MESA: Matching Everything by Segmenting Anything

Supplementary Material

In the following, we provide additional details about the proposed area matching method, MESA. Sec. 7 describes specific implementation details of our method. Sec. 8 analyzes the computation complexities of main components in MESA. Sec. 9 gives more insights and ablation studies, that motivates our design. Sec. 10 provides visualizations of area matching and point matching results. Sec. 11 states the limitation of MESA and our future work.

7. Implementation Details

In this section, we provide sufficient details about the implementation of MESA for reproduction, including detailed operations in Graph Completion (Sec. 7.1), training details about our Learning Area Similarity (Sec. 7.2) and complete parameter settings in Graphical Area Matching (Sec. 7.3).

7.1. Graph Completion

The detailed algorithm for our graph completion is depicted in Algorithm 1, which takes initial AG ($\mathcal{G}_{ini}$) as input and outputs the final AG ($\mathcal{G}$) with scale hierarchy. Additionally, we describe the area cluster and two main area operations in the algorithm as follows.

Area Cluster. For orphan nodes in each level, we cluster them based on their area centers to decide which operation will be performed on them. We use the k-means algorithm with elbow method [41] to determine the cluster number. The candidate cluster number is set as $\{1, \dots, n\}$, where n is the number of orphan nodes in the current level. This algorithm is fed with area centers and outputs labeled ones.

Area Fusion and Expansion. Area fusion and expansion are key operations in our graph completion algorithm. Specifically, area fusion is to find the largest outer rectangle of the two areas as the new area, as depicted in Fig. 6 (a). Due to the careful threshold settings of our area level, the fused area size will exceed current level and be awaited for subsequent operations. On the other hand, the expansion operation is to expand the area to the next level size (Fig. 6 (b)). In particular, suppose the lower bound of size for the next level is s^2, if both of the area width and height are smaller than s, we expand the height and width of the area to s, keeping the area center fixed. Otherwise if area width $w \geq s$, we let the area height $h = s^2/w$, keeping the area center fixed, and vice versa. The above operations are performed when the expanded area is inside the image. On the other hand, if the expanded area is outside the image, the area center will be moved as shown in Fig. 6 (b).

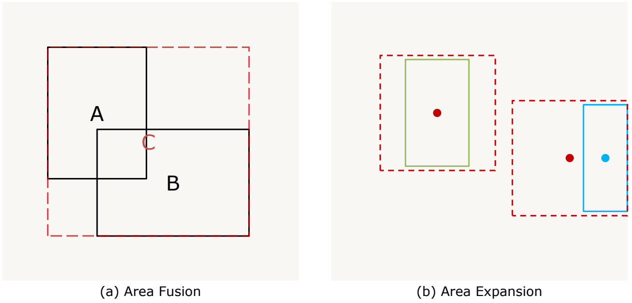

Figure 6. **The Area Fusion and Area Expansion. (a)** Area fusion is to achieve the smallest area (C) containing the input areas (A and B). **(b)** Generally, area expansion is to fix the original area center and expand its size to the smallest size of the next level. When the original area is too close to the image boundary, we will move the area center to keep the expanded area inside the image.

7.2. Learning Area Similarity.

We propose the learning area similarity model for area similarity calculation, which is the basis of our graphical area matching. In this section, we describe the training protocol of this model, including supervision and training details.

Supervision. We generate regular area images from both indoor and outdoor datasets [11, 27] as training data through the proposed method in Sec. 3. Then the area pairs with more than 30% overlap are collected. For each image patch p_i in these pairs, its ground truth activity σ_i^{gt} is set as 1, if more than 60% pixels in it have correspondences in the other area image, and 0 otherwise. As our classification formulation, we use the binary cross entropy (BCE) of each patch classification to form the loss function of area similarity calculation (L_{asc}).

$$L_{asc} = \frac{1}{Z} \sum_{i}^{Z} BCE(\sigma_i^{gt}, \sigma_i) \tag{16}$$

Based on this loss, our network can learn to achieve the similarity between two area images.

