Computational Visual Media
https://doi.org/10.1007/s41095-023-0333-9 Vol. 10, No. 2, April 2024, 309–330

Research Article

Learning accurate template matching with differentiable coarse-to-fine correspondence refinement

Zhirui Gao[1], Renjiao Yi[1], Zheng Qin[1], Yunfan Ye[1], Chenyang Zhu[1], and Kai Xu[1] (✉)

Abstract Template matching is a fundamental task in computer vision and has been studied for decades. It plays an essential role in manufacturing industry for estimating the poses of different parts, facilitating downstream tasks such as robotic grasping. Existing methods fail when the template and source images have different modalities, cluttered backgrounds, or weak textures. They also rarely consider geometric transformations via homographies, which commonly exist even for planar industrial parts. To tackle the challenges, we propose an accurate template matching method based on differentiable coarse-to-fine correspondence refinement. We use an edge-aware module to overcome the domain gap between the mask template and the grayscale image, allowing robust matching. An initial warp is estimated using coarse correspondences based on novel structure-aware information provided by transformers. This initial alignment is passed to a refinement network using references and aligned images to obtain sub-pixel level correspondences which are used to give the final geometric transformation. Extensive evaluation shows that our method to be significantly better than state-of-the-art methods and baselines, providing good generalization ability and visually plausible results even on unseen real data.

Keywords template matching; differentiable homography; structure-awareness; transformers

1 Introduction

Template matching aims to find given templates in captured images (source images), and is a fundamental technique for many computer vision tasks, including object detection, visual localization, pose estimation, etc. Numerous approaches have been proposed to overcome the difficulties and challenges in template matching, and the problem may seem to be solved. Nevertheless, this problem is a critical step in automatic processing on industrial lines, and in real scenarios, various challenges remain, including domain gap, size variance, and pose differences between template and source image. The above challenges motivate our approach of *accurate template matching based on differentiable correspondence refinement*.

Classic methods of template matching [1–4] generally calculate a similarity score between the template and a candidate image patch. Linemod-2D [1] utilizes gradient spreading and gradient orientation similarity measures, achieving real-time detection

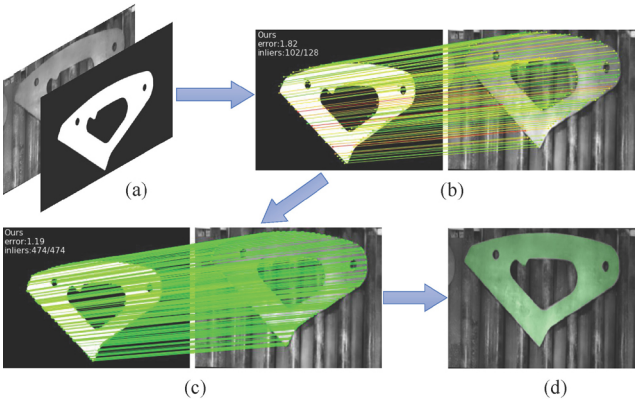

Fig. 1 Our template matching method. (a) Template T and image I. (b) Coarse matching. (c) Matching refinement. (d) Template warped to the image using the estimated geometric transformation.

1 College of Computer, National University of Defense Technology, Changsha 410073, China. E-mail: Z. Gao, gzrer2018@gmail.com; R. Yi, yirenjiao@nudt.edu.cn; Z. Qin, qinzheng12@nudt.edu.cn; Y. Ye, yunfan951202@gmail.com; C. Zhu, zhuchenyang07@nudt.edu.cn; K. Xu, kevin.kai.xu@gmail.com (✉).

Manuscript received: 2022-10-17; accepted: 2023-01-02

with high accuracy and robustness for untextured objects and is widely used in industry. However, its performance degrades significantly in the presence of cluttered backgrounds, image blurring, or non-rigid deformation between template and source; these are all common in real-world applications.

Deep learning has shown great potential to overcome such distractors, providing significant improvement for many similar tasks. Related work [5–8] has designed new network structures or similarity metrics based on deep features to improve the robustness of template matching. However, these works all aim to determine the bounding box of the target object, rather than accurate pixel-level matching and pose of the template. Bounding boxes are insufficient for tasks requiring high precision: for example, robot manipulators need a precise object pose to decide the best grasping direction. Rocco et al. [9] estimate such transformations, but their method fails in cross-modal scenarios where templates are mask images and observed images are in color or grayscale.

Thus, our work considers the design of an automatic pipeline for determining a high-quality transformation between the template mask and source image. To allow for the domain difference between a template mask and a grayscale image, an edge translation module is used to convert them to the same modality. To achieve a high-quality transformation estimate, we propose a novel structure-aware outlier rejection approach based on coarse-to-fine correspondence refinement. As a result, the proposed method not only tolerates different modalities in matching, but also deals with occlusion to some degree as well as complex deformations.

In feature correspondence matching, many recent works [10–13] have made remarkable progress in deep feature matching; e.g., LoFTR [13] uses transformers for this task and omits the feature detection step. However, there are limitations when applying LoFTR directly to template matching problems. Firstly, it tends to fail on cross-modal images, when mask images and grayscale images lie in very different in feature spaces. Secondly, the structural consistency of templates and images is not exploited, yet it is critical for accurate matching. More importantly, LoFTR cannot provide sufficiently accurate and reliable correspondences for untextured regions, or when large deformations exist, as in the cross-modal template matching problem.

Motivated by the challenges, we propose a differentiable structure-aware template matching pipeline. To address the modality difference between the template and the source image, we use a translation module to convert both of them to edge maps. We believe that structural information is particularly important for fast and robust template matching: *the template mask has a specific structure (shape) and correct correspondences between the template and the image should satisfy a specific transformation relationship.* Therefore, we fully exploit template contour information and consider compatibility of angles and distances between correspondences. Specifically, we apply three strategies in our model to better use the structural information of templates and images. Firstly, in order to focus the network on valid areas, we only sample contour regions of the template as the input. Then the transformer [14] using relative positional encoding [15] is used to explicitly capture relative distance information. A method based on distance-and-angle consistency rejects soft outliers.

In pursuit of high-quality template matches, the transformation between the template and the source image is estimated in a coarse-to-fine style. In the coarse-level stage, we use transformers [14] to encode local features extracted by the convolutional backbone and then establish feature correspondences using a differentiable matching layer. By assigning confidences to these coarse-level matches based on feature similarity and spatial consistency, we obtain a coarse estimate of the geometric transformation, a homography. This coarse matching overcomes differences in scale and large deformations between the source and template image, which is critical for accurate matching at the fine level. We apply the coarse spatial transform [16] to coarsely align the source image, which then provides an updated source image for the fine level. A refinement module is used at the fine-level to obtain global semantic features and to aggregate features at different scales. We then adopt a correlation-based approach to determine accurate dense matches at the sub-pixel level. These final correspondences are more accurate, and no outlier rejection is needed. All correspondences are used to calculate the final homography. Compared

to other recent matching methods [10–13], our correspondences have many fewer outliers, allowing our method to provide robust and accurate template matching without relying on RANSAC [17].

