Adaptive Spot-Guided Transformer for Consistent Local Feature Matching

Jiahuan Yu\textsuperscript{1,*}, Jiahao Chang\textsuperscript{1,*}, Jianfeng He\textsuperscript{1}, Tianzhu Zhang\textsuperscript{1,2†}, Jiyang Yu\textsuperscript{3}, Feng Wu\textsuperscript{1}

\textsuperscript{1}University of Science and Technology of China, \textsuperscript{2}Deep Space Exploration Lab, \textsuperscript{3}China Academy of Space Technology

Abstract

Local feature matching aims at finding correspondences between a pair of images. Although current detector-free methods leverage Transformer architecture to obtain an impressive performance, few works consider maintaining local consistency. Meanwhile, most methods struggle with large scale variations. To deal with the above issues, we propose Adaptive Spot-Guided Transformer (ASTR) for local feature matching, which jointly models the local consistency and scale variations in a unified coarse-to-fine architecture. The proposed ASTR enjoys several merits. First, we design a spot-guided aggregation module to avoid interfering with irrelevant areas during feature aggregation. Second, we design an adaptive scaling module to adjust the size of grids according to the calculated depth information at fine stage. Extensive experimental results on five standard benchmarks demonstrate that our ASTR performs favorably against state-of-the-art methods. Our code will be released on https://astr2023.github.io.

1. Introduction

Local feature matching (LFM) is a fundamental task in computer vision, which aims to establish correspondence for local features across image pairs. As a basis for many 3D vision tasks, local feature matching can be applied in Structure-from-Motion (SfM) \cite{49}, 3D reconstruction \cite{13}, visual localization \cite{48, 51}, and pose estimation \cite{18, 41}. Because of its broad applications, local feature matching has attracted substantial attention and facilitated the development of many researches \cite{14, 27, 42, 44, 50}. However, finding consistent and accurate matches is still difficult due to various challenging factors such as illumination variations, scale changes, poor textures, and repetitive patterns.

To deal with the above challenges, numerous matching methods have been proposed, which can be generally categorized into two major groups, including detector-based matching methods \cite{2, 14, 15, 39, 42, 47} and detector-free matching methods \cite{9, 23, 27, 43, 44, 50}. Detector-based matching methods require to first design a keypoint detector to extract the keypoints between two images, and then establish matches between these extracted keypoints. The quality of detected keypoints will significantly affect the performance of detector-based matching methods. Therefore, many works aim to improve keypoint detection through multi-scale detection \cite{36}, repeatable and reliable...
verification [42]. Thanks to the high-quality keypoints detected, these methods can achieve satisfactory performance while maintaining high computational and memory efficiency. However, these detector-based matching methods may have difficulty in finding reliable matches in textureless areas, where keypoints are challenging to detect. Differently, detector-free matching methods do not need to detect keypoints and try to establish pixel-level matches between local features. In this way, it is possible to establish matches in the texture-less areas. Due to the power of attention in capturing long-distance dependencies, many Transformer-based methods [9, 50, 52, 57] have emerged in recent years. As a representative work, considering the computation and memory costs, LoFTR [50] applies Linear Transformer [25] to aggregate global features at the coarse stage and then crops fixed-size grids for further refinement. To alleviate the problem caused by scale changes, COTR [24] calculate the co-visible area iteratively through attention mechanism. The promising performance of Transformer-based methods proves that attention mechanism is effective on local feature matching. Nevertheless, some recent works [28, 60] indicate Transformer lacks spatial inductive bias for continuous dense prediction tasks, which may cause inconsistent local matching results.

By studying the previous matching methods, we sum up two issues that are imperative for obtaining the dense correspondence between images. (1) **How to maintain local consistency.** The correct matching result usually satisfies the local matching consistency, i.e., for two similar adjacent pixels, their matching points are also extremely close to each other. Existing methods [24, 50, 57] utilize global attention in feature aggregation, introducing many irrelevant regions that affect feature updates. Some pixels are disturbed by noisy or similar areas and aggregate information from wrong regions, leading to false matching results. As shown in Figure 1 (b), for two adjacent similar pixels, highlighted regions of global linear attention are decentralized and inconsistent with each other. The inconsistency is also present in vanilla attention (see Figure 1 (c)). Therefore, it is necessary to utilize local consistency to focus the attention area on the correct place. (2) **How to handle scale variation.** In a coarse-to-fine architecture, since the attention mechanism at the coarse stage is not sensitive to scale variations, we should focus on the fine stage. Previous methods [9, 27, 50, 57] select fixed-size grids for matching at the fine stage. However, when the scale varies too much across images, the correct match point may be out of the range of the grid, resulting in matching failure. Hence, the scheme of cropping grids should be adaptively adjusted according to scale variation across views.