Training Details. Following previous point matching work [40], the proposed learning similarity module is trained utilizing MegaDepth [27] and ScanNet [11] dataset. For each image pair sampled in the datasets in every epoch following [40], we collect up to 5 area image pairs for training. Fortunately, there is no need to train this model from scratch, thanks to the similar objective between the coarse point matching and the area similarity calculation, both of which aims at patch-level similarities. Therefore, we adopt the pretrained weights of coarse level feature operation in ASpan [8] on our network. With the modified output head,

Algorithm 1: Graph Completion

```
Input: $\mathcal{G}_{ini} = \langle \mathcal{V}_{ini}, \mathcal{E}_{ini} \rangle$
Output: $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$
for $l$ in $[0, L-1]$ do
    initial orphan node set $\mathcal{O} = \varnothing$;
    for $v_i \in \{v_i | l_{a_i} = l\}$ do
        if $v_i$ has no parent then
            add $v_i$ into $\mathcal{O}$;
    cluster the nodes in $\mathcal{O}$ based on their area
      centers;
    for each node cluster $\mathcal{C}_h = \{v_k\}_{k=0}^{C}$ do
        if $C \geq 2$ then
            for each $v_k \in \mathcal{C}_h$ do
                if $v_k$ has not been fused then
                    fuse area $a_k$ with its nearest
                      neighbor $a^n | v^n \in \mathcal{C}_h$:
                      $a^f = F(a_k, a^n)$;
                    generate higher level node $v^f$ for
                      $a^f$;
                    add $v^f$ into $\mathcal{V}_{ini}$;
                    form edges by Link Prediction:
                      $\{e_h\}_h = LP(v^f, \mathcal{V}_{ini})$;
                    add $\{e_h\}_h$ into $\mathcal{E}_{ini}$;
        else
            Update the single node $v_0$:
              $v_0^u = Up(v_0)$;
            construct edges: $\{e_j\}_j = LP(v_0^u, \mathcal{V}_{ini})$;
            add $\{e_j\}_j$ into $\mathcal{E}_{ini}$;
$\mathcal{E} = \mathcal{E}_{ini}$ ;
$\mathcal{V} = \mathcal{V}_{ini}$ ;
output the updated AG: $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$;
```

our network is fine-tuned using AdamW [30] on 2 NVIDIA RTX 4090 GPUs. We use the same learning rate and batch size settings as ASpan and LoFTR [40] during training. This model can converge within 3 epochs.

7.3. Graphical Area Matching

In this section, we describe some implementation details of our graphical area matching. First, we illustrate the complete forward process of area matching. Then, we specific the parameter settings we recommend and utilized in our experiments. Next, the details about area image cropping is presented, which is cropping area images from original images to serve as input images for the combined point matcher. The post-processing of point matching within areas is also illustrated.

Area Matching Process. Given the image pair (I_0, I_1) and their area graphs $(\mathcal{G}^0, \mathcal{G}^1)$, we first collect area nodes with specific size level l_{a^*} from $\mathcal{G}^0$ as the *source nodes*, which have proper sizes for subsequent point matching. Then, area matches are found for these nodes from $\mathcal{G}^1$ by our method. Afterwards, we exchange the two images to repeat the above operations. The final area matches are the common matching results.

Parameter Settings. We describe the common parameters for different scenes here. During the AG construction, the input image are resized to 640×480. The aspect ratio threshold $T_r = 4$ and minimal size threshold is $T_s = 80^2$. The number of area size threshold is 4 and specific TL_is are $80^2, 130^2, 256^2, 390^2, 560^2$. The δ_l is 0.1 and δ_h is 0.8. In graphical area matching, the λ in Eq. (5) is 0.1. The area similarity threshold $T_{as} = 0.05$. The energy balance weights $(\mu, \alpha, \beta, \gamma)$ in Eq. (11) are $4, 2, 2, 2$. The specific area level l_{a^*} for point matching is 1. The T_{Er} in Eq. (15) is 0.1. Other parameters are specified for different scenes as described in our paper.

Area Image Cropping and Point Matching. Following the A2PM framework [52], the next step after area matching is to perform point matching inside area matches. Thus, these area matches need to be achieved through image cropping. At the same time, one of the benefits of A2PM is the input image with high resolution that can be provided to the point matcher. Therefore, the cropping is performed on original images with the highest resolution. A straightforward cropping approach is to crop areas along the exact bounding box and then resize them to default input size of point matchers. However, as area sizes can be quite different from default input sizes of point matchers, direct cropping and resize introduce severe distortion for point matching, resulting in decreased matching precision (cf. Sec. 9.1). In order to address this issue, we propose to crop area image by considering the aspect ratio. To be specific, we force the cropped area image to possess the same aspect ratio as the input size of the point matcher, while trying to keep the area center unchanged. If the area respect ratio (W_a/H_a) is larger than the input aspect ratio (W_i/H_i), we fix the width W_A of area image and expand the height H_a to $(W_a \times H_i)/W_i$. Otherwise, we fix the height and expand the width to $(H_a \times W_i)/H_i$. Moreover, as solid feature points often cluster in the boundaries of objects, we set a spread ratio $(r_s = 1.2)$ to slightly increase the cropping size, allowing for more precise correspondences on the boundaries. Finally, if the cropped area exceeds the original image, we will move its center to keep it inside the image, similar to our Area Expansion in Fig. 6.