We use a linear transformer [18] in our pipeline to reduce computational complexity. Farthest point sampling (FPS) is applied to the template image to reduce the input data while retaining its structure. To solve the problem of insufficient training data, GauGAN [19] is adopted to generate synthetic images of industrial parts for network training.

We have evaluated the proposed method on three datasets, including two newly-collected industrial datasets and a dataset based on COCO [20]. Our approach provides significantly improved homography estimates compared to the best baseline, as we show later.

Our main contributions can be summarized as

- An accurate template matching method, robust in challenging scenarios including cross-modality images, cluttered backgrounds, and untextured objects.
- A structure-aware, fully differentiable, template matching pipeline, avoiding the use of RANSAC found in other feature matching approaches; it achieves state-of-the-art accuracy.
- Two new datasets with accurate ground truth, of potential benefit to future research on learning-based template matching.

2 Related work

2.1 Template matching

Traditional methods of template matching mostly rely on comparing similarities and distances between the template and candidate image patch, using such approaches as sum of squared differences (SSD), normalized cross-correlation (NCC), sum of absolute differences (SAD), gradient-based measures, and so on. Linemod-2D and the generalized Hough transform (GHT) [2] are widely applied in industry. Such approaches degrade significantly in the presence of cluttered backgrounds, image blurring, or large deformations. Deep learning-based template matching algorithms [5–9] can handle more complex deformations between the template and source image. They usually adopt trainable layers with parameters to mimic the functionality of template matching. Feature encoding layers are assumed to extract the features from both inputs; these deep feature encoders dramatically improve template matching results. While these methods still rely on the rich textures of input images. However, deep learning methods are prone to fail with cross-modal input and are typically unable to provide an accurate pose for the target object.

Motivated by these challenges, our method predicts a homography transformation, and uses an edge-aware module to eliminate the domain gap between the mask template and the grayscale image for robust matching.

2.2 Homography estimation

Classical homography estimation methods usually comprise three steps: keypoint detection (using, e.g., SIFT [21], SURF [22], or ORB [23]), feature matching (feature correlation), and robust homography estimation (using, e.g., RANSAC [17] or MAGSAC [24]). However, RANSAC-like approaches are non-differentiable. Furthermore, differentiable RANSAC algorithms [25, 26] hinder generalization to other datasets. Other methods, such as the seminal Lucas–Kanade algorithm [27], can directly estimate the homography matrix without detecting features. The first deep learning-based homography estimation model was proposed in Ref. [28]. Its network regresses the four corner displacement vectors of the source image in a supervised manner and yields the homography using a direct linear transform (DLT) [29]. Many unsupervised approaches [30–32] have been proposed to minimize the pixel-wise photometric error or feature difference between the template and source image.

These methods are likely to fail under large viewpoint change, when textures are lacking, and for differing input modalities. Our work uses the template's structural (shape) properties and samples, valid region features in the template to learn the correlation with the source image. An edge-aware module is used to translate the source image and template mask to bypass the effect of modality differences between two inputs.

2.3 Feature matching

Before the era of deep learning, hand-crafted local features such as SIFT, SURF, and ORB were widely adopted. Deep learning-based methods [33–35]

significantly improve the feature representation, especially in the cases of significant viewpoint and illumination changes. SuperPoint [33], D2-Net [34], and ASLFeat [36] propose joint learning of feature descriptors and detectors; most computations of the two tasks are shared for fast inferencing using a unified framework. A significant improvement in feature matching was achieved by SuperGlue [12], which accepts two sets of keypoints with their descriptors, and updates their representations with an attentional graph neural network (GNN). Drawing inspiration from GNN, more methods [37–40] further improve the accuracy and efficiency of graph-based feature matching. Recently, several works [11, 13, 41] have attempted to adopt transformers to model the relationship between features and provide impressive results. In this work, we build on the success of transformers and learn accurate template matching with coarse-to-fine correspondence refinement.

2.4 Vision transformers

Transformers [14] were initially proposed in natural language processing (NLP). Vision transformers [42] have attracted attention due to their simplicity and computational efficiency for image sequence modeling. Many variants [18, 43–45] have been proposed for more efficient message passing. In our work, we utilize self and cross attention to establish larger receptive fields and capture structural information from the inputs. In particular, linear transformers [18] with relative positional encoding are adopted to ensure low computational costs and more efficient message passing.

3 Overview

In industrial template matching, it is usual for the template to be represented as a binary mask indicating only the shape of the source object. In contrast, the source image is often grayscale. Thus, we first use an edge-aware translation module before feature extraction to eliminate the domain difference between these two images: see Section 4.2.1. We propose a differentiable feature extraction and aggregation network with transformers in Sections 4.2.2 and 4.2.3. The whole matching pipeline is performed in a coarse-to-fine style. At the coarse level, to estimate the homography from matched features, we combine spatial compatibility and feature similarity for soft outlier filtering: see Section 4.3; this is RANSAC-free and differentiable. A coarse homography is obtained from the coarse correspondences. Then, we apply the spatial transform [16] to the source image to provide a coarsely-aligned image. At the fine-level, we combine global semantics and local features to achieve sub-pixel dense matching and obtain an accurate homography estimate, as explained in Section 4.4. The final correspondences are precise between the template mask and source image, ensuring a plausible template matching result. Inspired by LoFTR, we adopt a coarse-to-fine matching pipeline, as shown in Fig. 2. Note that unlike LoFTR, our approach takes full advantage of the geometric properties of the template and spatial consistency between the template and the object. In addition, our coarse-to-fine matching process is fully differentiable via a spatial transform connection, while LoFTR's coarse-to-fine strategy only enhances correspondence accuracy and is not fully differentiable.

4 Method

4.1 Task

Given a binary template image T and a source search image I, our method aims to estimate a homography transformation between T and I to provide the precise position and pose of the object in the image I. For applications whose scenes have multiple objects and multiple candidate templates, the coarse stage of our method may be performed first to estimate the initial homography for selecting the correct template for each object. We then use the refinement stage to obtain the precise position and pose.