To deal with the above issues, we propose a novel Adaptive Spot-guided Transformer (ASTR) for consistent local feature matching, including a spot-guided aggregation module and an adaptive scaling module. In the **spot-guided aggregation module**, towards the goal of maintaining local consistency, we design a novel attention mechanism called spot-guided attention: each point is guided by similar high-confidence points around it, focusing on a local candidate region at each layer. Here, we also adopt global features to enhance the matching ability of the network in the candidate regions. Specifically, for any point \( p \), we pick the points with high feature similarity and matching confidence in the local area. Their corresponding matching regions are used for the next attention of point \( p \). In addition, global features are applied to help the network to make judgments. The coarse feature maps are iteratively updated in the above way. With our spot-guided aggregation module, the red and green pixels are guided to the correct area, avoiding the interference of repetitive patterns (see Figure 1 (d)). In Figure 1 (e), our ASTR produces more accurate matching results, which maintains local matching consistency. In the **adaptive scaling module**, to fully account of possible scale variations, we attempt to adaptively crop different sizes of grids for alignment. In detail, we compute the corresponding depth map using the coarse matching result and leverage the depth information to crop adaptive size grids from the high-resolution feature maps for fine matching.

The contributions of our method could be summarized into three-fold: (1) We propose a novel Adaptive Spot-guided Transformer (ASTR) for local feature matching, including a spot-guided aggregation module and an adaptive scaling module. (2) We design a spot-guided aggregation module that can maintain local consistency and be unaffected by irrelevant regions while aggregating features. Our adaptive scaling module is able to leverage depth information to adaptively crop different size grids for refinement. (3) Extensive experimental results on five challenging benchmarks show that our proposed method performs favorably against state-of-the-art image matching methods.

### 2. Related Work

In this section, we briefly review several research lines that are related to sparse matching methods, dense matching methods, and vision Transformer.

**Local Feature Matching.** Local feature matching can be categorized into detector-based and detector-free methods. Detector-based methods can be divided into three stages: feature detection, feature description, and feature matching. SIFT [34] and ORB [46] are the most popular hand-crafted local features, while learning-based methods [2, 14, 15, 19, 42, 46, 63] also obtain good performance improvement compared to classical methods. There are also some works focusing on improving the feature matching stage. D2Net [15] fuses the detection and description stages. R2D2 [42] attempts to train a network to find reliable and repeatable local features. SuperGlue [47] pro-
proposes an attention-based GNN network to update extracted local features in alternating self and cross-attentions. However, detector-based methods rely on local feature extractors, which limits the performance in challenging scenarios such as repetitive textures, weak textures, and illumination changes. Unlike detector-based approaches, detector-free approaches do not require a local feature detector, but find dense feature matching between pixels directly. The classical methods [21, 35] exist, but few of them outperform detector-based methods. Learning-based methods change the game, which can be divided into cost-volume-based methods [27, 44, 53, 54] and Transformer-based methods [9, 10, 22, 24, 50, 57]. Good performance have been achieved by cost-volume-based methods, but most of them are limited by the small receptive field of CNN, which is overcome by Transformer-based methods [50]. Detector-free methods attain better performance in local feature matching, so we adopt this paradigm as the baseline.

**Vision Transformer.** Transformer [56] has been proven to be better at capturing long-range correlations than CNN in vision tasks [7, 37, 38]. Despite the great success, the computational cost of vanilla attention at high resolution is unacceptable, so some approximations [25, 33, 52, 59] have been proposed, which inevitably leads to performance degradation. Linear Attention [25] approximates softmax with ELU [11] to reduce the computational complexity to linear but degrades the focusing ability of attention. Swin-Transformer [33] limits attention in local windows, which harms the ability to establish long-range associations. At the same time, QuadTree [52] calculates attention in a coarse-to-fine manner, and ASpanFormer [9] proposes an adaptive method for selecting attention spans, but few of them consider local consistency. Different from the existing attention mechanism, we explicitly model local consistency in our spot-guided attention without introducing excessive computation and memory costs.

