor datasets.

Finally, PV selects the best pose among the K_2 candidate poses. A synthesized view is rendered from the RGBD data scanned at the position of the retrieved image. Subsequently, the similarity between the synthesized image and query image is evaluated by comparing pixel-wise local patch descriptors, DenseRootSIFT [2, 33]. The similarity score is defined as the median value of the pixel-wise distances between the descriptors disregarding the missing pixels in the synthesized image.

3.2. Key limitations in the baseline

In the large-scale indoor spaces, previous coarse-to-fine methods [19, 42, 54, 55], including our baseline, exhibit limitations due to the characteristics of the sparsity in the image database. For example, while the spaces of small-scale indoor datasets (*e.g.* [52, 56]) are typically reconstructed by densely captured RGB-D data, those of large-scale indoor datasets (*e.g.* [19, 54]) are reconstructed by data scanned from sparsely located positions (*c.f.* Table 1). The sparsity causes problems with respect to *view-difference* and *selection of reliable candidates*.

As mentioned in InLoc [54] in detail, there is no practical approach that constructs a densely captured image database in a way that reduces the data acquisition time and manual work. Therefore, the distance between two consecutive database images is very large, considering the accuracy level of the visual localization. For example, in the InLoc dataset [54], the scans at 277 distinct positions cover 25,287 m^2 indoor spaces, while its performance metric is set to 0.25 m. This sparsity induces a significant view-difference between a query and a retrieved image, as shown in Figure 2(b), which yields a poor performance in the pose estimation.

The sparsity also makes it hard to select reliable candidates. When the overlap between a query and a retrieved database image is small, the number of inliers in the local feature matches for the true positive candidates may be less than that for the false positive candidates. Subsequently, the true positive candidates may not be selected among the top-K_2 candidates in the pose estimation step.

3.3. Pose correction

To circumvent the two limitations, we propose a complementary step named pose correction in between the pose estimation and the PV, as shown in Figure 1. This step consists of two building blocks. One is a pose-update that converts X^- from the pose estimation into X^+. The other is a reranking that selects more reliable candidates.

While constructing the database, we group the local features of the images from a scanned position p_i to create a local feature map, $\mathbb{F}_i = \{^{p_i}\mathbf{F}_j | j = 1, 2, ..., n\}$, where $\mathbf{F}_j$ contains the scanned position p_i, the local features (*i.e.* SuperPoint [10]) in an image $\mathbf{I}_j$ and their corresponding 3D points in the global coordinate system, and n is the number of images covering the scan-view as shown in Figure 3(a).

In the pose correction step, each candidate has information regarding the index i of the scanned position p_i. The local features in $\mathbb{F}_i$ are projected onto the image plane of X^-, making it a *synthetic local feature image*, $\mathbf{I}'$, as shown in Figure 3(b). The projected local features are used for feature matching with those of the query image using SuperGlue. With the 2D-to-3D correspondences, the PnP algorithm in the RANSAC loop follows to update the pose to X^+. The inliers from the 2D-to-3D correspondences are used in reranking which reorders the K_2 candidate set conveyed from the pose estimation into a new K_3 set.

The pose correction step has two properties that are superior to the pose estimation step: proximity and abundance of features. It resolves the view-difference problem of the pose estimation step using features that are visible from X^-, which is a pose that shares a similar view to the query's view. As a result, true positive features for feature matching can be located near the query's local features in the image coordinate system. In addition, pose correction extends the local features extracted from a database image to the local features that are extracted from the multiple images, resulting in an abundance of features. Consequently, they contribute to the improvement in localization accuracy.

3.4. Extended pose correction

In this section, we propose an extended pose correction that utilizes the properties of the pose correction step and reduces redundant features to further improve localization accuracy.

Divided matching Employing the property of proximity of pose correction, we propose divided matching, which segments an image into sub-regions such as the top, bottom, left, and right halves of an image to find feature matchings in each area. It helps in finding inliers that are spatially distributed in larger areas of the image without fine-tuning the pre-trained SuperGlue model [43]. As the spatial distribution of the inliers is vital for an accurate pose estimation [15, 48, 58], divided matching leads to performance improvement of pose correction.

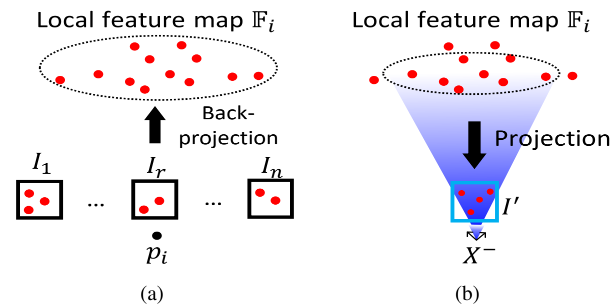


Figure 3. (a) While constructing the database, local features $\mathbf{F}_j$ that are extracted from database images captured at p_i are back-projected to the 3D space to create the local feature map $\mathbb{F}_i$. (b) In the pose correction step, visible local features are projected onto the X^- image plane to create a synthetic local feature image $\mathbf{I}'$.

