This CVPR 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.

IMP: Iterative Matching and Pose Estimation with Adaptive Pooling

Fei Xue Ignas Budvytis Roberto Cipolla
University of Cambridge
{fx221, ib255, rc10001}@cam.ac.uk

Abstract

*Previous methods solve feature matching and pose estimation using a two-stage process by first finding matches and then estimating the pose. As they ignore the geometric relationships between the two tasks, they focus on either improving the quality of matches or filtering potential outliers, leading to limited efficiency or accuracy. In contrast, we propose an **i**terative **m**atching and **p**ose estimation framework (IMP) leveraging the geometric connections between the two tasks: a few good matches are enough for a roughly accurate pose estimation; a roughly accurate pose can be used to guide the matching by providing geometric constraints. To this end, we implement a geometry-aware recurrent attention-based module which jointly outputs sparse matches and camera poses. Specifically, for each iteration, we first implicitly embed geometric information into the module via a pose-consistency loss, allowing it to predict geometry-aware matches progressively. Second, we introduce an **e**fficient IMP, called EIMP, to dynamically discard keypoints without potential matches, avoiding redundant updating and significantly reducing the quadratic time complexity of attention computation in transformers. Experiments on YFCC100m, Scannet, and Aachen Day-Night datasets demonstrate that the proposed method outperforms previous approaches in terms of accuracy and efficiency. Code is available at* https://github.com/feixue94/imp-release

1. Introduction

Feature matching and relative pose estimation are two fundamental tasks in computer vision and especially important to visual localization and 3D reconstruction. Traditionally, the two tasks are performed in two stages separately by first finding correspondences between keypoints extracted from two images with nearest neighbor (NN) matching and then estimating the relative pose from predicted matches with robust estimators, *e.g.* RANSAC [6, 7, 20, 32]. This pipeline has been the de-facto standard framework for decades [5]. However, due to repetitive textures/structures, changing appearances and viewpoint variations, matches given by NN often contain a large number of outliers, leading to poor pose accuracy [36, 37]. To mitigate this problem, some works [8, 10, 16, 27, 40, 48, 49, 52] filter potential outliers of predicted matches with neural networks to improve the pose accuracy. Although they report better results, their performance is limited by the quality of initial matches and require extra time for filtering at test time. Alternatively, advanced matchers such as SuperGlue [36] enhance the matching quality directly by using global information from all keypoints via transformers [45] with a fixed number (*e.g.* 9) of iterations. These methods have obtained remarkable performance. Yet, their quadratic time complexity for the attention computation degrades the efficiency in real applications. Some following works [12, 38, 41] explore more efficient variations, they run faster but are significantly less accurate (see Table 1 and 3).

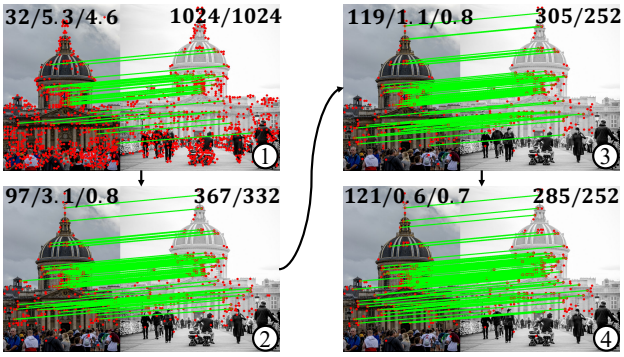


Figure 1. **Process of iterative matching and pose estimation**. For each image pair, we report the number inliers/rotation/translation errors (top-left) and retained keypoints in left/right images (top-right) at iterations from 1 to 4. In the iterative process, our method finds more inliers spanning almost the whole image, estimates increasingly precise pose and discards keypoints without true matches gradually.

In this paper, we aim to introduce an efficient and accurate framework for iterative matching and pose estimation. Our approach is built upon the following observations: (1) a few well distributed matches (*e.g.* 5) could give a roughly accurate pose (*e.g.* essential matrix); (2) in turn, a roughly

accurate pose could provide strong geometric constraints (*e.g*. epipolar line) to find more accurate matches at low cost; (3) the pose also reveals which keypoints have potential correspondences, preventing redundant operations. Based on the geometric connections of the two tasks, we propose an **i**terative **m**atching and **p**ose estimation framework (IMP), to perform matching and pose estimation iteratively as opposed to in two separate stages. Specifically, we progressively augment descriptors with self and cross attention as [12,36,38,41], find matches and estimate the relative pose. As descriptors get gradually more discriminative, more correct matches can be found, leading to increasingly more precise pose, as shown in Fig. 1. However, due to the noise [26] and degeneration (*e.g*. co-planar keypoints) [13], not all inliers could give a good pose [4, 18]. In addition to the classification loss mainly used by prior methods [12,36], we apply a pose-consistency loss [49] to the matching process, enabling the model to find matches which are not only accurate but also able to give a good pose.

Moreover, in order to avoid redundant operations on uninformative keypoints, we employ a sampling strategy by combining the matching and attention scores of keypoints and the uncertainty of predicted poses to adaptively remove useful keypoints, as shown in Fig. 1. Compared with prior sampling approaches [19, 44] based mainly on attention scores, our adaptive strategy overcomes the over-sampling problem effectively. Our framework reduces the time cost from two aspects. First, in contrast to adopting a fixed number of iterations for all cases [12, 36, 38], it runs fewer iterations for easy cases with few viewpoint or appearance changes and more for challenging cases. Second, it reduces the cost of each iteration, significantly reducing the quadratic time complexity of attention computation. We also show that discarding potential outliers increases not only efficiency but also accuracy (see Sec. 5). The efficient version of IMP is called EIMP. Ours contributions are as follows:

- We propose to perform geometry-aware matching and pose estimation iteratively, allowing the two tasks to boost each other in an iterative manner.
- We adopt a robust sampling strategy to adaptively discard redundant keypoints in the iteration process, significantly decreasing the time complexity.
- We apply the pose uncertainty to the sampling strategy, which further improves the accuracy matching and pose estimation.

Our experiments on relative pose estimation and large-scale localization tasks demonstrate that our method outperforms previous competitors and is more efficient. We organize the rest of the paper as follows. In Sec. 2, we discuss related works. In Sec. 3, we give a detailed description of our method. We test the performance of our model in Sec. 5 and conclude the paper in Sec. 6.

2. Related works

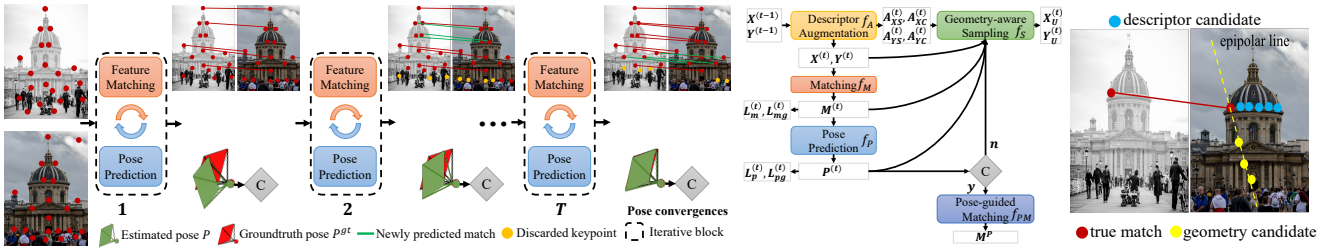
In this section, we discuss related work on local feature matching, efficient attention, and outlier filtering.