**Local Feature Matching with Scale Invariance.** Scale variation is one of the main challenges faced by local feature matching. Many works have explored solutions. Handcrafted local features [5, 31, 45, 46] use Gaussian pyramid model to alleviate the problem. Following the hand-crafted methods, Some learning-based descriptors [2, 4, 32, 36, 42, 63] also use the multi-scale representation. ScaleNet [3] and Scale-Net [17], instead, try to directly estimate the scale ratio. Another popular paradigm is to perform a wrap or scaling operation to eliminate the distortion caused by the scale variance. GeoWrap [6] introduces a homography regression and warps images to increase overlap area. OETR [10] limits the keypoint detection in estimated overlap areas. COTR [24] estimates scale by finding co-visible regions, and then finds correspondence by recursively zooming. However, most of above methods require significant modifications to the network architecture, and introduce additional computation overhead. Therefore, we design a fully pluggable, lightweight and training-free module for coarse-to-fine architecture.

### 3. Our Approach

In this section, we present our proposed Adaptive Spot-guided Transformer (ASTR) for Consistent Local Feature Matching. The overall architecture is illustrated in Figure 2.

#### 3.1. Overview

As shown in Figure 2, the proposed ASTR mainly consists of two modules, including a spot-guided aggregation module and an adaptive scaling module. Here we give a brief introduction to the entire process. Given an image pair $I_{Ref}$ and $I_{Src}$, to start with, we extract multi-scale feature maps of each image through a shared Feature Pyramid Network (FPN) [30]. We denote feature maps with the size of $1/i$ as $F^{1/i}_i$. Then, $F^{1/32}$ and $F^{1/8}$ are fed into the spot-guided aggregation module for coarse-matching and depth maps. Here, the coarse matching result is acquired in three phases. First, we need to compute the similarity matrix, which can be given by $S(i, j) = \tau(F^{1/8}_{ref}(i), F^{1/8}_{src}(j))$ with flattened features, where $\tau$ is the temperature coefficient.

Then we perform dual-softmax operator on $S$ to calculate matching matrix $P_c$:

$$P_c(i, j) = \text{softmax}(S(i, :))(i, j) \cdot \text{softmax}(S(:, j))(i, j).$$

(1)

Finally, we use the mutual nearest neighbor strategy and the threshold $\theta_c$ to filter out the coarse-matching result $M_c$. According to depth information and coarse-matching result, we can crop different size grids on the high-resolution feature map $F^{1/2}$. After linear self and cross attention layers, features of the cropped grids are used to produce the final fine-level matching result.

#### 3.2. Spot-Guided Aggregation Module

Correct matching always satisfies the local matching consistency, i.e., the matching points of two similar adjacent pixels are also close to each other in the other image. When humans establish dense matches between two images, they will first scan through the two images quickly and keep in mind some landmarks that are easier to match correctly. For those trouble points similar to surrounding landmarks, it is not easy to obtain correct matches in the beginning. However, now, they can focus attention around the matching points of landmarks to revisit trouble points’ matches. In this way, more correctly matched landmarks are obtained. After several iterations of the above process, eventually, they will get the matching result for the whole image. Inspired by this idea, we design a spot-guided aggregation module. Section 3.2.1 introduces the preliminaries of vanilla attention and linear attention. Section 3.2.2 describes our spot-guided attention mechanism. Section 3.2.3 demonstrates the design of the entire spot-guided aggregation module.
3.2.1 Preliminaries

The calculation of vanilla attention requires three inputs: query $Q$, key $K$, and value $V$. The output of vanilla attention is a weighted sum of the value, where the weight matrix is determined by the query and its corresponding key. The process can be described as

$$\text{Attention}(Q, K, V) = \text{softmax}(QK^T)V.$$  \hfill (2)

However, in vision tasks, the size of the weight matrix $\text{softmax}(QK^T)$ increases quadratically as the image resolution grows. When the image resolution is large, the memory and computational cost of vanilla attention is unacceptably large. To solve this problem, Linear attention [25] is proposed to replace the softmax operator with the product of two kernel functions:

$$\text{Linear attention}(Q, K, V) = \phi(Q)(\phi(K^T)V),$$  \hfill (3)

where $\phi(\cdot) = \text{elu}(\cdot) + 1$. Since the number of feature channels is much smaller than the number of pixels, the computational complexity is reduced from quadratic to linear.