Divided matching is useful when the views between two images are sufficiently similar. Therefore, it can be used when the database poses are ideally dense such that there always exists a database image that is similar to an arbitrary query's view, or for the pose correction that updates X^- from a view similar to that of the query.

Inter-pose matching Extending the property of abundance of pose correction, we propose inter-pose matching, which utilizes multiple $\mathbb{F}_i$ to find even more feature matchings in the pose correction step. For this, we utilize *Scangraph* [55] that contains connectivity information, of which a node is a scanned position p_i, and an edge is the connectivity information indicating that adjacent nodes share adequate view. This enables to consider co-visibility [29, 30, 42, 44] when the database is not constructed with structure-from-motion techniques. Inter-pose matching is applied in the pose correction step to use one or more $\mathbb{F}_i$ according to the connectivity information to create one or more synthetic local feature images. The found matches are concatenated for use in the PnP algorithm inside the RANSAC loop.

In indoor spaces, the distance to the scene geometry tends to be short, and concave structures or clutters often cause significant occlusions. In these cases, the inter-pose matching helps in finding correct local features that are captured from different scanned positions.

Filtering process As the projection of the local feature map in the pose correction step does not consider occlusions, reducing redundant local features projected onto the synthetic local feature image $\mathbf{I}'$ is beneficial for better feature matching. For this, we employ two approaches: preprocessing with virtual local feature (VLF) map and point normal filtering on the fly.

Similar to [20], which is conducted in the context of image retrieval via bag-of-words models, the VLF map adds virtual positions to the database and finds features that are visible from the virtual positions. Specifically, a VLF map ($\mathbb{F}'$) extends $\mathbb{F}$ by adding virtual positions, p'_l, to the database and by removing local features that are invisible from p'_l ahead of inference time (*i.e*. database construction time). The VLF map increases the density of the database and reduces the chances of invisible local features being projected on $\mathbf{I}'$ during the inference time.

A virtual position, p'_l, is set for each edge in the Scan-graph under the following conditions: p'_l should be located inside the map, and the local features extracted from the two adjacent positions observed from p'_l should be as many and as even as possible. To detect visible local features at p'_l, we employed the hidden point removal algorithm [21], which is a robust and efficient algorithm to remove occluded points and select only the visible points in the point cloud map. The newly extended feature $\mathbb{F}'_l$ at p'_l is defined as $\mathbb{F}'_l = \{f | f \in \mathbb{F}$, and visible at $p'_l\}$, where f is a local feature and its associated 3D point.

In the pose correction step, $\mathbb{F}_i$ and $\mathbb{F}'_l$ are used in a similar way to inter-pose matching, where p'_l that is closest to X^- is selected.

Meanwhile, the point normal filtering removes the invisible features in the inference time based on the cosine distance between a point normal of the local feature and direction vector from X^- to the point. For this, we add point normal information in $\mathbb{F}_i$ to create $\hat{\mathbb{F}}_i$ based on the surface normal of local features in the database images when constructing the database.

These two filtering methods are optional, but we found them to be effective when used along with other proposed matching methods. More details are provided in the supplementary material.

3.5. Modified pose verification

PV is the final step that determines the most appropriate pose among candidates, and thus has a direct effect on the overall pipeline performance.

To improve overall performance and leverage the effect of our proposed pose correction module, we propose MPV. It is a simple and effective modification of PV, which removes outlier pixels in the rendered image that are not appropriate to compare with the query image. Figure 4 illustrates an example wherein MPV successfully finds a correct pose by removing outliers in the rendered image.

First, we remove lower outlier pixels in score distribution using opening [49], which is a simple morphological image processing that removes isolated small pixels in an image. Owing to the implementation of the DenseRootSIFT, the pixel with many invalid pixels in the neighbor shows significantly low value in the Euclidean distance of descriptors (*e.g*. Figure 4(b)). We remove such pixels and preserve the valid area using opening in error maps (*e.g*. Figure 4(c)). For the opening process, the pixels are binarized according to whether they are valid or missing.