Local feature matching. Traditionally, NN matching and its variants, *e.g*., mutual NN (MNN) and NN with ratio test (NN-RT) [28] are widely used for finding correspondences between two sets of keypoints [2, 17, 28, 34]. Since they perform point-wise matching, they are fast but not robust to large viewpoint and appearance changes, *e.g*. illumination and season variations. Recently, SuperGlue [36] utilizes transformers [45] with self and cross attention to embed spatial information and achieves remarkable performance. Despite its excellent accuracy, its limitations are twofold. First, the complexity of attention is quadratic to the number of keypoints, degrading the efficiency in real applications. Second, it adopts a fixed number (9) of layers for message propagation for all input cases. This causes extra time cost especially for simple cases with few viewpoint and appearance changes, which require much fewer number of iterations to find potential inliers. Therefore, running a fixed number iterations for all cases leads to redundancy.

Some variants [12, 38, 41] try to solve the first problem by using a fixed number of seeded keypoints for message propagation [12], performing cluster-wise matching [38] or aggregating local and global information separately [41]. In spite of their high efficiency, they lose significant accuracy. Besides, like SuperGlue, they also update keypoints without correspondences redundantly in each iteration and adopt a fixed number of iterations for all cases constantly. Unlike these approaches, our model utilizes predicted poses to prevent extra iterations for easy cases. Additionally, as our method removes keypoints without true matches adaptively to reduce time complexity in each iteration, both efficiency and accuracy are guaranteed.

Efficient attention. Many works are proposed to mitigate the quadratic time complexity of the attention mechanism [45]. They reduce the complexity by learning a linear projection function [22, 46], a token selection classifier [19, 24, 33] or shared attention [11], to name a few. We refer the reader a comprehensive survey [42] for more details. Since most of these methods are designed to extract high-level features for downstream tasks *e.g*. image recognition [1, 9, 11, 19, 22, 44] usually with a fixed number of tokens, directly transferring these techniques to the matching task which has dynamic numbers of keypoints as input could cause performance loss. Instead, we utilize the geometric properties of the matching task to adaptively discard redundant keypoints. Our strategy depends mainly on the number of inliers in the input and is completely adaptive.

Outlier filtering. Ratio test [28] is often used to remove



(a) Pipeline of our method (b) tth iterative block (c) Pose-guided matching

Figure 2. **Pipeline of our method.** In the iteration process, descriptors are gradually augmented by our recurrent attention-based module. Augmented descriptors are then used to compute matches $\mathcal{M}^{(t)}$, which are further used for estimating the pose $P^{(t)}$. As descriptors become more discriminative, more correct matches can be found, leading to more precise poses. The pose is utilized to provide geometric guidance to find more matches and discard redundant keypoints with geometry-aware sampling f_S, making next iterations faster. If the pose converges, iteration stops. Our recurrent attention-based module at tth iteration (b) and pose-guided matching (c) are also visualized.

matches with high uncertainties. Recently, CNNs are used to leverage the spatial context of input keypoints to filter potential outliers [8,10,16,27,40,48,49,52]. ACNe [40] introduces an attention mechanism to gather information from other keypoints for filtering. OANet [49] embeds geometric priors to further improve the performance. LMCNet [27] and CLNet [51] use local motion consistency to filter inconsistent matches. MS2DG-Net [16] infuses semantics into a graph network to enhance the accuracy. These methods have achieved better results than ratio test and could serve as an additional module to filter potential outliers. However, their performance relies heavily on the quality of initial matches and require extra cost for filtering at test time. In contrast, our method embed geometric information directly into the matching module, allowing the model to predict accurate and pose-ware correspondences end-to-end.

3. Method

We first give an introduction to performing iterative transformer-based matching in Sec. 3.1. Then, we describe our iterative pose estimation framework, the adaptive sampling strategy and application at test time in Sec. 3.2, Sec. 3.3 and Sec 3.4, respectively. We visualize the pipeline of our framework in Fig. 2a.

3.1. Iterative transformer-based matching

In this section, we introduce how to perform transformer-based matching iteratively.

Problem formulation. Given two sets of keypoints (*e.g.* SuperPoint [17] or SIFT [28]) $\mathcal{X} = \{x_1, x_2, ..., x_m\}, \mathcal{Y} = \{y_1, y_2, ..., y_n\}$ (m, n are the number of keypoints) extracted from two images, matchers predict matches between $\mathcal{X}$ and $\mathcal{Y}$, denoted as $\mathcal{O} = \{o_1, o_2, ..., o_k\}$, where $o_j = (x_k, y_l)$ is the matched pair. Each keypoint $x_i = (u_i, v_i, c_i, \mathbf{d}_i)$ comprises its 2D coordinates (u_i, v_i), confidence c_i and descriptor $\mathbf{d}_i \in R^d$ (d is the descriptor size).

Descriptor augmentation. As [12,36,38], for each keypoint x_i, its coordinates (u_i, v_i) and confidence c_i are encoded into a high-dimension vector with a multi-layer perception (MLP) f_{enc}, which is then added to its descriptor $\mathbf{d}_i$, as: $\mathbf{d}_i^{'} = \mathbf{d}_i + f_{enc}(u_i, v_i, c_i)$. $\mathbf{d}_i^{'}$ is used to replace $\mathbf{d}_i$ as input for descriptor augmentation, as:

$$\mathcal{A}_{XS}^{(t)} = softmax(\frac{f_q^S(\mathcal{X}^{(t-1)})(f_k^S(\mathcal{X}^{(t-1)}))^T}{\sqrt{d}}), \quad (1)$$

$$\mathcal{A}_{XC}^{(t)} = softmax(\frac{f_q^C(\mathcal{X}^{(t-1)})(f_k^C(\mathcal{Y}^{(t-1)}))^T}{\sqrt{d}}), \quad (2)$$

$$\mathcal{X}^{(t)} = \mathcal{X}^{(t-1)} + f_{mlp}^S(f_p^S(\mathcal{A}_{XS}^{(t)}(f_v^S(\mathcal{X}^{(t-1)})))||\mathcal{X}^{(t-1)}) + f_{mlp}^C(f_p^C(\mathcal{A}_{XC}^{(t)}(f_v^C(\mathcal{Y}^{(t-1)})))||\mathcal{X}^{(t-1)}). \quad (3)$$

$\mathcal{A}_{XS}^{(t)}$ and $\mathcal{A}_{XC}^{(t)}$ are self (S) and cross (C) attention matrices for $\mathcal{X}^{(t)}$. $f_{q/k/v/p}^{S/C}$ are FC layers. d is descriptor dimension. $f_{mlp}^{S/C}$ are 3-layer MLPs. $||$ is channel-wise concatenation. We use sharing attention mechanism [11] to further augment descriptors with pre-computed attention matrices:

$$\mathcal{X}^{(t)} = \mathcal{X}^{(t)} + f_{mlp}^S(f_p^S(\mathcal{A}_{XS}^{(t)}(\bar{f}_v^S(\mathcal{X}^{(t-1)})))||\mathcal{X}^{(t-1)}) + f_{mlp}^C(\phi_p^C(\mathcal{A}_{XC}^{(t)}(\bar{f}_v^C(\mathcal{Y}^{(t-1)})))||\mathcal{X}^{(t-1)}). \quad (4)$$

$\bar{f}_v^{S/C}$ are FC layers as well. We conduct the same operations to $\mathcal{Y}^{(t-1)}$ to obtain augmented descriptors $\mathcal{Y}^{(t)}$, self $\mathcal{A}_{YS}^{(t)}$ and cross $\mathcal{A}_{YC}^{(t)}$ attention matrices. As shown in Fig. 2b, $\mathcal{X}^{(t)}$ and $\mathcal{Y}^{(t)}$ are augmented descriptors used for the next iteration. $\mathcal{A}_{XS}^{(t)}$, $\mathcal{A}_{XC}^{(t)}$, $\mathcal{A}_{YS}^{(t)}$ and $\mathcal{A}_{YC}^{(t)}$ are self and cross attention matrices used for message propagation in the augmentation function f_A. In prior works [12,36], self and cross attention matrices are only used for augmenting descriptors. However, in our model, they are further utilized for adaptive sampling (Sec. 3.3).