3.2.2 Spot-Guided Attention

It is known from the local matching consistency that the matching points of similar adjacent pixels are also close to each other. In Figure 3, we illustrate the case that the reference image as query aggregates features from the source image. Given reference and source feature maps $F^{1/8} = \{F_{\text{ref}}^{1/8}, F_{\text{src}}^{1/8}\}$, we compute a matching matrix $P$ across images. For any pixel $p$ in Figure 3, we first compute the similarity score $S_{\text{sim}}(p) \in \mathbb{R}^{l^2-1}$ between $p$ and other pixels in the $l \times l$ area around $p$. Specifically, the similarity score can be obtained as

$$S_{\text{sim}}(p) = \text{softmax} \left( \{ \langle F_{\text{ref}}^{1/8}(p), F_{\text{src}}^{1/8}(p_i) \rangle \}_{p_i \in N(p)} \right),$$  \hfill (4)

where $\langle \cdot, \cdot \rangle$ is the inner product, and $N(p)$ is the set of pixels in the $l \times l$ field around pixel $p$. In addition, we should also consider the reliability of points in $N(p)$. For each $p_i \in N(p)$, confidence can be viewed as the highest similarity to all pixels on the source images. Meanwhile, we can also get the matching point position of $p_i$, denoted as $\text{Loc}(p_i)$. Hence, $\text{Loc}(p_i)$ and confidence score $S_{\text{conf}}(p) \in \mathbb{R}^{l^2-1}$ can

---

Figure 2. The architecture of ASTR. Our ASTR consists of two major components: spot-guided aggregation module and the adaptive scaling module. “Cross Attention” means vanilla cross attention, unless otherwise stated. Please refer to the text for detailed architecture.

Figure 3. The illustration of our spot-guided attention.
be computed in the following way:
\[
S_{\text{conf}}(p) = \max \{P_x(p_i, \cdot)\}_{p_i \in N(p)},
\]
\[
\text{Loc}(p_i) = \arg \max \{P_x(p_i, \cdot)\},
\]
(5)

Combining two scores, we select \(p\) and top-k points 
Topk\((p)\) whose matching points are used as seed points \(\text{Seed}(p)\):
\[
\text{Topk}(p) = \{p\} \cup \text{topk}\{S_{\text{sim}}(p) \cdot S_{\text{conf}}(p)\},
\]
\[
\text{Seed}(p) = \{\text{Loc}(q)\}_{q \in \text{Topk}(p)}.
\]
(6)

Following that, we extend \(l \times l\) regions centered on these seed points \(\text{Seed}(p)\) on \(I_{\text{src}}\), which are the spot areas of \(p\). Finally, cross attention is performed between \(p\) and corresponding spot areas. After exchanging the source image and the reference image, the source feature map is updated in the same way.

### 3.2.3 Spot-Guided Feature Aggregation

For the input features \(F^{1/32}\) and \(F^{1/8}\), \(F^{1/32}\) is updated by vanilla cross attention, and \(F^{1/8}\) is updated by linear cross attention for initialization. Then, two features of different resolutions are fed into the spot-guided aggregation blocks. In each block, \(F^{1/32}\) and \(F^{1/8}\) are first fused into each other in the following way:
\[
\hat{F}^{1/32} = F^{1/32} + \text{Conv}_{1 \times 1}(\text{Down}(F^{1/8})),
\]
\[
\hat{F}^{1/8} = F^{1/8} + \text{Conv}_{1 \times 1}(\text{Up}(F^{1/32})),
\]
(7)

where \(\hat{F}^{1/32}\) and \(\hat{F}^{1/8}\) are features after fusion. \(\text{Down}(\cdot)\) and \(\text{Up}(\cdot)\) are downsampling and upsampling. And then, \(\hat{F}^{1/32}\) aggregate features across images by vanilla attention. In the meantime, \(\hat{F}^{1/8}\) aggregate features across images by spot-guided attention. After four spot-guided aggregation blocks, \(1/32\)-resolution features are fused into \(1/8\)-resolution features, which are used to obtain the coarse-matching result \(M_c\).