For point matching inside area matches, in practice, we empirically set the input size of point matchers to a square, *i.e.* the input aspect ratio is 1, as it can lead to statistically the smallest area size adjustment. However, as the learning models are sensitive to training size and scale (the area input has different scale with original image), especially the Transformer [13], the performance of baselines are decreased, as we shown in Sec. 9.1. Therefore, we fine-tune the baselines under the square sizes with ground truth area matches generated by our method, *e.g.* 640×640 area matches in indoor scenes, to achieve similar accuracy with original models, before combined with MESA.

Post-processing. In SGAM [52], *Global Match Collection* (GMC) is proposed to collect precise point matches through globally entire-image matching, which significantly improve the matching precision [52]. Therefore, we adopt this module in our method as well and set the occupancy ratio as 0.6. However, as SAM enables areas throughout the image, MESA usually achieves enough area matches to cover the whole overlap between images. Thus, this module is only activated in few cases, but also helpful to precise matching. As we mentioned before, we also adopt GAM [52] as the post-processing of area matching.

8. Computation Complexity

Here, we analyze the computation complexity of proposed graphical area matching.

8.1. Area Similarity Calculation

Firstly, area similarity calculation is performed to achieve the required node energies in the graph, serving as the prerequisite of our graphical area matching. Suppose we have two AGs, $\mathcal{G}^0$ and $\mathcal{G}^1$, for the input image pair, $\mathcal{G}^0$ gets N nodes ($|\mathcal{V}^0| = N$) and $\mathcal{G}^1$ gets M nodes ($|\mathcal{V}^1| = M$). Therefore, the dense graph energy calculation needs $M \times N$ times similarity calculation. However, owing to the similarity conditional independence of ABN (Sec. 4.3), the actual number ($M' \times N'$) of similarity calculation is smaller than $M \times N$, as $N' < N$. Nevertheless, directly setting children pair similarities as 0 is too rough (Eq. (10)), as large scale differences also leads to near-zero similarity between areas. In practise, we only set the related similarities of *next level children* as 0 for area matching accuracy and the efficiency from ABN is still helpful to our approach. Moreover, we only care about the similarities between source nodes in $\mathcal{G}^0$ and other nodes in $\mathcal{G}^1$, because we collect source nodes with specific level from $\mathcal{G}^0$ to match, *e.g.*, usually $3 \sim 4$ areas in indoor scene and less in outdoor scene. Therefore, we have $M' < M$. Similarly, in the case of duality, *i.e.*, collecting source nodes from $\mathcal{G}^1$ to match, we only need to perform a few supplementary calculations, as similarities are symmetric and reusable. Thus, the real computation complexity of area similarity computation is $O(M' \times N')$, where $M' \times N' < M \times N$.

8.2. Edge Energy Calculation

Except the node energy calculation, the edge energy is also needed to be determined for *Graph Cut*. The computation complexity of edge energy calculation is related to edge number of $\mathcal{G}^0$ and $\mathcal{G}^1$. Assume $|\mathcal{E}_0| = E$ and $|\mathcal{E}_1| = K$, the specific computation complexity is $O(E + K)$.

8.3. Global Energy Minimization

In our global energy minimization for area matching refinement, the matching energy of parent, children and neighbour pairs all need to be calculated. Taking parent matching energy for example, we derive its computation complexity as follows. Suppose n nodes are achieved as match candidates through *Graph Cut* and each node gets $Q_i,\ i \in (0, n]$ parent nodes, there are $Q_i \times V$ node similarities need to be accessed (as the similarity calculation is finished), where V is the parent node number of the source node. Hence, the total computation complexity for parent matching energy in global energy minimization is $O(\sum_i^n Q_i \times V)$. The children matching energy and neighbour matching energy are similar. As n is the number of node after *Graph Cut*, it is small in most cases, *e.g.*, usually < 3 area nodes. Moreover, the number of parent nodes (or children, neighbour nodes) is also limited. Therefore, the computation complexity for global energy minimization is acceptable in practise.