Iterative matching prediction. Augmented descriptors $\mathcal{X}^{(t)}, \mathcal{Y}^{(t)}$ are used to compute the matching matrix $\mathcal{M}^{(t)} \in R^{m \times n}$ with function f_M. f_M first computes Euclidean distance $\mathcal{D}^{(t)} \in R^{m \times n}$ for all pairs in $\mathcal{X}_A^{(t)} \times \mathcal{Y}_A^{(t)}$ and then uses Sinkhorn algorithm [14,39] to obtain $\mathcal{M}^{(t)}$ by optimizing $\mathcal{D}^{(t)}$ a transport problem [30]. Matches with scores over than a certain threshold θ_m are deemed as predicted matches. In our framework, we predict matches at

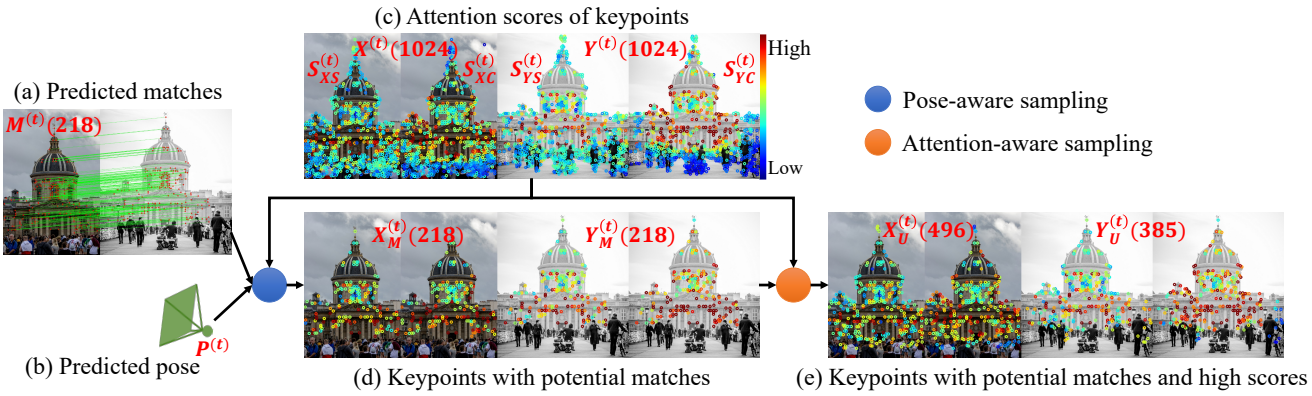


Figure 3. **Adaptive sampling.** At iteration t, predicted matches $\mathcal{M}^{(t)}$ and pose $P^{(t)}$ are used to select keypoints with potential correspondences. These selected keypoints are expanded by those with high self or cross attention scores. The finally preserved keypoints are those with potential matches and high contribution, guaranteeing both the efficiency and accuracy.

each iteration t for pose estimation and keypoint sampling as opposed to at only the last iteration [36, 38].

Iterative matching loss. Similar to SuperGlue [36], we adopt the classification loss of minimizing the negative log-likelihood of the matching matrix to enforce the network to predict correct matches for each iteration t, as:

$$L_m^{(t)} = -\sum_{(i,j)\in\mathcal{M}^{gt}} log\bar{\mathcal{M}}_{i,j}^{(t)} - \sum_{i\in\bar{\mathcal{M}}^{gt}} log\bar{\mathcal{M}}_{i,n+1}^{(t)} - \sum_{j\in\bar{\mathcal{M}}^{gt}} log\bar{\mathcal{M}}_{m+1,j}^{(t)}. \quad (5)$$

$\bar{\mathcal{M}}^{(t)}$ is the expanded matrix of $\mathcal{M}^{(t)}$ with an additional row and column for unmatched keypoints [36] in $\mathcal{X}^{(t)}$ and $\mathcal{Y}^{(t)}$. $\mathcal{M}^{gt}$ and $\bar{\mathcal{M}}^{gt}$ are the groundtruth matching matrix and its expansion. Applying the matching loss to each iteration t enables our model to predict correct matches at each iteration progressively.

3.2. Iterative pose estimation

In this section, we give a detailed description of how to predict poses and the usage of predicted poses for enhancing the matching process in each iteration, as shown in Fig. 2b.

Pose-consistency loss. Because of the noise [26] and degenerated keypoints [13], not all correct matches could give a good pose [4, 18]. Eq (5) only guarantees that keypoints with more discriminative descriptors have higher matching scores and are recognized first to attend pose estimation, but ignore the geometric requirements needed for robust pose estimation. Therefore, directly using all potential inliers with matching score over than θ_m for pose estimation could be inaccurate. Instead, we implicitly infuse geometric information into matching transformers, enforcing the matching module to focus first on matches which are not only correct but also have high probability to give a good pose.

To this end, at each iteration t, matches with score over θ_m in $\mathcal{M}^{(t)}$ along with their scores are used for fundamental matrix estimation with weighted 8 points function f_P, as:

$$P^{(t)} = f_P(\{(x_i^{(t)}, y_i^{(t)}, s_i^{(t)})\}). \quad (6)$$

$P^{(t)} \in R^{3\times3}$ is the predicted fundamental matrix and $\{(x_i^{(t)}, y_i^{(t)}, s_i^{(t)})\}$ are predicted matches with matching score $s_i^{(t)} = M^{(t)}[idx(x_i^{(t)}), idx(y_i^{(t)})]$ ($idx(.)$ is the index function). We enforce the geometric consistency between $P^{(t)}$ and the groundtruth P^{gt} by minimizing the pose and epipolar errors jointly, as:

$$L_p^{(t)} = ||P^{(t)} - P^{gt}||_2, \quad (7)$$

$$L_{pg}^{(t)} = \frac{1}{N_{gt}}\sum_k f_{epipolar}(P^{(t)}, x_k^{gt}, y_k^{gt}), \quad (8)$$

$$L_{mg}^{(t)} = \frac{1}{N_p^{(t)}}\sum_k f_{epipolar}(P^{gt}, x_k^{(t)}, y_k^{(t)}). \quad (9)$$

$f_{epipolar}$ is Sampson distance [21]. (x_k^{gt}, y_k^{gt}) and $(x_k^{(t)}, y_k^{(t)})$ are the groundtruth and predicted matches. $L_p^{(t)}$ and $L_{pg}^{(t)}$ enforce the predicted pose $P^{(t)}$ to be correct with constraints provided by the groundtruth pose P^{gt} and matches (x_k^{gt}, y_k^{gt}), respectively. $L_{mg}^{(t)}$ additionally ensures the correctness of predicted matches $(x_k^{(t)}, y_k^{(t)})$ with groundtruth pose P^{gt}. N_{gt} and $N_p^{(t)}$ are the number of groundtruth and predicted matches.