### 3.3. Adaptive Scaling Module

At the fine stage, previous methods usually crop fixed-size grids based on the coarse matching result. When there is a large scale variation, fine matching may fail since the ground-truth matching points are out of grids. Thus, we refer to depth information to adaptively crop grids of different sizes between images. Section 3.3.1 describes the way to obtain depth information from the coarse-matching result. Section 3.3.2 demonstrates the process of adaptively cropping grids.

#### 3.3.1 Depth Information

With the coarse-level matching result, we can obtain the relative pose of two images \(\{R, T\}\) through RANSAC [16]. It should be noted that the \(T\) calculated here has a scale uncertainty, i.e., \(T_{\text{real}} = \alpha T\), where \(\alpha\) is the scale factor. Given the image coordinates of any pair of matching points \(\{x_i, x_j\}\) from coarse-level matching result, they satisfy the following equation:
\[
d_jK_j^{-1}(x_j, 1)^T = d_iRK_i^{-1}(x_i, 1)^T + \alpha T,
\]
(8)

where \(d_i\) and \(d_j\) are the depth values of \(x_i\) and \(x_j\). \(K_i\) and \(K_j\) are corresponding camera intrinsics. We let \(p_i = RK_i^{-1}(x_i, 1)^T\) and \(p_j = K_j^{-1}(x_j, 1)^T\). From Equation (8) it can be deduced that:
\[
d_j\alpha p_j = d_i\alpha p_i + \alpha T,
\]
\[
\Rightarrow \begin{cases} (d_j/\alpha)p_j \wedge p_i = 0 + T \wedge p_i, \\ 0 = (d_i/\alpha)p_i \wedge p_j + T \wedge p_j, \end{cases}
\]
\[
\Rightarrow \begin{cases} d_j/\alpha = \text{mean}(\text{div}(T \wedge p_i, p_j \wedge p_i)), \\ d_i/\alpha = \text{mean}(\text{div}(-T \wedge p_j, p_i \wedge p_j)), \end{cases}
\]
(9)

where \(\wedge\) indicates outer product. \(\text{div}(\cdot, \cdot)\) denotes element-wise division between two vectors. \(\text{mean}(\cdot)\) is the scalar mean of each component of a vector. In this way, we have obtained depth information of \(x_i\) and \(x_j\) with scale uncertainty.

#### 3.3.2 Adaptive Scaling Strategy

As shown in Figure 4, \(x_i\) and \(x_j\) are a pair of matching points at the coarse stage. \(d_i\) and \(d_j\) are depth values of \(x_i\) and \(x_j\). To begin with, we crop a \(s_i \times s_i\) region centered on \(x_i\). When the scale changes too much, the correct matching point \(\hat{x}_j\) may be beyond the \(s_i \times s_i\) region around \(x_j\). Because everything looks small in the distance and big on the contrary, the size of cropped grid \(s_j\) should satisfy:
\[
\frac{s_j}{s_i} = \frac{d_i}{d_j} = (\frac{d_i}{\alpha})(\frac{d_j}{\alpha})^{-1},
\]
(10)

Following the above approach, we can crop different sizes of grids adaptively according to the scale variation. After the same refinement as LoFTR [50], we get the final matching position \(\hat{x}_j\) of \(x_i\).
4.2. Homography Estimation

More details about the operator on CUDA. Please refer to the Supplementary Material for a highly optimized general sparse attention operator based to establish spot-guided attention efficiently, we implement optimizer. The initial learning rate is $\frac{1}{18}$ [20] with features at 15 epochs with a batch size of 8 by Adam [26].

For training, we input images with the size of $832 \times 1024$. At the fine stage, window size $s$ is adapted to $5$, and window size $j_k$ in the reference image is $3$ pixels, $3.5\%$, and $4.4\%$ error thresholds, which strongly proves the effectiveness of our method. ASTR outperforms the best method before (PDC-net+ [54]), achieving a large margin of $4.4\%$ under 3 pixels, $3.5\%$ under 5 pixels, and $2.5\%$ under 10 pixels. Thanks to the proposed spot-guided aggregation module and adaptive scaling module, our method can yield more accurate matches under extreme viewpoint and illumination variations.