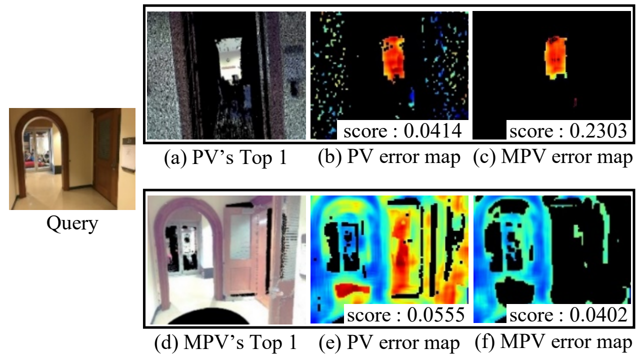


Figure 4. From the same candidate poses given, PV and MPV select different final poses, (a) and (d), respectively. The error maps with final scores are shown in (b), (c), (e), and (f), where a lower score means better candidate. The blue pixels represent lower values in Euclidean distance between descriptors, whereas red pixels represent higher values. MPV removes sparse pixels, as it assumes them to be lower outliers as in (c). In addition, MPV neglects the upper outlier pixels caused by changes in illumination or open doors as in (f). Invalid or removed pixels are colored in black.

Second, we remove upper outlier pixels by modifying the method of evaluating similarity from the median value to the average value below the median. The value represents an overall score of similar area between the query and the rendered image and reduces the effect of changes in the scene due to dynamic features and illumination changes by ignoring such pixels (*e.g*. Figure 4(f)).

4. Experimental Setup

4.1. Evaluation dataset

The best-known indoor visual localization benchmark datasets are 7-scenes [52], 12-scenes [56], and InLoc [54]. Many regression-based methods [22, 23, 28] and 3D scene coordinate regression-based methods [4, 5] employ the 7-scenes and 12-scene datasets. However, these datasets consist of non-dynamic small spaces that are not appropriate for our study. Hence, we evaluated our method using the InLoc and M-site [19] datasets.

The InLoc dataset provides 10k images and corresponding depth data using a camera mounted on a laser scanner. It covers very large indoor spaces ($25,287\,m^2$), which comprise multiple floors in multiple university buildings with different properties [57]. In addition, it contains large textureless places, many repetitive areas, illumination changes, highly occluded places, and numerous dynamic features, which make localization difficult. The 329 query images were recorded by an iPhone7 approximately a year after the database was generated, allowing evaluation of long-term localization. In addition, the query images are distributed across two places (DUC1 and DUC2) and captured from significantly distant positions from the database scans.

Error [m, 10°]	DUC1 0.25	DUC1 0.5	DUC1 1.0	DUC2 0.25	DUC2 0.5	DUC2 1.0
InLoc [54]	40.9	58.1	70.2	35.9	54.2	69.5
HfNet [42]	39.9	55.6	67.2	37.4	57.3	70.2
KAPTURE [17]	41.4	60.1	73.7	47.3	67.2	73.3
D2Net [13]	43.9	61.6	73.7	42.0	60.3	74.8
Oracle [55]	43.9	66.2	78.3	43.5	63.4	76.3
Sparse NCNet [40]	47.0	67.2	79.8	43.5	64.9	80.2
RLOCS [14]	47.0	71.2	84.8	58.8	77.9	80.9
SuperGlue [43]	46.5	65.7	77.8	51.9	72.5	79.4
Baseline (3,000)	53.0	76.8	85.9	61.8	80.9	87.0
Ours (3,000)	59.6	78.3	89.4	**71.0**	**93.1**	**93.9**
Ours (4,096)	**60.6**	**79.8**	**90.4**	70.2	92.4	93.1

Table 2. Evaluation results for the InLoc dataset.

Error [m]	M-site 0.25	M-site 0.5	M-site 1.0	M-site 3.0	M-site 5.0
InLoc [54]	40.7	56.8	68.6	75.6	76.1
KR-Net [19]	47.0	58.9	66.1	72.3	73.1
Baseline	46.0	65.9	75.0	79.0	79.7
Ours	**50.6**	**68.9**	**76.3**	**80.1**	**81.1**

Table 3. Evaluation results for the M-site dataset.

Error [m, 10°]	DUC1 0.25	DUC1 0.5	DUC1 1.0	DUC2 0.25	DUC2 0.5	DUC2 1.0
(a) Baseline+PV	53.0	76.8	85.9	61.8	80.9	87.0
(b) Baseline+MPV	56.1	76.8	88.4	65.6	82.4	85.5
(c) Proposed+PV	56.1	76.3	86.4	63.4	84.7	90.8
(d) Proposed+MPV	59.1	77.8	89.9	68.7	92.4	93.9
(e) Baseline+SGPV	56.1	73.7	83.8	58.0	77.1	83.2
(f) Baseline+SGMPV	57.1	74.7	87.4	63.4	79.4	84.0
(g) Proposed+SGPV	59.1	77.8	89.9	68.7	92.4	93.9
(h) Proposed+SGMPV	59.6	78.3	89.4	71.0	93.1	93.9

Table 4. Evaluations of modified pose verification. The best accuracy in each column is in red and the second best in blue. In (e-h), SG represents the usage of Scangraph [55], applied in the pose verification step.