4.2 Feature extraction and aggregations

4.2.1 Edge translation

Unlike some other template matching and homography estimation tasks, the case considered here has a domain difference between the template T and source image I. The former is a binary mask, and the latter is a grayscale image; their features are too different to use common image matching approaches. Grayscale images may furthermore exhibit strong reflections if the material is glossy. Matching based on photometric similarity are not applicable in such cases. Firstly, to ensure domain consistency of the template and source image, and

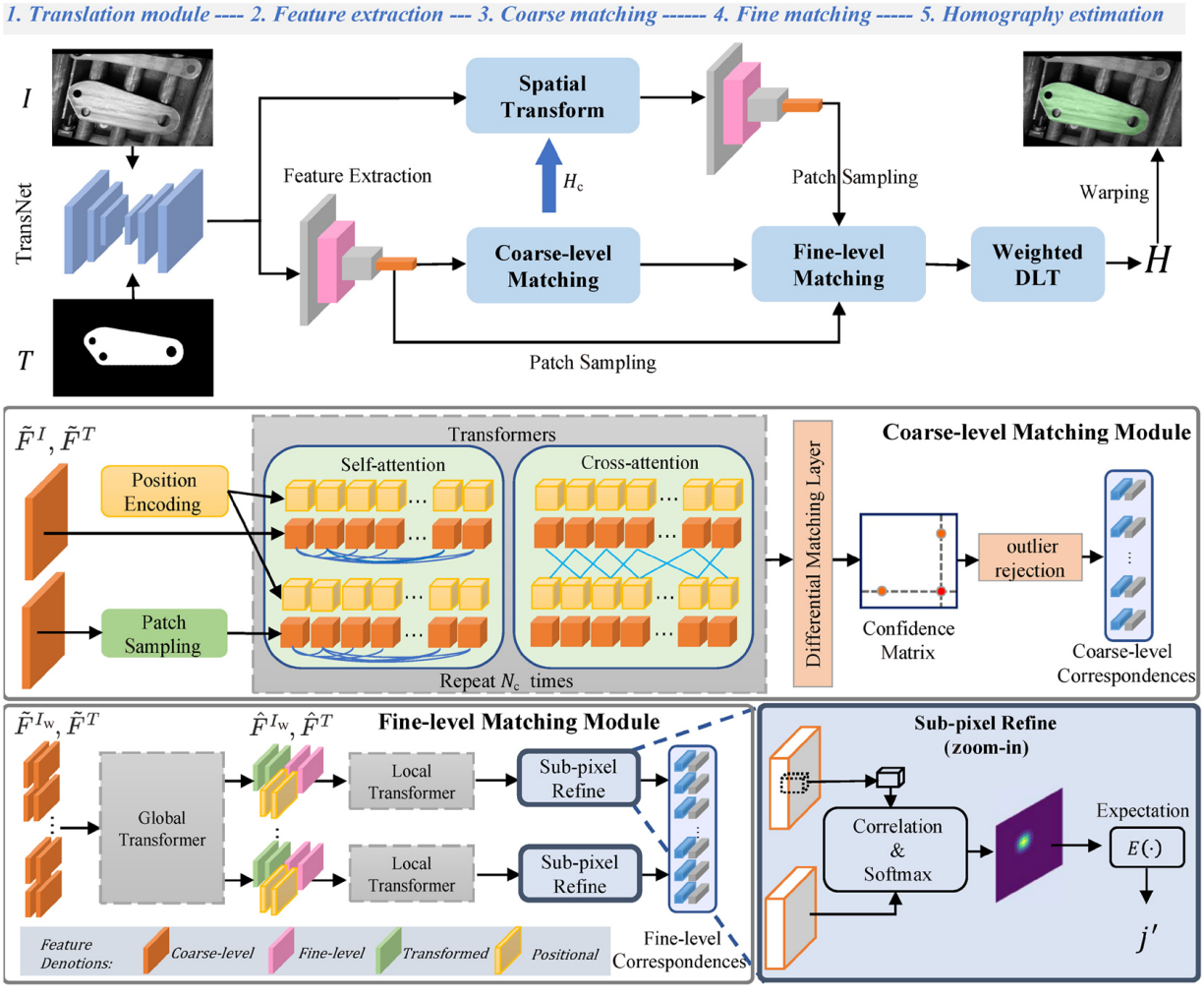

Fig. 2 Pipeline: the proposed method has five steps. (1) Translation module: convert the source image I and template mask T into edge maps (Section 4.2.1). (2) Feature extraction: extract coarse-level feature maps and fine-level feature maps (Section 4.2.2). (3) Coarse matching: two sets of coarse-level features are aggregated by interleaving self and cross attention layers to provide the initial homography transformation H_c (Section 4.3). (4) Fine-level matching: global and local features are fused to give the set of sub-pixel level matches $\mathcal{M}_f$ (Section 4.4.1). (5) Homography estimation (Section 4.4.2).

to avoid complications from reflections, we adopt a translation network to convert both into edge maps. In this step, we adopt PiDiNet [46], a lightweight and robust edge detector, to compute the edge maps. This conversion is crucial to permit later feature matching.

4.2.2 Feature extraction

We use a standard convolutional architecture similar to SuperPoint to extract features at different scales from both images after translation. SuperPoint has a VGG-style [47] encoder trained by self-supervision and shows leading performance in many vision tasks [12, 48, 49]. We only retain the encoder architecture of SuperPoint as our local feature extraction network. Given an input image of size $H \times W$, our feature extraction networks produce feature maps at four resolutions; we save the second layer feature map ($\hat{F} \in \mathbb{R}^{H/2\times W/2\times D}$) and the last layer feature map ($\tilde{F} \in \mathbb{R}^{H/8\times W/8\times C}$). Thus, $\tilde{F}^T$ and $\tilde{F}^I$ are the coarse-level features, $\hat{F}^T$ and $\hat{F}^I$ are the fine-level features.

4.2.3 Feature aggregation with transformers

Since edge images are not richly textured, the features extracted by the local convolutional neural network are inadequate for robust feature matching. Structural and geometric features are more significant [50]. Therefore, we adopt transformer blocks [14] to encode $\tilde{F}^T$ and $\tilde{F}^I$ to produce more global, position-aware features denoted $\tilde{F}^T_{\text{tr}}$ and $\tilde{F}^I_{\text{tr}}$. A transformer block consists of a self-attention layer to aggregate the global context and a cross-attention layer to exchange information between two feature sets.

Patch sampling. Unlike previous work [11, 13] which passes all patches of the image into the transformer module, we only keep meaningful feature map patches in $\tilde{F}^T$. Specifically, we use furthest point sampling [51] to sample N_{p} patches in which edge pixels exist, both to reduce computational cost and increase the efficiency of message passing. $\tilde{F}^T$ henceforth denotes the feature map after sampling. We do not drop any patches of the source image I: every location in I could be a potential match. We perform experiments to show the effect of patch sampling with various numbers of patches in Section 5.5.1.