4.3. Relative Pose Estimation

Dataset and Metric. We use MegaDepth [29] and ScanNet [12] to demonstrate the performance of our ASTR in relative pose estimation. MegaDepth [29] is a large-scale outdoor dataset that contains 1 million internet images of 196 different outdoor scenes. Each scene is reconstructed by COLMAP [49]. Depth maps as intermediate results can be converted to ground truth matches. We sample the same 1500 pairs as [50] for testing. All test images are resized such that their longer dimensions are 1216. ScanNet [12] is usually used to validate the performance of indoor pose estimation. It is composed of monocular sequences with ground truth poses and depth maps. Wide baselines and extensive textureless regions in image pairs make ScanNet [12] challenging. For a fair comparison, we follow the same testing pairs and evaluation protocol as [50].

In Table 1, we can see that our ASTR achieves new state-of-the-art performance on HPatches [1] under all error thresholds, which strongly proves the effectiveness of our method. ASTR outperforms the best method before (PDC-net+ [54]), achieving a large margin of $4.4\%$ under 3 pixels, $3.5\%$ under 5 pixels, and $2.5\%$ under 10 pixels. Thanks to the proposed spot-guided aggregation module and adaptive scaling module, our method can yield more accurate matches under extreme viewpoint and illumination variations.

### Table 1. Evaluation on HPatches [1] for homography estimation.

<table>
<thead>
<tr>
<th>Category</th>
<th>Method</th>
<th>Homography est.</th>
<th>AUC @1px</th>
<th>AUC @3px</th>
<th>AUC @10px</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detector-based</td>
<td>D2Net [15]+NN</td>
<td>23.2</td>
<td>35.9</td>
<td>53.6</td>
<td>0.2K</td>
</tr>
<tr>
<td></td>
<td>R2D2 [42]+NN</td>
<td>50.6</td>
<td>63.9</td>
<td>76.8</td>
<td>0.5K</td>
</tr>
<tr>
<td>Detector-free</td>
<td>DISK [55]+NN</td>
<td>52.3</td>
<td>64.9</td>
<td>78.9</td>
<td>1.1K</td>
</tr>
<tr>
<td></td>
<td>SP [14]+SuperGlue [47]</td>
<td>53.9</td>
<td>68.3</td>
<td>81.7</td>
<td>0.6K</td>
</tr>
<tr>
<td></td>
<td>Patch2Pix [64]</td>
<td>46.4</td>
<td>59.2</td>
<td>73.1</td>
<td>1.0K</td>
</tr>
<tr>
<td></td>
<td>ASTR(ours)</td>
<td>71.7</td>
<td>80.3</td>
<td>88.0</td>
<td>1.0K</td>
</tr>
</tbody>
</table>

Results. In Table 2, our ASTR outperforms other state-of-the-art methods on MegaDepth [29]. In par-
4.4. Visual Localization

It is particularly notable, our ASTR improves by 3.1% in AUC@5° and 1.6% in AUC@0.1°. Table 3 summarizes the performance comparison between the proposed ASTR and state-of-the-art methods on ScanNet [12]. Our ASTR ranks first when only considering models trained on ScanNet [12] indicating the impressive generalization of our method. Thanks to the proposed spot-guided aggregation module and adaptive scaling module, our method can yield more correct matches, resulting in more accurate pose estimation. In order to further demonstrate the effectiveness of our ASTR, in Figure 5, we visually demonstrate the comparison with other methods on the matching result. Notably, our methods can better handle the challenges such as textureless areas, repetitive patterns, and scale variations.

4.4. Visual Localization

Dataset and Metric. In this experiment, InLoc [51] and Aachen Day-Night v1.1 [62] are used to verify the ability of our ASTR in visual localization. InLoc [51] is an indoor dataset with 9972 RGBD images, of which 329 RGB images are employed as queries for visual localization. The challenge of InLoc [51] mainly comes from textureless regions and repetitive patterns under large viewpoint changes. In Aachen Day-Night v1.1 [62], 824 day-time images and 191 night-time images are chosen as queries for outdoor visual localization. Large illumination and viewpoint changes pose challenges for Aachen [62]. For both benchmarks, we evaluate the performance of our ASTR trained on MegaDepth [29] in the same way as [50]. The metrics of InLoc [51] and Aachen [62] are the same, which measure the percentage of images registered within given error thresholds.