The M-site database provides 25k images and the corresponding depth data using a robot system (LiDARs and 360° camera). It covers a large-scale indoor space ($12,557\,m^2$). Most places in the M-site are featureless and similar spaces, which makes feature matching difficult. The 472 query images were recorded using an RGB-D camera (RealSense) on different dates and times.

Overall, InLoc and M-site are the most appropriate datasets for evaluating pose correction and large-scale indoor visual localization. Although the ground truth of the InLoc dataset is not publicly available, we choose the dataset to evaluate our pipeline as it is the most suitable and widely used benchmark.

4.2. Implementation details

We used NetVLAD pre-trained on the Pitts30K [1] dataset with the VGG-16 [53] model for image retrieval. For local feature extraction, we used Superpoint [10] with 3,000 local features in the InLoc, and 4,096 in the M-site dataset. We used SuperGlue [43] pre-trained on the MegaDepth dataset [31] for local feature matching. The query image used as input was resized to the longest length of 1200 pixels.

We retrieved 100 candidate images ($K_1 = 100$) and used 10 candidates for PV ($K_3 = 10$), the same as in InLoc [54]. In the pose correction step, we used 20 candidate poses in our experiments ($K_2 = 20$).

5. Experimental Evaluation

5.1. Comparison with the state-of-the-art methods

To evaluate the proposed method, we compare it with the state-of-the-art methods on the InLoc and M-site datasets. The results for the InLoc and M-site are presented in Tables 2 and 3, respectively.

For the InLoc dataset, we compared our results to the recent state-of-the-art methods. As listed in Table 2, our proposed method outperforms every existing state-of-the-art method by a large margin. In addition, we evaluated the proposed method using 3,000 and 4,096 SuperPoint [10] local features to verify that the number of local features used does not affect the performance. Every evaluation was conducted with the online visual localization benchmark server[2].

Further, we evaluated pose correction on the M-site dataset to confirm the relevance of our results. We compared our proposed method to the InLoc and KR-Net [19]. The results reveal that the proposed method shows better performance within every threshold, as summarized in Table 3. In addition, we compared our method to our baseline that does not use pose correction. The result shows that using pose correction improves accuracy, especially within $0.5\,m$, compared to the baseline. This indicates that pose correction updates X^- more accurately as intended.

Overall, our proposed method achieves a new state-of-the-art performance in both the InLoc and M-site datasets.

5.2. Evaluation of each component

Modified pose verification Before evaluating the components in the pose correction, we first evaluate the MPV. With a better pose selection module (*i.e.* MPV), it is easier to find better components in an earlier stage of entire pipeline when they change.

Table 4 presents the comparisons between PV and MPV. As can be seen in (a) and (b) in Table 4, MPV outperforms PV on almost every error criterion when the baseline method is used without using Scangraph in PV. If we

[2] https://www.visuallocalization.net

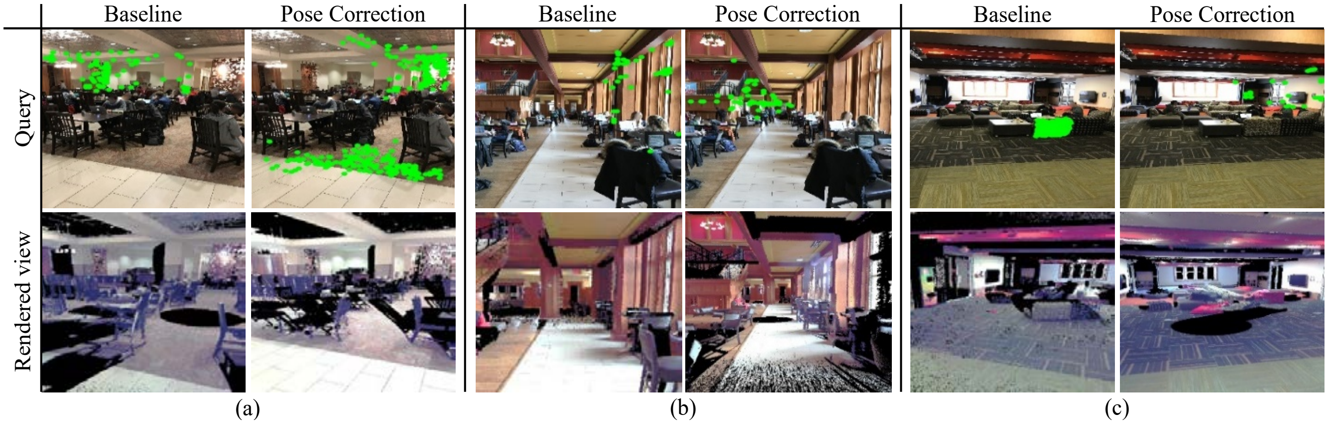


Figure 5. Qualitative comparisons between our baseline and the pose correction. Green dots are inlier features used for estimating X^- and X^+. (a) Local features in the baseline image are clustered in a smaller area in the query image than those in the pose correction. (b) A noticeable transitional error appears in the rendered view of the baseline because of the repetitive patterns in the indoor structures. (c) A noticeable rotational error appears in the baseline owing to the moved furniture. (a–c) Thus, pose correction circumvents the issues that the baseline confronts frequently by reorganizing local features and enhancing the localization accuracy.