Positional encoding. In transformers, all inputs are fed in simultaneously, and furthermore, do not encode any information concerning input ordering (unlike RNNs). We must encode positional information for the tokens input into transformers in order to make available the order of the sequences. Previous feature matching work using transformers [11, 13] uses a 2D extension of the standard absolute positional encoding, following DETR [42]. In contrast, Refs. [15, 52] showed that relative positional encoding is a better way of capturing the positional relationships between input tokens. We employ a rotary position embedding [53] proposed in natural language processing for position encoding which has recently been successfully adopted for point cloud processing [52]. We apply it to 2D images as it can express a relative position in a form like absolute position encoding. Furthermore, it can be perfectly incorporated in linear attention [18] at almost no extra cost. In order to obtain the relative positional relationship of the local features between the template and image, we thus use relative positional encoding in a linear transformer. For a given 2D location $n = (x, y) \in \mathbb{R}^2$, and its feature $f_n \in \mathbb{R}^C$, the relative positional encoding is defined as

$$\mathcal{P}(n, f_n) = \Theta(n) f_n = \begin{pmatrix} M_1 & & \\ & \ddots & \\ & & M_{C/4} \end{pmatrix} f_n$$

where

$$M_k = \begin{pmatrix} \cos x\theta_k & -\sin x\theta_k & 0 & 0 \\ \sin x\theta_k & \cos x\theta_k & 0 & 0 \\ 0 & 0 & \cos y\theta_k & -\sin y\theta_k \\ 0 & 0 & \sin y\theta_k & \cos y\theta_k \end{pmatrix}$$

$$\theta_k = \frac{1}{10000^{4(k-1)/C}}, \quad k \in [1, \cdots, C/4]$$

and C is the number of feature channels.

Rotary position embedding satisfies:

$$(\Theta(m) f_m)^T (\Theta(n) f_n) = f_m^T \Theta(n - m) f_n \tag{1}$$

and $\Theta(n - m) = \Theta(m)^T \Theta(n)$. Thus, relative position information between features f_n and f_m can be explicitly revealed by taking the dot product in the attention layer. This position encoding is more suitable in our application than absolute positional encoding, since relative positional relationships between template T and image I is crucial. $\Theta(\cdot)$ is an orthogonal operation on features, which means that it only changes the directions but not the lengths of feature vectors. Therefore, rotary position embedding stabilizes and accelerates the training process [52], facilitating downstream feature matching tasks. An experimental comparison to absolute positional encoding can be found in Section 5.5.1.

Self-attention and cross-attention layers. The key to the transformer model is attention. We use self and cross attention alternately in our pipeline. The input vectors for an attention layer are query vector Q, key vector K, and value V, and a basic attention layer is given by

$$\mathrm{Attention}(Q, K, V) = \mathrm{Softmax}(QK^T)V$$

Suppose Q and K have length N, and their feature dimensionality is C. Then the computational cost of the transformer grows as the square of the length of the input. The length of the source image T's input token makes a basic version of the transformer

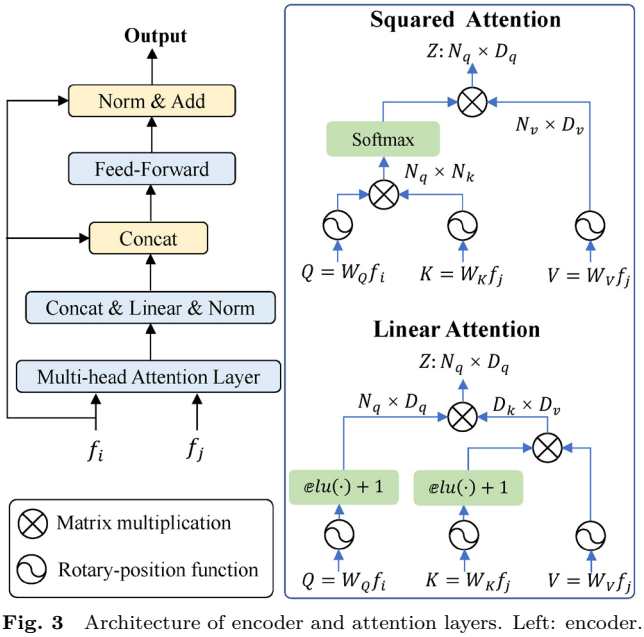

Fig. 3 Architecture of encoder and attention layers. Left: encoder. Right: squared ($O(N^2)$ complexity) attention layer and linear ($O(N)$ complexity) attention layer.

impractical for local feature matching. Following Ref. [13], we adopt a more efficient variant of the attention layer, linear transformer [18]. We use a kernel function $\mathrm{sim}(Q,K)=\phi(Q)\phi(K)^T$ to replace the softmax calculation, where $\phi(\cdot)=\mathrm{elu}(\cdot)+1$. The computational cost is reduced from $O(N^2)$ to $O(N)$ when $C \ll N$. Following RoFormer [53], we do not inject rotary position embedding in the denominator to avoid the risk of dividing by zero. Differing from Refs. [52, 53] as well as query Q and key K, the value V is also multiplied by $\Theta(\cdot)$, since we consider the position information to be important auxiliary information for value V. Experiments justifying this approach are described in Section 5.5.1.

Overall, each token in a linear transformer with relative positional encoding is given by

$$\mathrm{Attention}(Q,K,V)_m = \frac{\sum_{n=1}^{N}\left(\Theta(m)\phi(q_m)\right)^T\left(\Theta(n)\phi(k_n)\right)\left(\Theta(n)v_n\right)}{\sum_{n=1}^{N}\phi(q_m)^T\phi(k_n)}$$

4.3 Coarse matching

4.3.1 Establishing coarse matches

We establish coarse matches using the transformed features $\tilde{F}_{\mathrm{tr}}^T$ and $\tilde{F}_{\mathrm{tr}}^I$. An optimal transport (OT) layer is adopted as our differentiable matching layer. We first calculate a score matrix S using dot-product similarity of the transformed features:

$$S(i,j)=\left\langle \tilde{F}_{\mathrm{tr}}^T(i), \tilde{F}_{\mathrm{tr}}^I(j)\right\rangle$$

This score matrix S is used as the cost matrix in a partial assignment problem, following Refs. [12, 13]. This optimization problem can be efficiently solved with the Sinkhorn algorithm [54] to obtain the confidence assignment matrix C.

To obtain more reliable matches, the mutual nearest neighbor (MNN) criterion is enforced, and only matching pairs with confidence values higher than a threshold θ_{c} are preserved. The set of coarse-level matches $\mathcal{M}_{\mathrm{c}}$ is thus:

$$\mathcal{M}_{\mathrm{c}}=\{\,(i,j)\mid \forall\,(i,j)\in \mathrm{MNN}(C),\ C(i,j)\geqslant \theta_{\mathrm{c}}\}$$

Another matching layer approach is based on dual-softmax (DS) [55, 56]. It applies softmax to both dimensions of S to get the probability of a mutual nearest neighbor match. A comparison of OT and DS methods can be found in Section 5.5.1.

4.3.2 Confidence weights based on spatial consistency

The differentiable matching layer provides a tentative match set $\mathcal{M}_{\mathrm{c}}$ based on feature dot-product similarity. In this way, two irrelevant points may be regarded as a matching pair due to similarity of appearance. To prevent this, we add a new constraint, based on the observation that template matching, has an essential property: correct correspondences (inliers) have similar geometric transformations, while transformations of outliers are random.