Results. For InLoc [51] benchmark, our method achieves the best performance on DUC1 and is on par with state-of-the-art methods on DUC2 (in Table 4). For Aachen [62] benchmark, our ASTR performs comparative with others on Day and Night scenes (in Table 5). Overall, our method exhibits strong generalization ability in visual localization.

4.5. Ablation Study

To deeply analyze the proposed method, we perform detailed ablation studies on MegaDepth [29] to evaluate the effectiveness of each component in ASTR. Here, we use images with a size of 544 for training and evaluation. As shown in Table 6, we intend to gradually add these components to the baseline. The baseline (Index-1) we used is

Table 2. Evaluation on MegaDepth [29] for outdoor relative position estimation.

<table>
<thead>
<tr>
<th>Category</th>
<th>Method</th>
<th>Pose estimation AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detector-based</td>
<td>SP [14]+SuperGlue [47]</td>
<td>42.2 59.0 73.6</td>
</tr>
<tr>
<td>Detector-free</td>
<td>DRC-Net [27]</td>
<td>27.0 42.9 58.3</td>
</tr>
<tr>
<td></td>
<td>MatchFormer [57]</td>
<td>53.3 69.7 81.8</td>
</tr>
<tr>
<td></td>
<td>QuadTree [52]</td>
<td>54.6 70.5 82.2</td>
</tr>
<tr>
<td></td>
<td>ASTR(ours)</td>
<td>58.4 73.1 83.8</td>
</tr>
</tbody>
</table>

Table 3. Evaluation on ScanNet [12] for indoor relative position estimation. * indicates models trained on MegaDepth [29].

<table>
<thead>
<tr>
<th>Category</th>
<th>Method</th>
<th>Pose estimation AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detector-based</td>
<td>D2-Net [15]+NN</td>
<td>5.3 14.5 28.0</td>
</tr>
<tr>
<td>Detector-free</td>
<td>MatchFormer [57]*</td>
<td>7.7 17.9 30.5</td>
</tr>
<tr>
<td></td>
<td>QuadTree [52]*</td>
<td>19.0 37.3 53.5</td>
</tr>
<tr>
<td></td>
<td>ASTR(ours)*</td>
<td>19.4 37.6 54.4</td>
</tr>
</tbody>
</table>


<table>
<thead>
<tr>
<th>Method</th>
<th>DUC1</th>
<th>DUC2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(0.25m, 5°)</td>
<td>(0.6m, 0.1°)</td>
</tr>
<tr>
<td>Patch2Pix [69]</td>
<td>SP [14]+CAPS [58]</td>
<td>42.4/62.6/76.3</td>
</tr>
<tr>
<td>LoFTR [50]</td>
<td>47.5/72.2/84.8</td>
<td>54.2/74.8/85.5</td>
</tr>
<tr>
<td>MatchFormer [57]</td>
<td>46.5/73.2/85.9</td>
<td>55.7/71.8/81.7</td>
</tr>
<tr>
<td>ASpanFormer [9]</td>
<td>51.5/73.7/86.4</td>
<td>55.0/74.8/81.7</td>
</tr>
<tr>
<td>ASTR(ours)</td>
<td>53.0/73.7/84.7</td>
<td>52.7/76.3/84.0</td>
</tr>
</tbody>
</table>

Table 5. Visual localization evaluation on the Aachen Day-Night benchmark v1.1 [62].

<table>
<thead>
<tr>
<th>Method</th>
<th>Day</th>
<th>Night</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(0.25m, 5°)</td>
<td>(0.6m, 0.1°)</td>
</tr>
<tr>
<td></td>
<td>(0.25m, 5°)</td>
<td>(0.6m, 0.1°)</td>
</tr>
<tr>
<td></td>
<td>(0.25m, 5°)</td>
<td>(0.6m, 0.1°)</td>
</tr>
<tr>
<td></td>
<td>(0.25m, 5°)</td>
<td>(0.6m, 0.1°)</td>
</tr>
<tr>
<td></td>
<td>(0.25m, 5°)</td>
<td>(0.6m, 0.1°)</td>
</tr>
</tbody>
</table>

Table 6. Ablation Study of each component on MegaDepth [29].