For each iteration, our final loss combines $L_m^{(t)}$, $L_{pg}^{(t)}$ and $L_{mg}^{(t)}$ with weights of α_m, α_p and α_g, as:

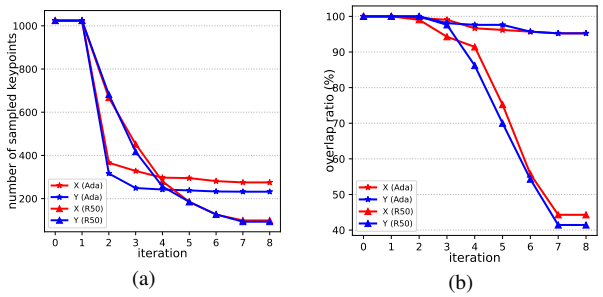
$$L^{(t)} = \alpha_m L_m^{(t)} + \alpha_p L_p^{(t)} + \alpha_g(L_{pg}^{(t)} + L_{mg}^{(t)}). \quad (10)$$

We apply $L^{(t)}$ to each iteration and compute the total loss over T iterations, as $L_{total} = \frac{1}{T}\sum_t^T L^{(t)}$.

3.3. Adaptive geometry-aware sampling

In fact, lots of keypoints are uninformative and a large number of keypoints don't have correspondences in the other image (about 70% of 2k keypoints in YFCC100m dataset [43]). Updating these keypoints leads to extra time cost, so we propose an effective strategy to remove these keypoints, as shown in Fig.3.

Informative keypoints. The information contained by each keypoint is defined by its contribution to others in the



(a) (b)

Figure 4. **Number of Retained keypoints and overlap ratio.** We show (a) the number of retained keypoints and (b) ratio of inliers in two sets at each iteration. Compared to R50, the adaptive sampling reduces redundant keypoints more effectively at early stages (a) and preserves much more inliers at latter iterations (b).

attention matrix $\mathcal{A}_{XS}^{(t)} \in R^{m\times n\times h}$ (m, n are the number of keypoints in the query and key and h is the number of heads). As [42, 44], we compute the score of each keypoint by averaging values along the head and key dimension, as $S_{XS}^{(t)} = \frac{1}{n*h}\sum_{(i,j)}\mathcal{A}_{XS.,i,j}$. $S_{XS} \in R^m$ is normalized with sum of 1. We also compute scores from $\mathcal{A}_{XC}^{(t)}$, $\mathcal{A}_{YS}^{(t)}$, $\mathcal{A}_{YC}^{(t)}$ as $S_{XC}^{(t)}$, $S_{YS}^{(t)}$, $S_{YC}^{(t)}$. Self and cross attention scores of 1024 keypoints in $\mathcal{X}^{(t)}$ and $\mathcal{Y}^{(t)}$ are visualized in Fig. 3c. Although attention scores can be directly used for sampling keypoints by keeping a certain ratio of keypoints with the highest scores [24, 44], this strategy is prone to over-pruning. Instead, we use keypoints with potential matches as guidance to mitigate this problem.

Adaptive sampling. At iteration t, the matching matrix $\mathcal{M}^{(t)}$ containing matching confidence of all pairs, reveals which keypoints potentially have true correspondences. We visualize the potential matches and scores of self and cross attention in Fig. 3a and Fig. 3d.

Based on matching matrix $\mathcal{M}^{(t)}$, we generate two subsets $\mathcal{X}_{\mathcal{M}}^{(t)} \subseteq \mathcal{X}^{(t)}$ and $\mathcal{Y}_{\mathcal{M}}^{(t)} \subseteq \mathcal{Y}^{(t)}$ which contain matched keypoints (matching score over θ_m). Since keypoints in $\mathcal{X}_{\mathcal{M}}^{(t)}$ and $\mathcal{Y}_{\mathcal{M}}^{(t)}$ are potential inliers, they can provide guidance to find more informative ones. In detail, let $S_{MSX}^{(t)} = \{S_{XS}^{(t)}[idx(x)], s.t. \quad x \in \mathcal{X}_{\mathcal{M}}^{(t)}\}$ be self-attention scores of keypoints in $\mathcal{X}_{\mathcal{M}}^{(t)}$. We generate another set of keypoints with high self-attention scores as $\mathcal{X}_{S}^{(t)} = \{x, s.t. \quad x \in \mathcal{X}^{(t)}, S_{XS}^{(t)}[idx(x)] \geq f_{md}(S_{MSX}^{(t)})\}$ (f_{md} returns the median value). By repeating this process, we obtain another subset from $\mathcal{X}^{(t)}$ with high cross-attention scores as $\mathcal{X}_{C}^{(t)}$ and two sets $\mathcal{Y}_{S}^{(t)}$, $\mathcal{Y}_{C}^{(t)}$ from $\mathcal{Y}^{(t)}$ with high self and cross attention scores as well. The final sets are the union of informative keypoints and those with matches, as $\mathcal{X}_{U}^{(t)} = \mathcal{X}_{\mathcal{M}}^{(t)} \cup \mathcal{X}_{S}^{(t)} \cup \mathcal{X}_{C}^{(t)}, \mathcal{Y}_{U}^{(t)} = \mathcal{Y}_{\mathcal{M}}^{(t)} \cup \mathcal{Y}_{S}^{(t)} \cup \mathcal{Y}_{C}^{(t)}$. As shown in Fig. 3e, $\mathcal{X}_{U}^{(t)}, \mathcal{Y}_{U}^{(t)}$ will replace $\mathcal{X}^{(t)}, \mathcal{Y}^{(t)}$ to join the next iteration. The sampling reduces the number of keypoints from 1024 to 496 and 358 in two sets, reducing the time cost significantly for next iterations.

Fig. 4a visualizes the number of keypoints in $\mathcal{X}_{U}^{(t)}$ and $\mathcal{Y}_{U}^{(t)}$. Fig. 4b shows the value of $\frac{|\mathcal{X}_{U}^{(t)} \cap X_F|}{|X_F|}$ and $\frac{|\mathcal{Y}_{U}^{(t)} \cap Y_F|}{|Y_F|}$ (X_F and Y_F are two sets of matched keypoints without any sampling; $|.|$ indicates the number of elements), which measure the ability of retaining inliers. For comparison, we also generate two sets of keypoints based on sampling ratio by choosing the top 50% keypoints with highest self and cross attention scores as $\mathcal{X}_{R50S}^{(t)}$, $\mathcal{X}_{R50C}^{(t)}$, $\mathcal{Y}_{R50S}^{(t)}$, and $\mathcal{Y}_{R50C}^{(t)}$. The final sampled sets are $\mathcal{X}_{R50}^{(t)} = \mathcal{X}_{R50S}^{(t)} \cup \mathcal{X}_{R50C}^{(t)}$ and $\mathcal{Y}_{R50}^{(t)} = \mathcal{Y}_{R50S}^{(t)} \cup \mathcal{Y}_{R50C}^{(t)}$. The number of retained keypoints and overlap ratios are also visualized.