RANSAC and its variants [57, 58] are widely adopted for outlier rejection. However, such methods are slow to converge and may fail in the cases of high outlier ratios. In contrast, spectral matching (SM) [59] and its variants [60–63] significantly improve results for rigid point cloud registration, by constructing a compatibility graph which preserves angle or distance invariance between point pairs. In contrast, our model assumes a non-rigid deformation in which pairwise distances between far points are more likely to vary than between closer ones. We thus extend SM and propose a method based on distance-and-angle consistency for non-rigid deformation outlier rejection.

Let β denote the distance compatibility term measuring the change in lengths of matched pairs. To allow for scale differences, we first normalize the distances between matching points on the template and image separately. Then for two coarse matches $a=(i,i')$ and $b=(j,j')$, β is defined as

$$\beta_{(a,b)}=\left[1-\left(d_{ij}/d_{i'j'}-1\right)^2/\sigma_{\mathrm{d}}^2\right]_+$$

where d_{ij} is the normalized pairwise distance between i and j, $[\cdot]_+$ means $\max(\cdot,0)$, and σ_{d} is a distance parameter controlling sensitivity to changes in relative length. Changes in directions are also penalized using a triplet-wise angle. Inspired by Ref. [64], we compute angular compatibility from triplets of coarse feature points. For a matching pair $a=(i,j)$ with positions p_i and p_j, we first select the k nearest neighbors $\mathcal{N}_i$ of p_i. For each $p_x\in\mathcal{N}_i$, the angle $c_{i,j}^x=\measuredangle(\triangle_{i,x},\triangle_{i,j})$, where $\triangle_{i,j}=p_i-p_j$. To improve robustness, we select the maximum value c_{ij} among the k nearest neighbors as the angle property for a matching pair (i,j). As for distance compatibility β, we now formulate the angular compatibility α as

$$\alpha_{(a,b)}=\left[1-\left(c_{ij}-c_{i'j'}\right)^2/\sigma_{\alpha}^2\right]_+$$

where σ_α is the angular parameter controlling the sensitivity to changes inn angle. Figure 4 illustrates the computation of distance and angular consistency.

The final spatial compatibility of matches a and b

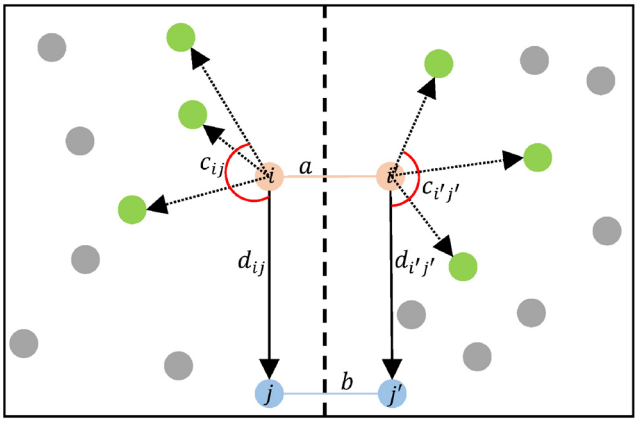

Fig. 4 Given two matching pairs $a = (i, i')$ and $b = (j, j')$, we calculate both their distance compatibility and their angular compatibility. Green nodes represent k-nearest neighbors.

is defined as

$$E(a,b) = \lambda_c \alpha_{(a,b)} + (1 - \lambda_c)\beta_{(a,b)}$$

where λ_c is a control weight. $E(a,b)$ is large only if the two correspondences a and b are highly spatially compatible. Following Refs. [59, 60], the leading eigenvector e of the compatibility matrix E is regarded as the inlier probability of the matches. We use the power iteration algorithm [65] to compute the leading eigenvector $e \in \mathbb{R}^k$ of the matrix E.

4.3.3 Initial homography estimation

Naturally, the inlier probability e together with feature score s must be combined to give the final overall inlier probability, where s is the corresponding element of the feature confidence matrix C. We simply compute $w_k = s_k \cdot e_k$: intuitively, w_k takes into account how similar the feature descriptors are (s_k) and how much the spatial arrangement is changed (e_k) for a matching pair k. Finally, we use the confidence w_k as a weight to estimate the homography transformation H_c, using the DLT formulation [29]. A weighted least squares solution is found to the linear system. The matches-with-confidence make our coarse-to-fine network differentiable and RANSAC-free, enabling end-to-end training. The effectiveness of confidence weights is explored in Section 5.5.1.

4.3.4 Coarse-level training losses

Following Ref. [13], we use negative log-likelihood loss over the confidence matrix C returned by either the optimal transport layer or the dual-softmax operation to supervise the coarse-level network. The ground-truth coarse matches $\mathcal{M}_c^{gt}$ are estimated from the ground-truth relative transformations (homographies). Using an optimal transport layer, the loss is

$$\mathcal{L}_c = -\frac{1}{\left|\mathcal{M}_c^{gt}\right|} \sum_{(i,j)\in\mathcal{M}_c^{gt}} \log(i,j) - \frac{1}{|\mathcal{I}|} \sum_{(i,j)\in\mathcal{I}} \log(i,j)$$

where $(i,j) \in \mathcal{I}$ means that i or j does not have any reprojection in the other image. With the dual-softmax operation, we minimize the negative log-likelihood loss in $\mathcal{M}_c^{gt}$:

$$\mathcal{L}_c = -\frac{1}{\left|\mathcal{M}_c^{gt}\right|} \sum_{(i,j)\in\mathcal{M}_c^{gt}} \log(i,j)$$

4.4 Fine-level matching

A coarse-to-fine scheme is adopted in our pipeline, a scheme which has been successfully applied in many vision tasks [10, 13, 66–69]. We apply the obtained coarse homography H_c to the source image I to generate a coarsely-aligned image I_w. We roughly align the two images, then use a refinement network to get sub-pixel accurate matches, and finally, a better-estimated transformation matrix is produced from the new matches.

4.4.1 Fine-level matching network

For a given pair of coarsely aligned images (warped image I_w and template T), sub-pixel level matches are calculated by our fine-level matching network to further enhance the initial alignment. Although Refs. [10, 56] claim that local features significantly improve matching accuracy in feature matching when refining, we find that local features are insufficient to achieve robust and accurate matching in untextured cases. Instead, we combine the global transformer and local transformer for feature aggregation to improve fine-level matching, as shown in Fig. 2.