<table>
<thead>
<tr>
<th>Index</th>
<th>Multi-Level</th>
<th>Spot-Guided</th>
<th>Scaling</th>
<th>Pose estimation AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>45.6 62.2 75.3</td>
</tr>
<tr>
<td>2</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>46.7 63.1 76.3</td>
</tr>
<tr>
<td>3</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>47.7 64.5 77.4</td>
</tr>
<tr>
<td>4</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>48.3 65.0 77.7</td>
</tr>
</tbody>
</table>
slightly different from LoFTR [50]. More details can be found in Supplementary Material.

**Effectiveness of Spot-Guided Aggregation Module.** We divide the spot-guided aggregation module into multi-level cross attention and spot-guided attention for ablation studies. We first add vanilla cross attention layers at 1/32 resolution to the baseline (Index-2 in Table 6). Comparing the results of Index-2 and Index-1, we conclude that 1/32 resolution global interaction across images is beneficial for image matching. Then, in Index-3, linear attention layers at 1/8 resolution are substituted for the spot-guided attention layers. The performance of Index-3 is improved compared with Index-2, which verifies the effectiveness of our spot-guided attention. In Figure 6, we visualize vanilla and our spot-guided cross attention maps for contrast, showing that spot-guided attention can indeed avoid interference from unrelated areas.

To maximize the effectiveness of our spot-guided attention, we explore how to set suitable parameters $l$ and $k$. First, in the setting of Index-3, we fix $l = 5$ and vary $k$ from 1 to 6. After observing the results in Table 7, the performance drops when $k$ is smaller than 4 or larger than 4.

Then, we fix $k = 4$ and vary $l$ from 3 to 9. As shown in Table 7, we find that the model achieves the best performance at $l = 5$. The reason may be that the spot area is too small to provide sufficient information from another image when using small $k$ or $l$. With large $k$ or $l$, for a certain pixel, some matching areas of low confidence or dissimilar points will damage its feature aggregation.

**Effectiveness of Adaptive Scaling Module.** As shown in Table 6, comparing the results of Index-4 and Index-3, we can see that the performance is improved, which indicates that coarse-level matching results are better refined with adaptive scaling module. In Figure 7, we visualize the cropped grids from adaptive scaling module, indicating that our adaptive scaling module can adaptively crop grids of different sizes according to scale variations.

5. Conclusion

In this paper, we propose a novel Adaptive Spot-guided Transformer (ASTR) for consistent local feature matching. To model local matching consistency, we design a spot-guided aggregation module to make most pixels avoid the impact of irrelevant areas, such as noisy and repetitive regions. To better handle large scale variation, we use the calculated depth information to adaptively adjust the size of grids at the fine stage. Extensive experimental results on five benchmarks demonstrate the effectiveness of the proposed method.

**Limitation.** Although our adaptive scaling module is lightweight and pluggable, it demands camera pose estimation in the coarse stage, which requires the camera intrinsic parameters. While camera intrinsic parameters are obtainable in standard datasets and most real-world scenarios, there are still some images from wild that lack them, rendering the adaptive scaling module disabled in such cases.

**Acknowledgement.** This work was partially supported by the National Nature Science Foundation of China (No. 62022078 and No. 62021001).
References


Jerome Revaud, Philippe Weinzaepfel, César Roberto de Souza, and Martin Humenberger. R2D2: repeatable and reliable detector and descriptor. In Advances in Neural Information Processing Systems, 2019. 1, 2, 3, 6, 7


Prune Truong, Martin Danelljan, and Radu Timofte. Glunet: Global-local universal network for dense flow and


Meng Meng, Tianzhu Zhang, Zhe Zhang, Yongdong Zhang, and Feng Wu. Task-aware weakly supervised object localization with transformer. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022. 3


Jerome Revaud, Philippe Weinzaepfel, César Roberto de Souza, and Martin Humenberger. R2D2: repeatable and reliable detector and descriptor. In Advances in Neural Information Processing Systems, 2019. 1, 2, 3, 6, 7


Prune Truong, Martin Danelljan, and Radu Timofte. Glunet: Global-local universal network for dense flow and