Fig. 4a shows the adaptive strategy performs more effectively at discarding keypoints at early stages and preserving keypoints at later iterations. Fig. 4b illustrates that although R50 has the close ratio to the adaptive strategy at the 2nd and 3rd iterations by keeping more redundant keypoints, the former loses overlap ratio dramatically after 4 iterations while the latter still keeps the ratio over 90%.

Adaptive sampling with pose uncertainty. The matching matrix $\mathcal{M}^{(t)}$ might not be very accurate at the first several iterations when descriptors are not discriminative, impairing the accuracy. To mitigate the problem, we make use of the predicted pose. Note, if most of the predicted matches are correct, the pose is more precise with higher inlier ratio. However, if predicted matches contain many outliers, the pose is probably not accurate and has low inlier ratio. Consequently, we define the uncertainty of the pose on its consistency with matches, as $r^{(t)} = \frac{|\{(x_i,y_i),s.t.f_{epipolar}(P^{(t)},x_i,y_i)\leq\theta_e\}|}{|\{(x_i,y_i,s_i),s.t.s_i\geq\theta_m\}|}$ (θ_e is threshold determining inliers). We use $r^{(t)}$ to adjust the sampling threshold θ_m, as $\theta_m^{(t)} = \theta_m * r^{(t)}$, allowing the model to dynamically sample fewer keypoints when both matches and pose are accurate and more when they are not.

3.4. Iterative process at test time

At test time, after each iteration, we compute matches $\mathcal{M}^{(t)}$ and estimate the pose from matches with RANSAC [20]. We adopt the relative error between consecutively predicted pose $P^{(t)}$ and $P^{(t-1)}$ as stop criteria to determine if continuing iteration. Specifically, if the maximum angular errors of rotations and translations is less than θ_P, the iteration stops. Benefiting from the embedded geometry information, for about 55% of the cases in YFCC100m dataset [43], 6 iterations rather than 9 are enough to find a good pose. The predicted pose is further used for adaptive sampling at test time (Sec. 3.3).

Once a good pose is predicted, more matches can be found from the guidance of the pose with function f_{PM}. f_{MP} first computes the epipolar distance for all pairs in $\mathcal{X}^{(t)}, \mathcal{Y}^{(t)}$ as $\mathcal{M}_e^{(t)}$ where

$\mathcal{M}_{eij}^{(t)} = f_{epipolar}(\mathcal{X}_i^{(t)}, \mathcal{Y}_j^{(t)})$ and $f_{epipolar}(x,y) = \frac{(y^T P^{(t)} x)^2}{(P^{(t)}x)_1^2+(P^{(t)}x)_2^2+(P^{(t)T}y)_1^2+(P^{(t)T}y)_2^2}$. Then $\mathcal{M}_e^{(t)}$ is binarized with errors smaller than $12px$ as 1 otherwise 0 as $\bar{\mathcal{M}}_e^{(t)}$. Finally, the pose-guided matching matrix is obtained as $\mathcal{M}^P = \mathcal{M}^{(t)} \bigodot \bar{\mathcal{M}}_e^{(t)}$ ($\bigodot$ is element-wise multiplication). As shown in Fig. 2c, matches with high uncertainties caused by similar descriptors at regions with repetitive textures can be effectively mitigated by pose constraints. These matches usually require more number iterations to be found.

4. Experiment Setup

Training. We implement the model in PyTorch [29] and train it on MegaDepth dataset [25] for 900k iterations with Adam optimizer [23], batch size of 16, initial learning rate of 0.0001. The learning rate decays with ratio of 0.99996 after 200k iterations and is fixed at 0.00001. We use 1024 SuperPoint [17] and RootSIFT [2, 28] keypoints for training. As with SuperGlue [36], our network has T (9) iterative blocks. α_m, α_p, α_g, θ_m, θ_P, θ_e are set to 0.6, 0.2, 0.2, 0.2, 1.5 and 0.005, respectively. Note that unlike SuperGlue which is first trained on Oxford-Paris dataset [31] and then fine-tuned on Scannet [15] and MegaDepth [25] datasets for indoor and outdoor models respectively, our models are trained only on the MegaDepth dataset from scratch.

Datasets and metrics. We first test our method on YFCC100m [43] and Scannet [15] datasets to evaluate the performance on relative pose estimation. YFCC100m is a large-scale outdoor dataset consisting of images with large illumination and season changes and viewpoint variations. Scannet is an indoor dataset widely used for depth prediction [3] and pose estimation [12, 36]. For relative pose estimation, we report the accurate cumulative error curve (AUC) [36] at thresholds of 5°, 10°, and 20°. The error is the maximum angular errors of rotations and normalized translations. We also report the mean matching score (M.S.) and mean precision (Prec.) of matches.

We additionally test our method on large-scale localization task at Aachen Day-Night (v1.0 and v1.1) datasets [37, 50]. Aachen v1.0 dataset comprises of 4,328 reference and 922 query (824 day, 98 night) images. Aachen v1.1 extends v1.0 by adding additional 2,369 reference and 93 night query images. We use HLoc [35] pipeline for mapping and localization and report the accuracy at error thresholds of $0.25m/2°$, $0.5m/5°$, and $5m/10°$.

Baselines. The baselines comprise of simple matchers such as MNN and NN-RT [28]. Also, filtering-based methods including OANet [49], AdaLAM [10], CLNet [51] and LMCNet [27] are also included. The final part is transformer-based matchers such as SuperGlue [36], SGMNet [12], and ClusterGNN [38]. As it is difficult to reproduce the results of SuperGlue from custom training (official training code is not released), we follow SGMNet and show results of original SuperGlue (SuperGlue*) and the model trained by SGMNet (SuperGlue).

Feature	*Matcher*	@5°	@10°	@20°	M.S.(%)	Prec.(%)
RootSIFT [28]	NN-RT [28]	26.7	43.2	59.4	4.4	56.4
	AdaLAM (4k) [10]	27.5	44.5	60.5	6.3	84.3
	OANet [49] [49]	22.4	36.3	50.3	5.6	53.7
	CLNet [51]	33.0	52.1	68.5	7.8	75.2
	LMCNet [27]	35.8	55.6	72.2	-	86.9
	SuperGlue* [36]	35.3	56.1	**73.6**	-	-
	SuperGlue [12, 36]	35.1	54.2	70.9	16.6	81.7
	SGMNet [12]	34.8	54.1	70.9	17.1	86.1
	ClusterGNN [38]	32.8	50.3	65.9	-	-
	IMP	36.7	**56.6**	72.9	**18.0**	87.6
	EIMP	**36.8**	56.3	72.8	13.7	**89.8**
SuperPoint [17]	MNN	6.5	15.4	28.5	16.2	16.2
	AdaLAM (2k) [10]	20.8	36.5	51.9	10.9	72.0
	OANet [49]	19.2	34.5	50.3	9.4	62.1
	CLNet [51]	27.8	46.4	63.8	11.9	75.1
	LMCNet [27]	17.4	33.2	51.1	-	**88.9**
	SuperGlue* [36]	37.1	57.2	73.6	21.7	88.5
	SuperGlue [12, 36]	33.2	53.5	70.8	19.7	78.7
	SGMNet [12]	33.0	53.0	70.0	22.3	85.1
	ClusterGNN [38]	35.3	56.1	73.6	-	-
	IMP	**39.4**	**59.4**	**75.2**	**23.0**	84.9
	EIMP	37.9	57.9	74.0	19.9	88.4

Table 1. **Results on YFCC100m dataset [43].** We report the AUC of relative poses at error thresholds of 5°, 10°, and 20°. The mean matching score (M.S.) and mean matching accuracy (Prec.) are also reported. The **best** and second best results are highlighted.