The global transformer is first adopted to aggregate coarse-level features as priors. In detail, for every sampled patch pair $(\tilde{i}, \tilde{j})$ at the same location on template T and warped image I_w, the corresponding coarse-level features are denoted $\tilde{F}^T(\tilde{i})$ and $\tilde{F}^{I_w}(\tilde{j})$, respectively. A global transformer module with N_f self- and cross-attention layers operates on these coarse-level features to produce transformed feature $(\tilde{F}_{tr}^T(\tilde{i}), \tilde{F}_{tr}^{I_w}(\tilde{j}))$. Note that, for efficiency we only consider those patches which coarse matching sampled. To deeply integrate global and local features, $\tilde{F}_{tr}^T(\tilde{i})$ and $\tilde{F}_{tr}^{I_w}(\tilde{j})$ are upsampled and concatenated with corresponding local (fine-level) features $\hat{F}^T(\tilde{i})$ and $\hat{F}^{I_w}(\tilde{j})$, respectively.

Subsequently, the concatenated features are used as inputs to a 2-layer MLP to reduce the channel dimensionality to the same as for the original local features, yielding the fused features. The effectiveness of this module is demonstrated in Section 5.5.1.

For every patch pair $(\tilde{i}, \tilde{j})$, we then locate their all finer positions (i,j) where i lies on the edge. As fused feature maps, we crop two sets of local windows of size $w \times w$ centered at (i, j) respectively. A local transformer module operates N_{f} times within each window to generate the final features $(F^T(i), F^{I_{\mathrm{w}}}(j))$. Following Refs. [13, 49], the center vector of $F^T(i)$ is correlated with all vectors in $F^{I_{\mathrm{w}}}(j)$ resulting in a heatmap that represents the matching probability for each pixel centered on j with i. Using 2D softmax to compute expectation over the matching probability distribution, we get the final position j' with sub-pixel accuracy matching i. The final set of fine-level matches $\mathcal{M}_{\mathrm{f}}$ aggregates all matches (i, j').

4.4.2 Fine-level homography estimation

For each match (i, j') in $\mathcal{M}_{\mathrm{f}}$, we use the inverse transformation of H_{c} to warp j' to its original position on image I. After coarse-to-fine refinement, the correspondences are accurate without obvious outliers (see the last column of Fig. 5 later). We obtain the final homography H by wighted least squares using the DLT formulation, based on all matching pairs. The final homography H indicates the transformation from the template T to the source image I, precisely locating the template object.

4.4.3 Fine-level training losses

While training the fine-level module, the coarse-level module is fine-tuned at the same time. The training loss $\mathcal{L}$ is defined as $\mathcal{L} = \lambda\mathcal{L}_{\mathrm{c}} + \mathcal{L}_{\mathrm{f}}$. In $\mathcal{L}_{\mathrm{f}}$, we use ground-truth supervision and self-supervision together for better robustness. For ground-truth supervision, we use the weighted loss function from Ref. [49]. For self-supervision, we use $L2$ similarity loss [70, 71] to minimize the differences between local appearances of the warped image I_{w} and template T. $\mathcal{L}_{\mathrm{f}}$ is formulated as

$$\mathcal{L}_{\mathrm{f}} = \frac{1}{|\mathcal{M}_{\mathrm{f}}|} \sum_{(i,j')\in\mathcal{M}_{\mathrm{f}}} \frac{1}{\sigma^2(i)} \|j' - j'_{\mathrm{gt}}\| + \sum_{(i,j')\in\mathcal{M}_{\mathrm{f}}} (m_i * \|P_i^T - P_{j'}^{I_{\mathrm{w}}}\|)$$

where for each query point i, $\sigma^2(i)$ is the total variance of the corresponding heatmap, P_i^T denotes a local window cropped from template image T with i as the center, m_i is a local area mask indicating presence of an edge pixel. Experiments on $L2$ similarity loss are presented in Section 5.5.1.

5 Experiments

After introducing the datasets used in our experiments (Section 5.1) and implementation details (Section 5.2), estimated homographies are compared for our proposed method and baselines (Section 5.3). Applications of our approach in industrial lines are shown in Section 5.4, while Section 5.5 considers the effectiveness of the components of our strategy. Further experimental details can be found in the Appendices.

5.1 Datasets

Here we outline the datasets used for testing. Further details are given in Appendix A.3, to ensure reproducibility.

5.1.1 Mechanical Parts

Obtaining poses of industrial parts is essential for robotic manipulator grasping on automated industrial lines. We collected a dataset based on hundreds of varied planar mechanical parts. To enrich the dataset while avoiding laborious annotation, we used GauGAN to generate an extra 40k pairs of matching data for training. The test dataset consisting of 800 samples was collected from an industrial workshop with human-labeled ground truth. It was used to quantitatively evaluate our method for single template and single object scenes, and to visually demonstrate the application of our approach to multi-template and multi-object scenes in Section 5.4.

5.1.2 Assembly Holes

Locating and matching assembly holes can help determine whether the product parts have been machined in the correct position. Thus, we collected data for dozens of different assembly holes in vehicle battery boxes, giving about 45k image pairs. Each sample contains a binarized template image, a gray image to be matched, and a human-labeled mask. To simulate a real industry scenario, we randomly scaled the template size and perturbed the image corners to simulate possible hole deformation or camera disturbance. We randomly selected 700 image pairs containing all hole types for testing, and the remainder for training and validation.