change the baseline method to the proposed method, MPV outperforms PV on all criteria (*c.f.* (c) and (d) in Table 4). Here, the proposed method refers to the use of methods like divided matching, inter-pose matching, and filtering processes. Similarly, when Scangraph is used in the baseline (*c.f.* (e) and (f) in Table 4) and in the proposed method (*c.f.* (g) and (h) in Table 4), MPV still outperforms PV on almost every metric.

In short, MPV selects better poses than PV when the same top-K_3 candidates are provided from the same front pipeline with or without Scangraph. Therefore, we use MPV instead of PV for all of the following experiments.

Pose correction To verify the effect of using the pose-update, we use 10 candidates from the pose estimation step. Row (b) in Table 5 updates their poses, whereas (a) does not. Experiments in (a) and (b) in Table 5 show that pose-update improves the localization accuracy, as intended.

Next, we verify the effect of using reranking by comparing the results between (b) and (c) in Table 5. The results show that using reranking in (c) enables the selection of more reliable candidates to be used in the PV than without using reranking in (b).

The results of (a) and (c) in Table 5 indicate that even basic pose correction improves the localization performance. Qualitative comparisons between the two are shown in Figure 5.

Extended pose correction The following experiments focused on extended pose correction: divided matching, inter-pose matching, and filtering process. For some experiments, Scangraph [54] in the PV is applied to assist each method.

First, divided matching is compared with the ones that do not use divided matching. For fair comparison, we choose three pairs for comparisons in Table 6, including (a-1, a-2) the basic pose correction, (b-1, b-2) pose correction using inter-pose matching, and (c-1, c-2) pose correction using a VLF map. The results indicate overall improvements in the performance for all criteria except for one for each pair, thereby indicating that divided matching is promising or even better than the original matching for pose correction.

	DUC1			DUC2		
Error [m, 10°]	0.25	0.5	1.0	0.25	0.5	1.0
(a) Baseline (10)	56.1	**76.8**	88.4	65.6	82.4	85.5
(b) PC (10, 10)	58.1	**76.8**	**89.4**	67.2	**90.1**	**92.4**
(c) PC (20, 10)	**58.6**	**76.8**	**89.4**	**67.9**	**90.1**	**92.4**

Table 5. Evaluation of the pose correction. (a) Baseline introduced in Section 3.1 using $K_2 = 10$. (b) Pose correction introduced in Section 3.3 using both $K_2 = 10$ and $K_3 = 10$. It updates the poses while excluding the effect of using reranking in pose correction. (c) Pose correction using $K_2 = 20$ and $K_3 = 10$.

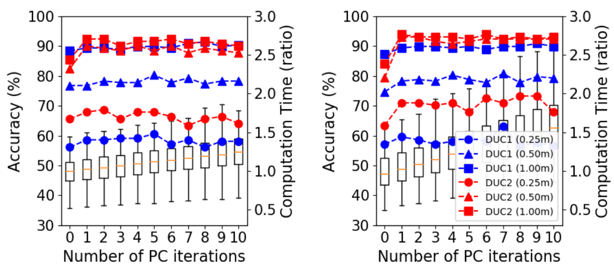
Second, to determine the effect of inter-pose matching, result pairs (a) and (b) in Table 6 are compared. While other performances do not seem to change considerably, a performance gain is achieved in DUC2 at the fine estimation, *i.e.* at 0.25 m, by up to 3.8 %p in the comparison between (a-3, b-3). We believe that the additional matches obtained from sub-scans make the pose refinement more precise.

The best performances can be obtained when filtering processes are used, *i.e.* the VLF map and point normal filtering, as shown with (c-4) and (c-5) in Table 6. In addition, experiments (c-1) and (c-2) achieve an accuracy above 90 % within 1.0 m in both spaces, DUC1 and DUC2. The results indicate that performance improvements can be achieved using the VLF map.