9. Additional Ablation Study

In this section, we examine the performance impact of more components in MESA, including the input image size (Sec. 9.1), image cropping approach (Sec. 9.2) and energy parameter setting (Sec. 9.3).

9.1. Ablation Study on Input Image Size

Input image size is a sensitive parameter for feature matching, as the larger the image size, the higher the resolution and the richer the information in the image. At the same time, especially for transformer-based point matchers [6, 8, 40], different input image sizes produce widely varying matching results. To investigate the effectiveness of our MESA under different image sizes, we construct experiments on ScanNet1500 benchmark [11]. In particular, we combine our MESA with both semi-dense point matcher ASpan [8] and dense point matcher DKM [17] to estimate relative pose from input images with three different sizes: 640×640, 480×480 and 320×320. It is noteworthy that we choose square sizes, because areas have different raw sizes and square input leads to smallest distortion from resize. The original point matchers (ASpan and DKM) are trained in 640×480 and fine-tuned in 640×640, using ground truth area matches generated by our method. We compare MESA_baselines with original baselines under the same input size to demonstrate the effectiveness of our methods. The results are summarised in Tab. 6. The fine-tuned point matchers achieve comparable results in 640×640 with original ones in 640×480, proving the fine-tuning is an

Input Image Size	Semi-Dense Matcher	AUC@5 ↑	AUC@10 ↑	AUC@20 ↑	Dense Matcher	AUC@5 ↑	AUC@10 ↑	AUC@20 ↑
640 × 640	ASpan	26.33	45.98	62.12	DKM	28.63	50.84	68.97
	MESA_ASpan	$27.51_{+4.48\%}$	$47.47_{+3.24\%}$	$65.04_{+4.70\%}$	MESA_DKM	$33.42_{+16.73\%}$	$55.04_{+8.26\%}$	$71.98_{+4.36\%}$
480 × 480	ASpan	21.38	40.53	58.74	DKM	28.85	50.06	68.20
	MESA_ASpan	$24.01_{+12.30\%}$	$43.32_{+6.88\%}$	$60.99_{+3.83\%}$	MESA_DKM	$33.00_{+14.38\%}$	$54.04_{+7.95\%}$	$71.02_{+4.13\%}$
320 × 320	ASpan	8.95	21.18	37.68	DKM	27.55	48.20	65.96
	MESA_ASpan	$9.60_{+7.26\%}$	$22.17_{+4.67\%}$	$38.70_{+2.71\%}$	MESA_DKM	$31.43_{+14.08\%}$	$52.56_{+9.05\%}$	$69.99_{+6.11\%}$

Table 6. **Ablation study of area image size.** We investigate the performance impact of three different input image sizes for both semi-dense and dense methods, along with our MESA combined with them. For point matchers, the input image size is the size of resized original image. For our method, the input image size is the area image size. The pose estimation AUC@$5^\circ/10^\circ/20^\circ$ are reported for evaluation.

Method	Cropping Approach	AUC@5 ↑	AUC@10 ↑	AUC@20 ↑
MESA_ASpan	*OAR*	24.67	43.72	61.29
	ARPM	**27.51**	**47.47**	**65.04**
MESA_DKM	*OAR*	30.19	51.49	68.79
	ARPM	**33.42**	**55.04**	**71.98**

Table 7. **Ablation study of area image cropping.** Two different image cropping methods are compared for the proposed MESA. Both semi-dense and dense point matchers are combined for evaluation. We report the pose estimation AUC@$5^\circ/10^\circ/20^\circ$ and the **best** results of two series are highlighted respectively.

E_G Parameters	$T_{E_{max}}$	AOR ↑	AMP@0.6 ↑	Pose AUC@5° ↑	AreaNum ↑
$\mu=5,\ \alpha=2,$ $\beta=2,\ \gamma=1$	0.35	61.76	65.54	23.57	4.69
	0.25	63.91	71.13	22.41	3.47
	0.15	60.44	62.57	21.46	3.27
$\mu=4,\ \alpha=2,$ $\beta=2,\ \gamma=2$	0.35	**67.98**	**80.09**	23.74	**5.76**
	0.25	64.94	72.24	**24.01**	4.62
	0.15	61.74	65.50	23.55	3.86
$\mu=7,\ \alpha=1,$ $\beta=1,\ \gamma=1$	0.35	65.98	78.10	22.71	3.27
	0.25	62.32	66.54	23.56	2.92
	0.15	60.32	64.38	22.37	2.77

Table 8. **Ablation study of global energy parameters.** We compare different parameter settings for global energy refinement in MESA_ASpan and report the area matching performance, area number per image (AreaNum), and the pose estimation performance. Results are highlighted as **first** , second and third .