5. Experiment Results

In this section, we first show results on relative pose estimation and localization tasks in Sec. 5.1 and Sec. 5.2. Next, we analyze the computational cost in Sec. 5.3. Finally, we conduct a full ablation study to test each component in our framework in Sec. 5.4.

5.1. Relative pose estimation

Comparison with filtering-based methods. Table 1 shows our IMP and efficient IMP (EIMP) give higher accuracy than previous filtering-based approaches such as OANet [49] and LMCNet [27] for both RootSIFT [2, 28] and SuperPoint [17] features. Because these filtering-based methods rely purely on the geometric information to filter potentially wrong correspondences, their performance is influenced heavily by the quality of initial matches. In contrast, we use both the geometric information and descriptors for matching, which are able to give more accurate matches, resulting in more precise poses.

Comparison with matchers. Augmented with spatial information, SuperGlue [36] and its variants [12, 38] outperform MNN and NN-RT obviously. Although SGMNet [12] and ClusterGNN [38] are faster, they give worse accuracy than SuperGlue* due to the loss of information in the message propagation process. With implicitly embedded geometric information, our IMP gives more pose-aware matches and hence outperforms SuperGlue* espe-

Feature	*Matcher*	@5°	@10°	@20°	M.S.(%)	Prec.(%)
RootSIFT [28]	NN+RT [28]	9.1	19.8	32.7	2.3	28.8
	AdaLAM (4k) [10]	8.2	18.6	31.0	3.1	**47.6**
	OANet [49]	10.7	23.1	37.4	3.2	36.9
	CLNet [51]	5.8	15.1	26.8	2.0	43.9
	LMCNet [27]	8.5	19.3	32.4	-	47.0
	SuperGlue* [36]	-	-	-	-	-
	SuperGlue [12, 36]	14.7	29.4	45.6	8.4	42.2
	SGMNet [12]	14.4	29.9	46.0	**8.8**	45.6
	IMP	**15.6**	**30.9**	**47.4**	5.8	42.0
	EIMP	15.3	30.8	46.6	7.7	45.7
SuperPoint [17]	MNN	9.4	21.6	36.4	13.3	30.2
	AdaLAM (2k) [10]	6.7	15.8	27.4	13.2	44.2
	OANet [49]	10.0	25.1	38.0	10.6	44.6
	CLNet [51]	4.1	11.0	21.6	8.6	44.2
	LMCNet [27]	14.6	**33.6**	**53.6**	-	36.8
	SuperGlue* [36]	16.2	32.6	49.3	16.8	47.9
	SuperGlue [12, 36]	12.0	26.3	42.4	15.0	45.9
	SGMNet [12]	16.4	32.1	48.7	**17.0**	**48.1**
	IMP	**16.6**	33.1	49.4	15.8	42.0
	EIMP	15.9	32.4	48.9	15.9	46.2

Table 2. **Results on Scannet dataset [15].** We show the AUC of relative poses at error thresholds of 5°, 10°, and 20°. The mean matching score (M.S.) and mean matching accuracy (Prec.) are also reported. The **best** and second best results are highlighted.

Feature	*Matcher*	Day $(0.25m, 2°)/(0.5m, 5°)/(5m, 10°)$	Night $(0.25m, 2°)/(0.5m, 5°)/(5m, 10°)$
SuperPoint [17]	MNN	85.4 / 93.3 / 97.2	75.5 / 86.7 / 92.9
	OANet [49]	-	77.6 / 86.7 / 98.0
	AdaLAM [10]	-	78.6 / 86.7 / 98.0
	ELAM [41]	-	78.6 / 87.8 / 96.9
	SuperGlue [12, 36]	-	76.5 / 88.8 / 99.0
	SuperGlue* [36]	89.6 / 95.4 / 98.8	**86.7** / 93.9 / **100.0**
	SGMNet [12]	86.8 / 94.2 / 97.7	83.7 / 91.8 / 99.0
	ClusterGNN [38]	89.4 / 95.5 / 98.5	81.6 / 93.9 / **100.0**
	IMP	89.1 / 95.4 / 99.0	**86.7** / **94.9** / **100.0**
	EIMP	**90.0** / **96.5** / **99.2**	84.7 / **94.9** / **100.0**
SuperPoint [17]	MNN	87.9 / 93.6 / 96.8	70.2 / 84.8 / 93.7
	AdaLAM [10]	-	73.3 / 86.9 / 97.9
	SuperGlue* [36]	89.8 / **96.6** / **99.4**	75.9 / 90.1 / **100.0**
	SGMNet [12]	88.7 / 96.2 / 98.9	75.9 / 89.0 / 99.0
	IMP	89.1 / 95.4 / 99.0	75.9 / **92.7** / 99.5
	EIMP	**90.0** / 96.5 / 99.0	**77.0** / 91.6 / 99.5

Table 3. **Results on Aachen v1.0 (top) and v1.1 (bottom) dataset [37, 50].** The **best** and second best results are highlighted.

cially for very precise pose estimation at error threshold of 5°. Enhanced by geometric information, our EIMP even gives slightly better results than SuperGlue* at error thresholds of 5° and 10°. Our EIMP outperforms previous efficient methods such as SGMNet and ClusterGNN at all error thresholds but has close even higher efficiency.

Table 2 shows results on the Scannet dataset [15]. Our IMP and EIMP also give higher accuracy than filtering-based approaches and other matchers on RootSIFT keypoints. When using SuperPoint, our IMP gives slightly better performance than SuperGlue* and SGMNet, which yield close performance to our EIMP. LMCNet [27] reports the best accuracy at error thresholds of 10° and 20° with SuperPoint because LMCNet uses SuperGlue* to provide initial matches and is further trained on the indoor dataset [47].

Qualitative comparison. As shown in Fig. 5, in the iteration process, IMP and EIMP produce more inliers and more accurate poses progressively. Besides, EIMP dynamically discards useless keypoints after each iteration. We also show the results of SuperGlue* [36] and SGMNet [12]. Due to large viewpoint changes, both SuperGlue and SGMNet give much fewer inliers from only a small area and have larger errors compared to our models. In the iterative process, inliers span almost the whole meaningful regions, resulting in more robust pose estimation.

5.2. Visual localization

As most filtering methods don't provide results on visual localization task, we only show results of OANet [49] and AdaLAM [10]. OANet and AdaLAM give close accuracy to MNN at error thresholds of $0.25m, 2°$ and $0.5m, 5°$, but much better performance at threshold of $5m, 10°$, because test images with larger viewpoint changes are more sensitive to outliers which can be partially filtered by OANet and AdaLAM. Our IMP obtains similar performance to SuperGlue* [36] and outperforms SGMNet and ClusterGNN especially for night images. Note that our models are only trained on the MegaDepth dataset [25] while SuperGlue* is additionally pretrained on Oxford and Pairs dataset [31].