Intriguingly, although adding each component step-by-

	idx	Error [m, 10°]	DUC1			DUC2		
			0.25	0.5	1.0	0.25	0.5	1.0
w/o inter-pose	(a-1)	F	58.6	76.8	89.4	67.9	90.1	92.4
	(a-2)	Div	60.1	75.8	89.4	69.5	91.6	92.4
	(a-3)	Div-N	59.6	80.8	89.4	67.2	90.8	91.6
	(a-4)	Div-SG	59.6	77.8	88.9	66.4	90.8	91.6
	(a-5)	Div-N-SG	59.6	80.8	89.4	67.2	90.8	91.6
w/ inter-pose	(b-1)	F	57.1	79.8	88.9	66.4	87.8	91.6
	(b-2)	Div	59.6	80.3	89.9	71.0	90.1	90.8
	(b-3)	Div-N	59.1	79.3	89.9	71.0	91.6	91.6
	(b-4)	Div-SG	59.6	79.8	88.9	69.5	90.1	90.1
	(b-5)	Div-N-SG	60.6	79.3	89.4	70.2	90.1	90.1
w/ VLF map	(c-1)	F	58.1	78.3	90.4	69.5	89.3	92.4
	(c-2)	Div	60.1	79.3	90.9	68.7	91.6	92.4
	(c-3)	Div-N	59.1	77.8	89.9	68.7	92.4	93.9
	(c-4)	Div-SG	60.6	77.8	89.9	70.2	92.4	93.9
	(c-5)	Div-N-SG	59.6	78.3	89.4	71.0	93.1	93.9

Table 6. Ablation studies for each module used in extended pose correction. Experiments are conducted (a) without using inter-pose matching, (b) using inter-pose matching, and (c) using the VLF map. Character F denotes full matching, which is the original matching method using SuperGlue, whereas Div represents the divided matching. N represents the usage of point normal filtering. The best accuracy in each column is in red and the second best in blue.



(a) (b)

Figure 6. (a) and (b) depict the results of iterating basic pose correction and extended pose correction, respectively. The accuracy results are depicted with dotted lines (left y-axis). The computational times are depicted with box plots (right y-axis). 0-iteration denotes our baseline method, and the computational time of each iteration is expressed proportionally to it.

step did not consistently lead to performance gain, the best performance was achieved when most proposed methods were used, such as the divided matching, point normal filtering, and VLF map (*i.e*. (c-2, c-4, or c-5)). We believe that the VLF map is beneficial because it uses the local features from the other scan positions, and the invisible local features are filtered out at the time of database construction.

5.3. Iteration of pose correction

Although we used a single iteration of pose correction, but the iterations can be more. We further evaluate the trade-off between the run-time efficiency and the performance gain for more iterations of pose correction.

The results in Figure 6 show that more iterations slow down the run-time speed. However, the performance gains over the iterations are not very noteworthy. This might occur because the performance of iterations relies on SuperGlue [43] in our settings. In practice, the initial pose correction already yields accurate poses and SuperGlue does not yield strictly better matches as the pose correction is iterated. As a result, iterations do not guarantee a better result than the initially corrected pose (*i.e*. the first iteration in Figure 6(a) and (b)), which is the approach that we propose.

6. Conclusion

We present a method for pose correction that exhibits robust and accurate localization when the sparsity of image positions inheres in the database, which has been the main limitation of previous coarse-to-fine methods for large-scale indoor localization. Pose correction reorganizes local features visible from the estimated pose, and the properties of pose correction are further extended by introducing divided matching, inter-pose matching, and filtering process. We demonstrate the superiority of pose correction and each component in extended pose correction through ablation studies. According to the experimental results, the first iteration of pose correction can improve performance, but subsequent iterations do not exhibit significant improvements. As a result, the proposed method sets a new state of the art in public benchmark datasets, InLoc, with an accuracy of more than 90 % within 1.0 m for the first time.

Pose correction can be beneficial for large-scale indoor visual localization where the database images need to be captured sparsely. This means that using the pose correction module may allow visual localization applications to reduce database size and enhance database efficiency.

Acknowledgement. This research was supported by the Technology Innovation Program (10073166) funded By the Ministry of Trade, Industry and Energy (MOTIE, Korea).

References

[1] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *CVPR*, 2016. 1, 2, 3, 6

[2] Relja Arandjelović and Andrew Zisserman. Three things everyone should know to improve object retrieval. In *CVPR*, 2012. 3

[3] Aayush Bansal, Hernán Badino, and Daniel Huber. Understanding how camera configuration and environmental conditions affect appearance-based localization. In *IV*, 2014. 1

[4] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. DSAC-Differentiable RANSAC for camera localization. In *CVPR*, 2017. 1, 2, 5

[5] Eric Brachmann and Carsten Rother. Learning less is more-6D camera localization via 3D surface regression. In *CVPR*, 2018. 1, 2, 5

[6] Eric Brachmann and Carsten Rother. Neural-guided RANSAC: learning where to sample model hypotheses. In *ICCV*, 2019. 1, 2