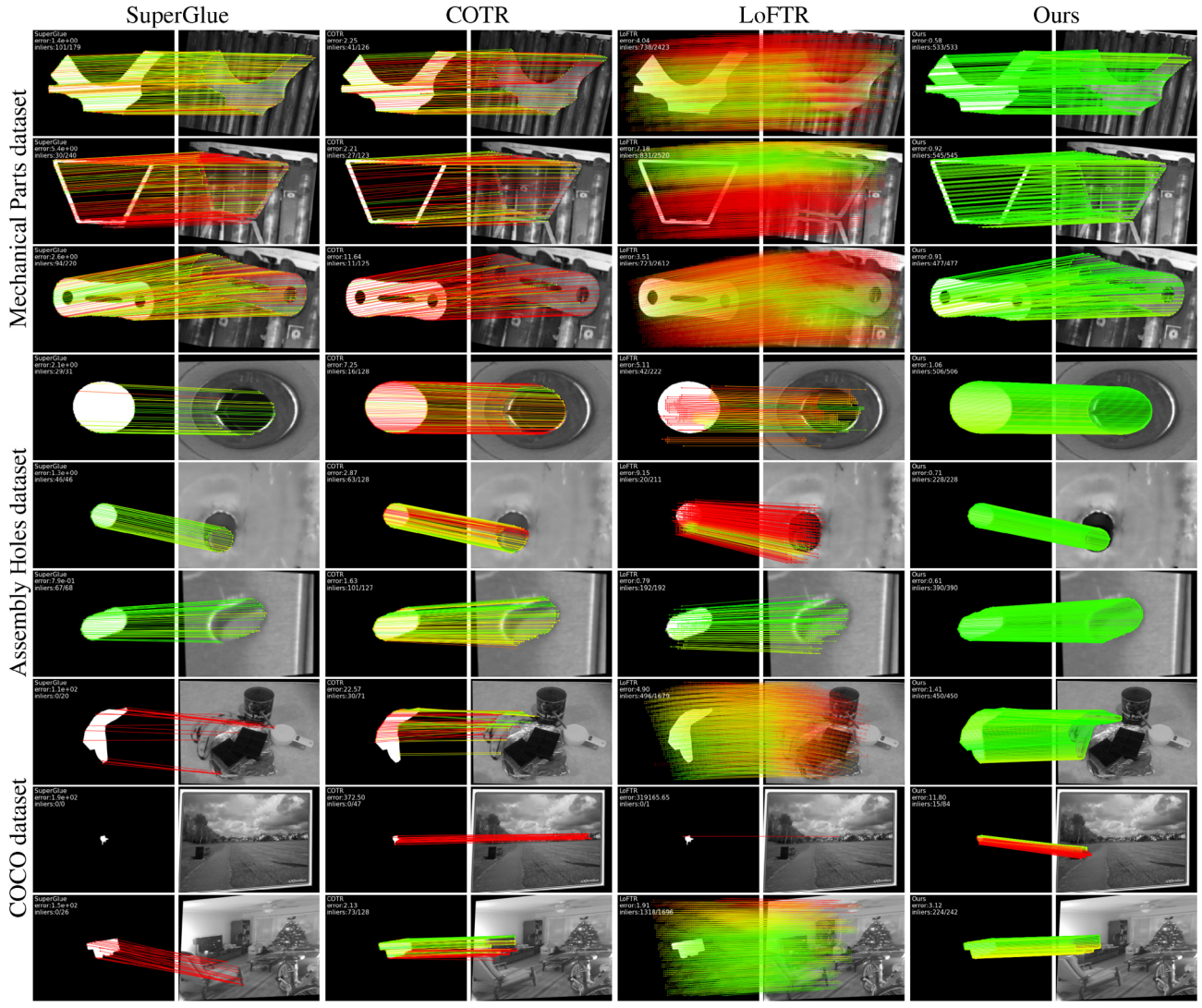

Fig. 5 Qualitative matching results for the three test datasets. Compared to SuperGlue, COTR, and LoFTR, our method consistently obtains a higher inlier ratio, successfully coping with large viewpoint change, small objects, and non-rigid deformation. Red indicates a reprojection error beyond 3 pixels for the Mechanical Parts and Assembly Holes datasets and 5 pixels for the COCO dataset. Further qualitative results can be found in the Availability of data and materials section.

5.1.3 COCO

Going beyond industrial scenarios, we also performed tests using the well-known computer vision dataset COCO [20] that contains common objects in natural scenes. Since COCO was not devised for template matching, we generated the image and template pair by selecting one instance mask and applying various kinds of transformations, including scaling, rotation, and corner perturbation. We randomly selected 50k and 500 images for training and testing from the COCO training and validation set, respectively.

5.2 Implementation details

For training and testing, all images were resized to 480×640. We use Kornia [72] for homography warping in the coarse alignment stage. Parameters were set as follows: window size $w = 8$, numbers of transformer layers: $N_c = 4$ and $N_f = 2$, match selection threshold $\sigma = 0.2$, loss weight $\lambda = 10$ is set to 10, maximum number of template patches $N_p = 128$, spatial consistency distance parameter $\sigma_d = 0.4$, angular consistency parameter $\sigma_\alpha = 1.0$, weight control parameter $= 0.5$, and the number of neighbors $k = 3$.

5.3 Evaluation

5.3.1 Evaluation metrics

Following Refs. [12, 13, 31], we compute the reprojection error of specific measurement points between the images warped with the estimated

Table 1 Homography estimation on the Mechanical Parts dataset. The AUC of the measurement point error is reported as a percentage. SuperGlue−− and SuperGlue use the pre-trained SuperPoint and our fine-tuned SuperPoint for keypoint detection, respectively

Category	Method	Homography est. AUC ↑		
		@3 px	@5 px	@10 px
Overall similarity measure	Linemod-2D	14.7	28.6	52.0
	GHT	1.7	3.5	6.5
Keypoints + MNN	SURF RANSAC	0.1	0.1	0.2
	SURF + MAGSAC	0.1	0.3	1.0
	D2Net + RANSAC	4.5	8.3	15.1
	D2Net + MAGSAC	20.6	36.5	58.2
	ASLFeat + RANSAC	7.5	14.3	26.5
	ASLFeat + MAGSAC	24.8	35.9	60.7
	SuperPoint + RANSAC	1.3	3.2	9.3
	SuperPoint + MAGSAC	12.0	25.7	50.1
Learning matchers	SuperGlue−− ∗ + RANSAC	18.4	35.2	60.3
	SuperGlue−− ∗ + MAGSAC	18.7	35.7	61.2
	SuperGlue + RANSAC	34.5	55.4	76.6
	SuperGlue + MAGSAC	32.2	53.3	75.5
	COTR + RANSAC	26.1	44.3	76.1
	COTR + MAGSAC	26.4	44.9	76.3
	LoFTR + RANSAC	40.0	60.9	80.0
	LoFTR + MAGSAC	40.6	61.4	80.2
	Ours	**58.8**	**74.7**	**87.3**

Table 2 Homography estimation on the Assembly Holes dataset. The AUC of the measurement point error is reported as a percentage. SuperGlue−− and SuperGlue use the pre-trained SuperPoint and our fine-tuned SuperPoint for keypoint detection, respectively

Category	Method	Homography est. AUC ↑		
		@3 px	@5 px	@10 px
Overall similarity measure	Linemod-2D	24.7	37.1	53.2
	GHT	18.7	31.2	49.3
Keypoints + MNN	SURF + RANSAC	0.2	0.5	2.0
	SURF + MAGSAC	0.8	2.1	7.5
	ORB + RANSAC	0.2	0.5	2.0
	ORB + MAGSAC	0.5	1.0	2.7
	D2Net + RANSAC	7.6	13.1	24.7
	D2Net + MAGSAC	19.9	31.8	49.1
	ASLFeat + RANSAC	16.4	28.2	40.3
	ASLFeat + MAGSAC	23.9	35.7	53.2
	SuperPoint + RANSAC	15.6	26.8	44.6
	SuperPoint + MAGSAC	17.2	31.1	52.0
Learning matchers	SuperGlue−− + RANSAC	15.1	26.2	43.6
	SuperGlue−− + MAGSAC	16.8	27.9	44.7
	SuperGlue + RANSAC	41.6	58.9	76.4
	SuperGlue + MAGSAC	41.5	58.9	76.3
	COTR + RANSAC	31.4	50.1	71.7
	COTR + MAGSAC	31.5	50.1	72.0
	LoFTR + RANSAC	54.3	68.8	81.8
	LoFTR + MAGSAC	54.3	68.7	81.8
	Ours	**69.1**	**81.0**	**90.4**

and the ground-truth homography. We then report the area under the cumulative curve (AUC) up to thresholds of [3, 5, 10] pixels for industrial datasets, and [5, 10, 20] pixels for the COCO dataset. To ensure a fair comparison, we sampled 20 points uniformly on each template boundary as measurement points for use throughout the experiments.