EIMP slightly outperforms IMP and SuperGlue*. That is because in the long-term large-scale localization task, query and reference images usually have larger viewpoint and illumination changes with more keypoints without true matches. EIMP effectively discards these keypoints, guaranteeing the quality of matches and thus improving the localization accuracy.

5.3. Running time

As the source code of ClusterGNN [38] is not released, we mainly compare our method with SuperGlue* [36] and SGMNet [12]. The total time for each method is the summary of the time used at each iteration. IMP reduces the time by reducing the number of iterations and EIMP further decreases the time of each iteration.

Fig. 6a shows the number of iterations for relative pose estimation on YFCC100m dataset [43]. SuperGlue* and SGMNet adopt a fixed number of layers and only give matches at the last layer, so they don't report any results before the last iteration. In contrast, IMP needs only 4 iterations for 30% cases and 5 iterations for 55% cases to find a good pose (relative pose error less than 1.5), avoiding extra computation. Although EIMP loses some matches by discarding useless keypoints, it obtains close results to IMP. Fig. 6b shows the running time of SuperGlue*, SGMNet, IMP and EIMP. When using 1k and 2k keypoints, IMP and EIMP run slower than SuperGlue* because of additional in shared attention. However, when using more than 3k key-

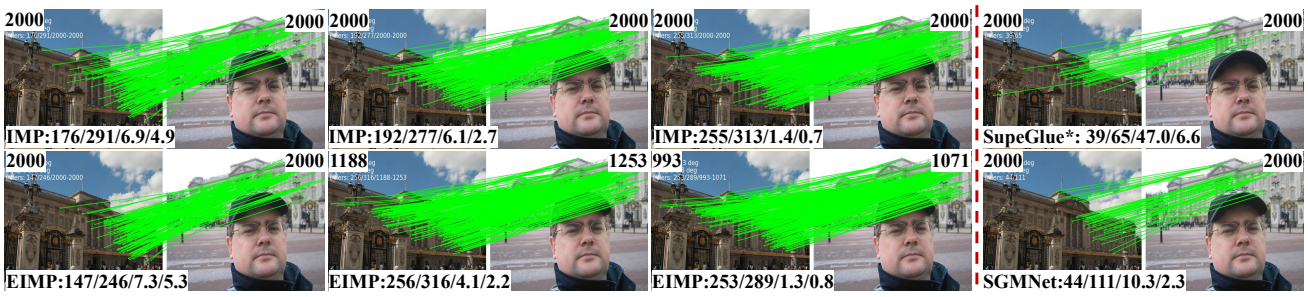
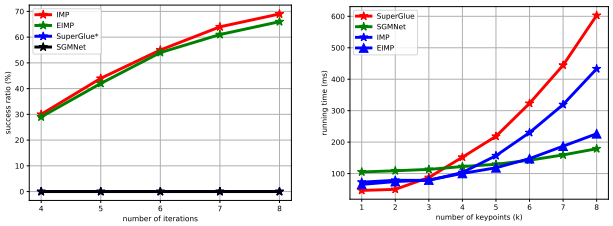


Figure 5. **Qualitative comparison.** We visualize the iterative process (left→right) of IMP (top) and EIMP (bottom). For each pair, we report the number inliers/matches and rotation/translation errors (bottom) and the number of preserved keypoints in two sets (top) at each iteration. Inliers between two images are visualized. In the iterative process, our model not only finds more inliers but enforces inliers to span wider regions. However, both SuperGlue* and SGMNet predict fewer matches from a small area, resulting higher pose errors.



(a) (b)

Figure 6. **Running time analysis.** We show (a) the number of iterations required for each pair YFCC100m dataset [43] and (b) the mean running time of 1k pairs on RTX 3090.

points, IMP runs faster because it uses fewer number of iterations (10) for sinkhorm optimization [14, 39]. Benefiting from adaptive pooling strategy, EIMP works even faster than SGMNet when using less than 6k keypoints but gives better accuracy, as demonstrated in Table 1.

5.4. Ablation study

We conduct a full ablation study for relative pose estimation on YFCC100m dataset [43] with SuperPoint [17] keypoints to verify all components in our model including the shared attention (S), pose-consistency loss (C), pose-aware iteration (P), adaptive pooling (A), and adaptive pooling with pose uncertainty (U).

Table 4 shows that shared attention (S) improves both the pose and matching accuracy by propagating more message to each keypoint. By infusing the geometric information in the iterative process, pose-consistency loss (C) and pose-aware iteration (P) also enhance the performance of IMP and EIMP. Comparisons between IMP-SC and EIMP-SCA indicate that our adaptive pooling loses 5% matches (M.S. 23.7 vs. 18.4) and gains 1% inliers (Prec. 87.7 vs. 88.7). The pose uncertainty (U) further increases the number of matches (M.S. 18.3 vs. 19.9) and inliers (Prec. 86.5 vs. 88.4) and enhances the pose accuracy about 0.7% by dynamically adjusting the sampling ratio to avoid over-pruning at each iteration. EIMP-S, which is a ratio-based sampling, gives promising pose accuracy but loses over 10% matches (M.S. 7.8 vs. 18.4) than adaptive pooling, EIMP-SA, which effectively mitigates this problem.

Model	S	C	P	A	U	@5°	@10°	20°	M.S.(%)	Prec.(%)
IMP	✗	✗	✗	✗	✗	36.9	57.1	73.7	**23.7**	85.2
IMP-S (base)	✓	✗	✗	✗	✗	38.6	58.3	74.6	23.6	87.2
IMP-SC	✓	✓	✗	✗	✗	39.1	59.0	75.0	**23.7**	**87.7**
IMP-SCP (full)	✓	✓	✓	✗	✗	**39.4**	**59.4**	**75.2**	23.0	84.9
EIMP-S	✓	✗	✗	✗	✗	35.4	55.5	72.1	7.8	**91.2**
EIMP-SA (base)	✓	✗	✗	✓	✗	36.4	56.5	73.0	18.4	88.8
EIMP-SCA	✓	✓	✗	✓	✗	36.9	56.6	73.3	18.4	88.7
EIMP-SCPA	✓	✓	✓	✓	✗	37.2	57.2	73.3	18.3	86.5
EIMP-SCPAU (full)	✓	✓	✓	✓	✓	**37.9**	**57.9**	**74.0**	**19.9**	88.4

Table 4. **Ablation study.** We test the efficacy of shared attention (S), pose-consistency loss (C), adaptive pooling (A), pose-aware iteration (P), and adaptive sampling with pose uncertainty (U) on YFCC100m dataset [43] with SuperPoint [17] features. The **best** and second best results are highlighted.

6. Conclusions

In this paper, we propose the iterative matching and pose estimation framework, allowing the two tasks to boost each other and thus improving the accuracy and efficiency. Particularly, we embed the geometric information into the matching module, enforcing the model to predict matches which are not only accurate but also able to give a good pose. Moreover, in each iteration, we utilize the predicted matches, relative pose, and attention scores to remove keypoints without potential true matches adaptively at each iteration, improving the efficiency and preserving the accuracy. Experiments demonstrate that our method achieves better performance than previous approaches on relative pose estimation and large-scale localization tasks and has high efficiency as well.

Acknowledgment

This project is supported by Toyota Motor Europe.