Method	LoFTR [40]	PATS [21]	DKM [17]	COTR [20]	MESA (Ours)
Time(ms)/img	181.5	450.4	1318.5	18975.3	3092.4

Table 9. **Time consumption comparison.** All methods run on 640 × 480 images.

effective way to remove the impact from inconsistent image sizes and scales between training and testing. Overall, MESA effectively increases the performance for both point matchers under all input sizes. For the semi-dense point matcher ASpan, the accuracy decreases significantly as the input size decreases, whereas the improvement achieved by MESA remains noticeable. Moreover, the improvement under 480×480 is much better than under 640×640 (12.30% $vs.$ 4.48%), revealing the benefits of high resolution provided by MESA. On the other hand, DKM is much more robust under different input sizes and only gets slight performance declines with smaller input sizes. Meanwhile, our MESA achieves impressive improvements under all input sizes proving the effectiveness of MESA. Furthermore, it is worth noting that MESA_DKM achieves better results with smaller input size than original DKM *e.g.*, 33.00 for MESA_DKM under 480×480 is better than 28.63 for DKM under 640×640, indicating the superiority of MESA. In sum, MESA enables effective matching redundancy reduction which allows for high resolution input with less matching noises, leading to advanced feature matching under different input image sizes.

9.2. Ablation Study on Image Cropping

The image cropping is a trivial yet important operation for the A2PM framework, as different cropping approaches lead to different image resolutions and distortions. Here, we construct experiments to investigate the impact of different cropping approaches. To be specific, we compare two different cropping methods descried in Sec. 7.3: **1)** the direct cropping method (*OAR*), which crops with Original Aspect Ratios of areas; **2)** the cropping method with the Aspect Ratio of Point Matcher (*ARPM*), which first expands the area to correspond with the aspect ratio of the point matcher input and then crop areas. The experiment is conducted on ScanNet1500 [11] benchmark. We combine MESA with both semi-dense (ASpan) and dense (DKM) point matchers for complete comparison. Results are summarized in Tab. 7. As we can seen that the ARPM cropping approach outperforms the OAR approach with a large margin for both MESA_ASpan and MESA_DKM, proving its superiority due to high resolution and less distortion. Therefore, we adopt the ARPM approach for area image cropping in MESA.

9.3. Ablation Study on Global Energy Parameters

The parameters for our global energy refinement mainly consists of global energy balance parameters (E_G Parameters) in Eq. (11) and the threshold parameter $T_{E_{max}}$. The four E_G Parameters reflect the importance of four energy terms, *i.e.*, self matching energy, parent, children and neighbour matching energy. The $T_{E_{max}}$ controls the maximum