[7] Samarth Brahmbhatt, Jinwei Gu, Kihwan Kim, James Hays, and Jan Kautz. Geometry-aware learning of maps for camera localization. In *CVPR*, 2018. 1, 2

[8] Robert Castle, Georg Klein, and David W Murray. Video-rate localization in multiple maps for wearable augmented reality. In *ISWC*, 2008. 1

[9] Ondřej Chum and Jiří Matas. Optimal randomized RANSAC. *PAMI*, 30(8):1472–1482, 2008. 3

[10] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperPoint: Self-supervised interest point detection and description. In *CVPR Workshops*, 2018. 1, 2, 3, 4, 6

[11] Mingyu Ding, Zhe Wang, Jiankai Sun, Jianping Shi, and Ping Luo. Camnet: Coarse-to-fine retrieval for camera relocalization. In *ICCV*, 2019. 2

[12] Nathan Doh, Hyunga Choi, Bumchul Jang, Sangmin Ahn, Hyojin Jung, and Sungkil Lee. TeeVR: Spatial template-based acquisition, modeling, and rendering of large-scale indoor spaces. In *SIGGRAPH Emerging Technologies*, 2019. 2

[13] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-Net: A trainable CNN for joint description and detection of local features. In *CVPR*, 2019. 1, 2, 3, 6

[14] Huanhuan Fan, Yuhao Zhou, Ang Li, Shuang Gao, Jijunnan Li, and Yandong Guo. Visual localization using semantic segmentation and depth prediction. *arXiv preprint arXiv:2005.11922*, 2020. 1, 2, 3, 6

[15] Luca Ferranti, Xiaotian Li, Jani Boutellier, and Juho Kannala. Can you trust your pose? confidence estimation in visual localization. *arXiv preprint arXiv:2010.00347*, 2020. 3, 4

[16] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *CACM*, 24(6):381–395, 1981. 3

[17] Martin Humenberger, Yohann Cabon, Nicolas Guerin, Julien Morat, Jérôme Revaud, Philippe Rerole, Noé Pion, Cesar de Souza, Vincent Leroy, and Gabriela Csurka. Robust image retrieval-based visual localization using kapture. *arXiv preprint arXiv:2007.13867*, 2020. 1, 2, 3, 6

[18] Janghun Hyeon, Hyunga Choi, JooHyung Kim, Bumchul Jang, Jaehyeon Kang, and Nakju Doh. Automatic spatial template generation for realistic 3D modeling of large-scale indoor spaces. In *IROS*, 2019. 2

[19] Janghun Hyeon, Dongwoo Kim, Bumchul Jang, Hyunga Choi, Dong Hoon Yi, Kyungho Yoo, Jeongae Choi, and Nakju Doh. KR-Net: A dependable visual kidnap recovery network for indoor spaces. In *IROS*, 2020. 1, 2, 3, 5, 6

[20] Arnold Irschara, Christopher Zach, Jan-Michael Frahm, and Horst Bischof. From structure-from-motion point clouds to fast location recognition. In *CVPR*, 2009. 2, 4

[21] Sagi Katz, Ayellet Tal, and Ronen Basri. Direct visibility of point sets. *TOG*, 26(3):24–es, 2007. 5

[22] Alex Kendall and Roberto Cipolla. Geometric loss functions for camera pose regression with deep learning. In *CVPR*, 2017. 1, 2, 5

[23] Alex Kendall, Matthew Grimes, and Roberto Cipolla. PoseNet: A convolutional network for real-time 6-DOF camera relocalization. In *ICCV*, 2015. 1, 2, 5

[24] 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 *CVPR*, 2011. 3

[25] Zuzana Kukelova, Martin Bujnak, and Tomas Pajdla. Real-time solution to the absolute pose problem with unknown radial distortion and focal length. In *ICCV*, 2013. 3

[26] Viktor Larsson, Zuzana Kukelova, and Yinqiang Zheng. Making minimal solvers for absolute pose estimation compact and robust. In *ICCV*, 2017. 3

[27] Karel Lebeda, Jirı Matas, and Ondrej Chum. Fixing the locally optimized RANSAC. In *BMVC*, 2012. 3

[28] Xiaotian Li, Shuzhe Wang, Yi Zhao, Jakob Verbeek, and Juho Kannala. Hierarchical scene coordinate classification and regression for visual localization. In *CVPR*, 2020. 1, 2, 5

[29] Yunpeng Li, Noah Snavely, Dan Huttenlocher, and Pascal Fua. Worldwide pose estimation using 3D point clouds. In *ECCV*, 2012. 2, 4

[30] Yunpeng Li, Noah Snavely, and Daniel P Huttenlocher. Location recognition using prioritized feature matching. In *ECCV*, 2010. 2, 4

[31] Zhengqi Li and Noah Snavely. MegaDepth: Learning single-view depth prediction from internet photos. In *CVPR*, 2018. 6