References

[1] Shir Amir, Yossi Gandelsman, Shai Bagon, and Tali Dekel. Deep vit features as dense visual descriptors. In *ECCV*, 2022. 2

[2] Relja Arandjelović and Andrew Zisserman. Three things everyone should know to improve object retrieval. In *CVPR*, 2012. 2, 6

[3] Gwangbin Bae, Ignas Budvytis, and Roberto Cipolla. Multi-View Depth Estimation by Fusing Single-View Depth Probability with Multi-View Geometry. In *CVPR*, 2022. 6

[4] Daniel Barath, Luca Cavalli, and Marc Pollefeys. Learning To Find Good Models in RANSAC. In *CVPR*, 2022. 2, 4

[5] Daniel Barath and Zuzana Kukelova. Relative Pose from SIFT Features. In *ECCV*, 2022. 1

[6] Daniel Barath, Jiri Matas, and Jana Noskova. MAGSAC: marginalizing sample consensus. In *CVPR*, 2019. 1

[7] Daniel Barath, Jana Noskova, Maksym Ivashechkin, and Jiri Matas. MAGSAC++, a fast, reliable and accurate robust estimator. In *CVPR*, 2020. 1

[8] Eric Brachmann and Carsten Rother. Neural-Guided RANSAC: Learning where to sample model hypotheses. In *ICCV*, 2019. 1, 3

[9] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021. 2

[10] Luca Cavalli, Viktor Larsson, Martin Ralf Oswald, Torsten Sattler, and Marc Pollefeys. Handcrafted outlier detection revisited. In *ECCV*, 2020. 1, 3, 6, 7

[11] Boyu Chen, Peixia Li, Baopu Li, Chuming Li, Lei Bai, Chen Lin, Ming Sun, Junjie Yan, and Wanli Ouyang. PSViT: Better vision transformer via token pooling and attention sharing. *arXiv preprint arXiv:2108.03428*, 2021. 2, 3

[12] Hongkai Chen, Zixin Luo, Jiahui Zhang, Lei Zhou, Xuyang Bai, Zeyu Hu, Chiew-Lan Tai, and Long Quan. Learning to match features with seeded graph matching network. In *CVPR*, 2021. 1, 2, 3, 6, 7

[13] Ondrej Chum, Tomas Werner, and Jiri Matas. Two-view geometry estimation unaffected by a dominant plane. In *CVPR*, 2005. 2, 4

[14] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *NIPS*, 2013. 3, 8

[15] Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM ToG*, 2017. 6, 7

[16] Luanyuan Dai, Yizhang Liu, Jiayi Ma, Lifang Wei, Taotao Lai, Changcai Yang, and Riqing Chen. MS2DG-Net: Progressive Correspondence Learning via Multiple Sparse Semantics Dynamic Graph. In *CVPR*, 2022. 1, 3

[17] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *CVPRW*, 2018. 2, 3, 6, 7, 8

[18] Hongyi Fan, Joe Kileel, and Benjamin Kimia. On the Instability of Relative Pose Estimation and RANSAC's Role. In *CVPR*, 2022. 2, 4

[19] Mohsen Fayyaz, Soroush Abbasi Koohpayegani, Farnoush Rezaei, and Sommerlade1 Hamed Pirsiavash2 Juergen Gall. Adaptive Token Sampling For Efficient Vision Transformers. In *ECCV*, 2022. 2

[20] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 1981. 1, 5

[21] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 4

[22] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *ICML*, 2020. 2

[23] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 6

[24] Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *ICML*, 2019. 2, 5

[25] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *CVPR*, 2018. 6, 7

[26] Philipp Lindenberger, Paul-Edouard Sarlin, Viktor Larsson, and Marc Pollefeys. Pixel-Perfect Structure-from-Motion with Featuremetric Refinement. In *ICCV*, 2021. 2, 4

[27] Yuan Liu, Lingjie Liu, Cheng Lin, Zhen Dong, and Wenping Wang. Learnable motion coherence for correspondence pruning. In *CVPR*, 2021. 1, 3, 6, 7

[28] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004. 2, 3, 6, 7

[29] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019. 6

[30] Gabriel Peyré and Marco Cuturi. Computational optimal transport. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019. 3

[31] Filip Radenović, Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondřej Chum. Revisiting oxford and paris: Large-scale image retrieval benchmarking. In *CVPR*, 2018. 6, 7

[32] Rahul Raguram, Ondrej Chum, Marc Pollefeys, Jiri Matas, and Jan-Michael Frahm. USAC: A universal framework for random sample consensus. *TPAMI*, 2012. 1

[33] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. In *NeurIPS*, 2021. 2

[34] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An efficient alternative to SIFT or SURF. In *ICCV*, 2011. 2

[35] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From Coarse to Fine: Robust Hierarchical Localization at Large Scale. In *CVPR*, 2019. 6

[36] 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, 7

[37] 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, 6, 7

[38] Yan Shi, Jun-Xiong Cai, Yoli Shavit, Tai-Jiang Mu, Wensen Feng, and Kai Zhang. ClusterGNN: Cluster-based Coarse-to-Fine Graph Neural Network for Efficient Feature Matching. In *CVPR*, 2022. 1, 2, 3, 4, 6, 7

[39] Richard Sinkhorn and Paul Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2):343–348, 1967. 3, 8

[40] Weiwei Sun, Wei Jiang, Eduard Trulls, Andrea Tagliasacchi, and Kwang Moo Yi. Acne: Attentive context normalization for robust permutation-equivariant learning. In *CVPR*, 2020. 1, 3

[41] Suwichaya Suwanwimolkul and Satoshi Komorita. Efficient linear attention for fast and accurate keypoint matching. In *ICMR*, 2022. 1, 2, 7

[42] Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. Efficient transformers: A survey. *ACM Computing Surveys*, 2020. 2, 5

[43] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. YFCC100M: The new data in multimedia research. *Communications of the ACM*, 2016. 4, 5, 6, 7, 8

[44] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, 2021. 2, 5

[45] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 1, 2

[46] Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020. 2

[47] Jianxiong Xiao, Andrew Owens, and Antonio Torralba. Sun3d: A database of big spaces reconstructed using sfm and object labels. In *ICCV*, 2013. 7

[48] Kwang Moo Yi, Eduard Trulls, Yuki Ono, Vincent Lepetit, Mathieu Salzmann, and Pascal Fua. Learning to find good correspondences. In *CVPR*, 2018. 1, 3

[49] Jiahui Zhang, Dawei Sun, Zixin Luo, Anbang Yao, Lei Zhou, Tianwei Shen, Yurong Chen, Long Quan, and Hongen Liao. Learning two-view correspondences and geometry using order-aware network. In *ICCV*, 2019. 1, 2, 3, 6, 7

[50] Zichao Zhang, Torsten Sattler, and Davide Scaramuzza. Reference pose generation for long-term visual localization via learned features and view synthesis. *IJCV*, 2021. 6, 7

[51] Chen Zhao, Yixiao Ge, Feng Zhu, Rui Zhao, Hongsheng Li, and Mathieu Salzmann. Progressive correspondence pruning by consensus learning. In *ICCV*, 2021. 3, 6, 7

[52] Zhen Zhong, Guobao Xiao, Linxin Zheng, Yan Lu, and Jiayi Ma. T-Net: Effective Permutation-Equivariant Network for Two-View Correspondence Learning. In *ICCV*, 2021. 1, 3