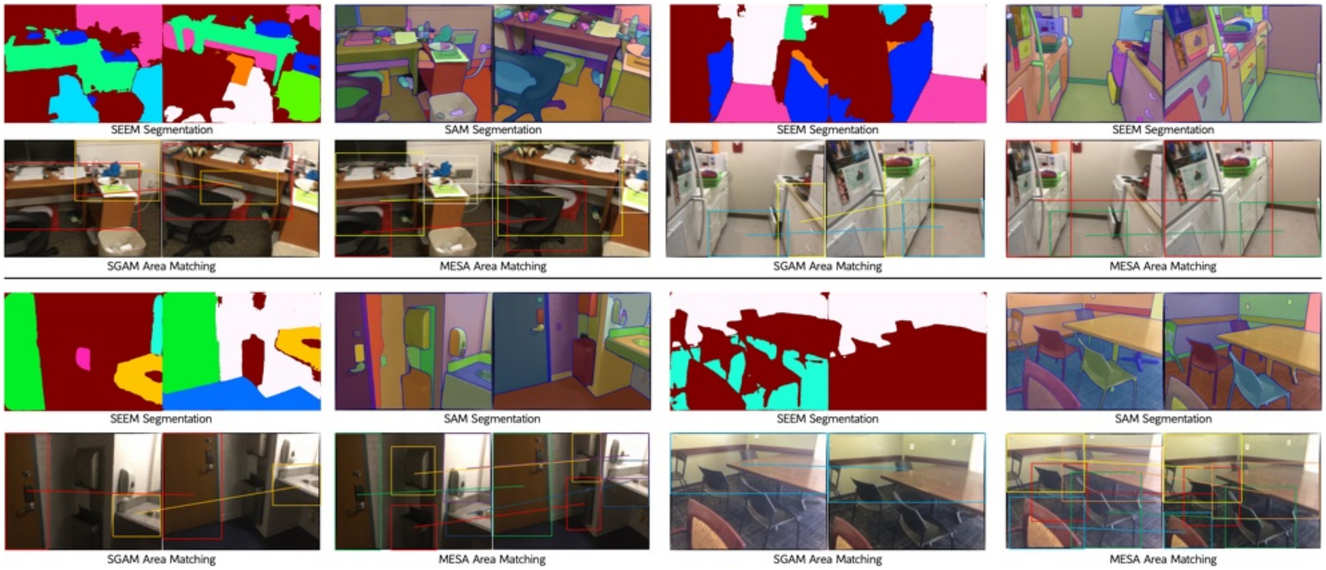

Figure 7. **The Qualitative Results of Area Matching on ScanNet.** The area matching results of both SGAM [52] and MESA are displayed. For better comparison, we also show the SEEM [54] and SAM [22] results which serve as the input for SGAM and MESA respectively. Different colors in SEEM results represent different labels, but colors in SAM results are only used to distinguish between different segments. The main drawbacks of SGAM are inaccurate semantic labeling (left top), semantic ambiguity (right top) and unrecognised objects (bottom), which are successfully avoided in our MESA.

energy of the final match, the smaller it is the stricter the refinement. Here, we construct experiments on ScanNet1500 to investigate the performance impact of these parameters. In particular, we compare three groups of E_G Parameters and three groups of $T_{E_{max}}$ to evaluate their impact on MESA_ASpan. The input size of ASpan is 480×480. The area matching performance, pose estimation performance and area number per image are summarised in Tab. 8. Generally, if two areas are matched, their parent, children and neighbour nodes should have high similarities due to spatial relationships between them. At the same time, the self matching energy should still be an important reference in matching refinement. Thus we choose three parameter settings including different weights on three kinds of node matching energies and different emphasis on self-matching energy. The experiment results in Tab. 8 show that the weights of three parameter settings set to the same is better for area matching performance ($\alpha = \beta = \gamma$ *vs.* $\alpha = \beta \neq \gamma$). Giving sufficient consideration on global matching leads to accurate area matching along with best point matching performance ($\mu = 4$ *vs.* $\mu = 7$). Despite the semi-dense matcher is not sensitive to area matching accuracy, better area matching leads to higher pose estimation precision. Therefore, we choose $[4, 2, 2, 2]$ as our energy setting. On the other, the $T_{E_{max}}$ is a critical parameter as well. The smaller $T_{E_{max}}$ means stricter global matching energy request, but it may also mistake some accurate area matches when too small. Different E_G Parameter settings prefer different values of $T_{E_{max}}$ and 0.35 suits the best for ours.

10. Qualitative Examples

We show some qualitative examples of our method in this section, including both area matching (Sec. 10.1) and point matching(Sec. 10.2 and Sec. 10.3).

10.1. Area Matching Comparison.

As can be seen in Fig. 7, we show some qualitative results of area matching of both MESA and SGAM [52] to compare their performances. It is evident that erroneous area matches (Fig. 7 top) achieved by SGAM are mainly resulted from inaccurate semantic labeling by SEEM [54]. At the same time, MESA addresses this issue successfully by the utilizing of SAM segmentation, global context modeling by AG and learning-based area similarity calculation. Thus MESA is able to obtain precise area matches in SGAM-failed examples. On the other hand, recognition failure in semantic segmentation also leads to few area matches in SGAM, as shown in Fig. 7 bottom, which hinder the performance of subsequent point matching. In contrast, MESA finds area matches from the SAM segmentation without semantic label, hereby overcoming this limitation, and is able to expand the benefit of A2PM to more general scenes.