[32] Hyon Lim, Sudipta N Sinha, Michael F Cohen, and Matthew Uyttendaele. Real-time image-based 6-DOF localization in large-scale environments. In *CVPR*, 2012. 1

[33] Ce Liu, Jenny Yuen, and Antonio Torralba. Sift flow: Dense correspondence across scenes and its applications. *PAMI*, 33(5):978–994, 2010. 3

[34] Will Maddern, Geoffrey Pascoe, Chris Linegar, and Paul Newman. 1 year, 1000 km: The oxford robotcar dataset. *IJRR*, 36(1):3–15, 2017. 2

[35] Eric Marchand, Hideaki Uchiyama, and Fabien Spindler. Pose estimation for augmented reality: a hands-on survey. *TVCG*, 22(12):2633–2651, 2015. 1

[36] Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand, and Bohyung Han. Large-scale image retrieval with attentive deep local features. In *ICCV*, 2017. 3

[37] Filip Radenović, Giorgos Tolias, and Ondřej Chum. Fine-tuning CNN image retrieval with no human annotation. *PAMI*, 41(7):1655–1668, 2018. 1, 2

[38] Jerome Revaud, Jon Almazán, Rafael S Rezende, and Cesar Roberto de Souza. Learning with average precision: Training image retrieval with a listwise loss. In *ICCV*, 2019. 1, 2, 3

[39] Jerome Revaud, Cesar De Souza, Martin Humenberger, and Philippe Weinzaepfel. R2D2: Reliable and repeatable detector and descriptor. In *NIPS*, 2019. 2, 3

[40] Ignacio Rocco, Relja Arandjelović, and Josef Sivic. Efficient neighbourhood consensus networks via submanifold sparse convolutions. *arXiv preprint arXiv:2004.10566*, 2020. 1, 2, 3, 6

[41] Ignacio Rocco, Mircea Cimpoi, Relja Arandjelović, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Neighbourhood consensus networks. In *NIPS*, 2018. 1, 3

[42] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From Coarse to Fine: Robust hierarchical localization at large scale. In *CVPR*, 2019. 1, 2, 3, 4, 6

[43] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *CVPR*, 2020. 1, 2, 3, 4, 6, 8

[44] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Improving image-based localization by active correspondence search. In *ECCV*, 2012. 4

[45] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, et al. Benchmarking 6DOF outdoor visual localization in changing conditions. In *CVPR*, 2018. 1

[46] 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 *CVPR*, 2017. 2

[47] Torsten Sattler, Tobias Weyand, Bastian Leibe, and Leif Kobbelt. Image retrieval for image-based localization revisited. In *BMVC*, 2012. 1, 2

[48] Amin Sedaghat, Mehdi Mokhtarzade, and Hamid Ebadi. Uniform robust scale-invariant feature matching for optical remote sensing images. *TGRS*, 49(11):4516–4527, 2011. 4

[49] Jean Serra. *Image analysis and mathematical morphology*. Academic Press, Inc., 1983. 5

[50] Tianxin Shi, Hainan Cui, Zhuo Song, and Shuhan Shen. Dense semantic 3D map based long-term visual localization with hybrid features. *arXiv preprint arXiv:2005.10766*, 2020. 1, 3

[51] Tianxin Shi, Shuhan Shen, Xiang Gao, and Lingjie Zhu. Visual localization using sparse semantic 3D map. In *ICIP*, 2019. 1, 3

[52] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in RGB-D images. In *CVPR*, 2013. 3, 5

[53] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 6

[54] Hajime Taira, Masatoshi Okutomi, Torsten Sattler, Mircea Cimpoi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, and Akihiko Torii. InLoc: Indoor visual localization with dense matching and view synthesis. In *CVPR*, 2018. 1, 2, 3, 5, 6, 7

[55] Hajime Taira, Ignacio Rocco, Jiri Sedlar, Masatoshi Okutomi, Josef Sivic, Tomas Pajdla, Torsten Sattler, and Akihiko Torii. Is this the right place? geometric-semantic pose verification for indoor visual localization. In *ICCV*, 2019. 1, 2, 3, 4, 6

[56] Julien Valentin, Angela Dai, Matthias Nießner, Pushmeet Kohli, Philip Torr, Shahram Izadi, and Cem Keskin. Learning to navigate the energy landscape. In *3DV*, 2016. 3, 5

[57] Erik Wijmans and Yasutaka Furukawa. Exploiting 2D floorplan for building-scale panorama RGBD alignment. In *CVPR*, 2017. 5

[58] Qing Zhu, Bo Wu, and Zhi-Xiang Xu. Seed point selection method for triangle constrained image matching propagation. *GRSL*, 3(2):207–211, 2006. 4