10.2. Outdoor Feature Matching.

We show some qualitative results of outdoor feature matching in Fig. 8. We combine MESA with two different point matchers, consisting of both semi-dense [8] and dense [17] matchers, to demonstrate the performance improvement achieved by MESA. It can be seen that our MESA can

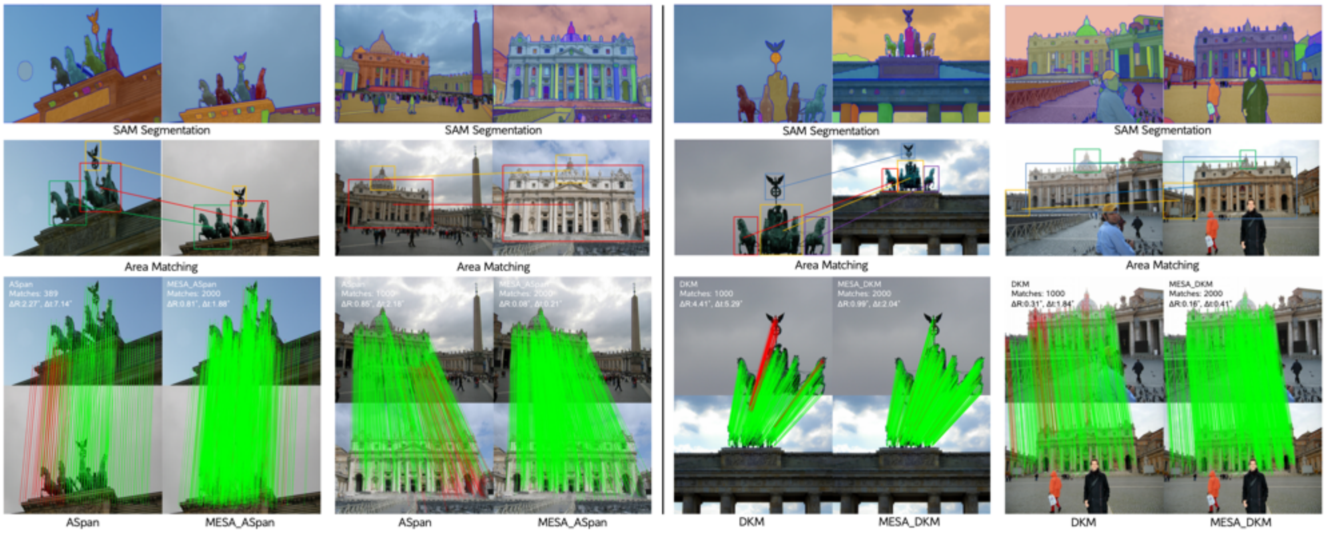

Figure 8. **The Qualitative Results on MegaDepth.** The improvements achieved by our MESA for both semi-dense [8] and dense [17] baselines are depicted. We draw more matches for MESA ($\leq 2k$) than baselines ($\leq 1k$), to better demonstrate the matching distribution and higher matching precision of MESA. All the pose errors in the top left of the bottom images are estimated with up to 1k matches.

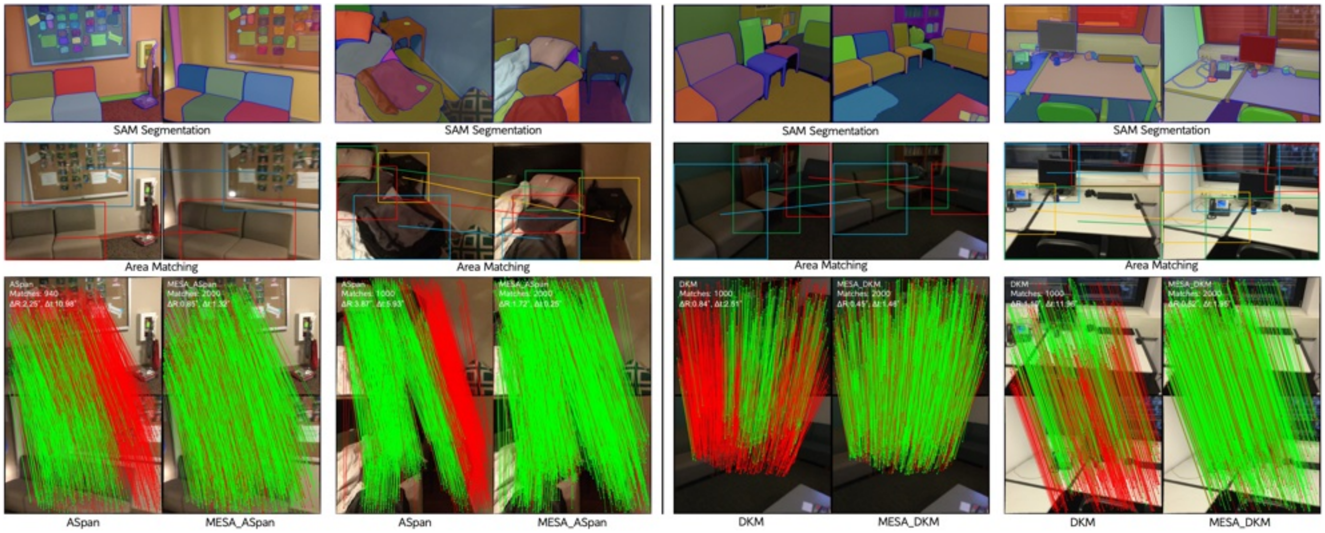

Figure 9. **The Qualitative Results on ScanNet.** The improvements achieved by our MESA for both semi-dense [8] and dense [17] baselines are depicted. We draw more matches for MESA ($\leq 2k$) than baselines ($\leq 1k$), to better demonstrate the matching distribution and higher matching precision of MESA. All the pose errors in the top left of the bottom images are estimated with up to 1k matches.

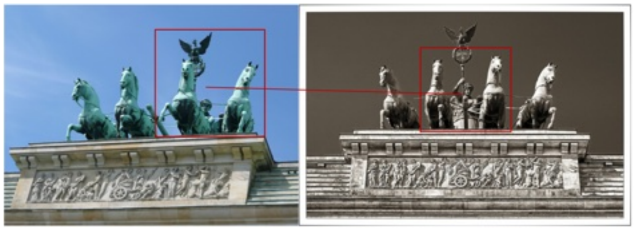

Figure 10. **The Failure case on MegaDepth.** MESA may produce false area matches when repeated objects and viewpoint variance occur at the same time. The impact of this kind of erroneous match can be alleviated by post-processing like GAM [52].

achieve accurate area matches from SAM segmentation, thereby obtains more accurate point matches and pose estimation results. Point matches of MESA are clustered in multiple areas. Moreover, the areas matched by MESA are unable to be specified with some semantic labels, thus can not be established by semantic segmentation-based method, *i.e.*, SGAM [52]. This proves the superiority of MESA to extend the benefits of A2PM framework in outdoor scenes.

10.3. Indoor Feature Matching.

In Fig. 9, we show some qualitative results of indoor feature matching. We also combine MESA with ASpan [8] and DKM [17], to compare the performances with original matchers. MESA achieves precise and robust area matches, leading to prominent matching precision improvement. Notably, although inconsistent area fusion in AG construction exists in MESA, *e.g.*, different fusions of multiple chairs in the first and third columns, the final weight-based fusion improves the accuracy of area matching, similar to PATS [21]. Finally, the superior point matching resulting from MESA contributes to impressive pose estimation improvement.

11. Limitation and Future Work

One limitation of MESA is its under-utilisation of SAM features. As we mentioned before, SAM possesses the high-level image understanding across a wide range of domains due to the massive training dataset and carefully designed models. Therefore, its image embedding is a extremely strong high-level representation, which has the potential to replace our learning similarity model. Then, the computation cost can be reduced as well. However, the naive attempt to use SAM features as descriptors of areas failed, possibly because the SAM segmentation pays more attention on intra-image contexts rather than inter-image ones like feature matching. Hence, the SAM feature needs further distillation for area matching, which will be a objective of our future work.

On the other hand, as MESA fuses image areas based on their 2D distances, which is not equivalent to the 3D situations. Thus, some inconsistent area fusions between two images arise and lead to inaccurate point matching, *e.g.*, shown in Fig. 10. Although the post-processing like GAM [52] may help, it also introduces extra computation cost. To address this issue, feature-guided fusion can be adopted, where the SAM feature can be employed and lead to consistent area fusion.

Another limitation of MESA is the speed, which takes around 3s per image for area matching as shown in Tab. 9, limiting its performance in latency-sensitive applications like SLAM. In Tab. 9, we also compare MESA with recent point matching methods with regard to running speed, all of them get 640×480 image as input and are implemented on a single NVIDIA 4090 GPU. Although MESA is slower than most point matchers, it is still faster than COTR [20]. As area matching is a pre-task for point matching, similar or even faster speed is the object of practical area matching for real-time tasks. It is also worthy to note that the speed of MESA is independent to image size (fixed area size is adopted in MESA), while these point matching methods will get significant increase in elapsed time when the image size is increased. This drawback is mainly caused by multiple similarity calculations in MESA, where the utilisation of SAM features may be helpful as described above. At the same time, engineering technologies, like parallel area similarity computing, can facilitate our area matching. We will investigate these possibilities in our future work.