5.3.2 Baselines

We compared our method to three kinds of methods, based on: (i) overall similarity measure-based template matching, including Linemod-2D and generalized Hough transform (GHT), which are widely used for industrial scenes, (ii) keypoint detection with MNN search, including SURF, D2Net,

ASLFeat, and SuperPoint, and (iii) matching learning, including SuperGlue, COTR [11], and LoFTR (state-of-the-art feature matching networks).

For overall similarity measure-based methods which cannot deal with perspective transformation, we apply a more tolerant evaluation strategy. Specifically, we generate templates at multiple scales (step size = 0.01) and orientations (step size = 1°) for matching. We use the centroids of generated templates as measure points and select the template with the best score as the final result. For SURF, we use the PiDiNet edge detector to preprocess the input images. In SuperGlue, we choose SuperPoint for keypoint detection and descriptor extraction. All learning-based baselines were fine-tuned on each dataset until convergence, based on the parameters of the source model. Further details of training setup are provided in Appendix A.2.

We adopted RANSAC and MAGSAC for outlier rejection for all correspondence-based baselines when estimating the homography transformation, following Ref. [31]. Direct linear transformation (DLT) is applied directly in a differentiable manner to our method, assuming matches have high inlier rates and trustworthy confidence weights.

5.3.3 Qualitative comparison

We provide qualitative results in Figs. 5 and 6. In both figures, the first three rows use the Mechanical Parts dataset, the next three, the Assembly Holes dataset, and the last three, COCO. Figure 5 shows that, compared to SuperGlue, COTR, and LoFTR, the correspondences of our method are more accurate and reliable. While the correspondences predicted by SuperGlue and COTR, like ours, lie on the contour of the object, they contain more outliers. LoFTR yields more correspondences even in the blank

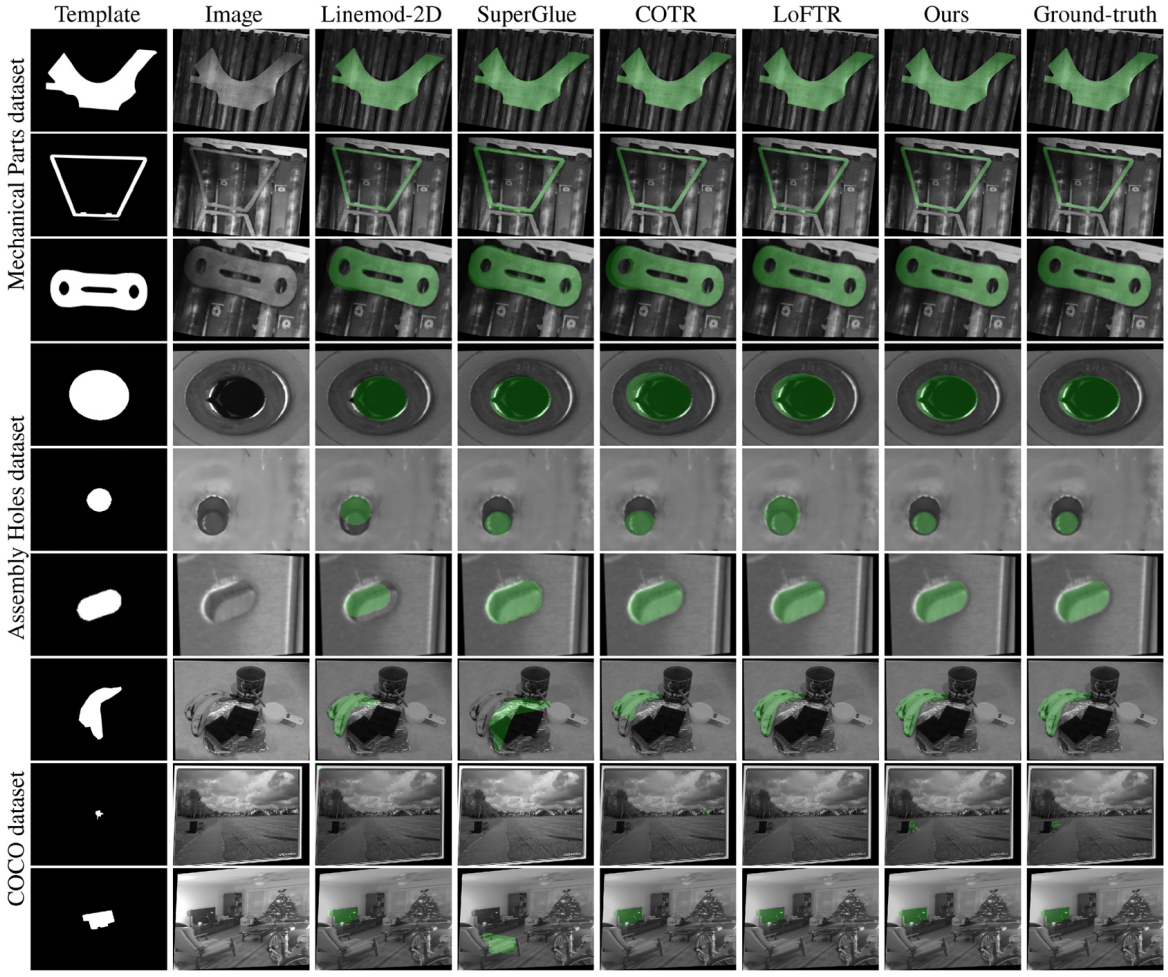

Fig. 6 Qualitative registration results for the three test datasets. The green area represents template mask placed in the input image using the estimated homography. For Linemod-2D, we selected the template with the best match from the set of templates. MAGASAC was used for outlier rejection for SuperGlue, COTR, and LoFTR.

area. However, these matching pairs tend to become inaccurate when further from the object. Instead, our method effectively uses contour information by focusing the matching points on the contour. With more correct matches and fewer mismatches, our approach does not need RANSAC or its variants for post-processing, which are essential for other methods. The second example from the COCO dataset demonstrates our method's superior ability to stably match small target objects.

In Fig. 6, we qualitatively compare our registration results to those of a classic template matching method, Linemod-2D, and three deep feature matching methods. Linemod-2D is susceptible to cluttered backgrounds. Learning-based matching baseline methods perform better but are prone to unstable results, especially for small objects. Our method produces a warped template with more pixels aligned in all these scenarios. Figure 7 shows that our approach provides much more accurate registration when examined in fine detail.

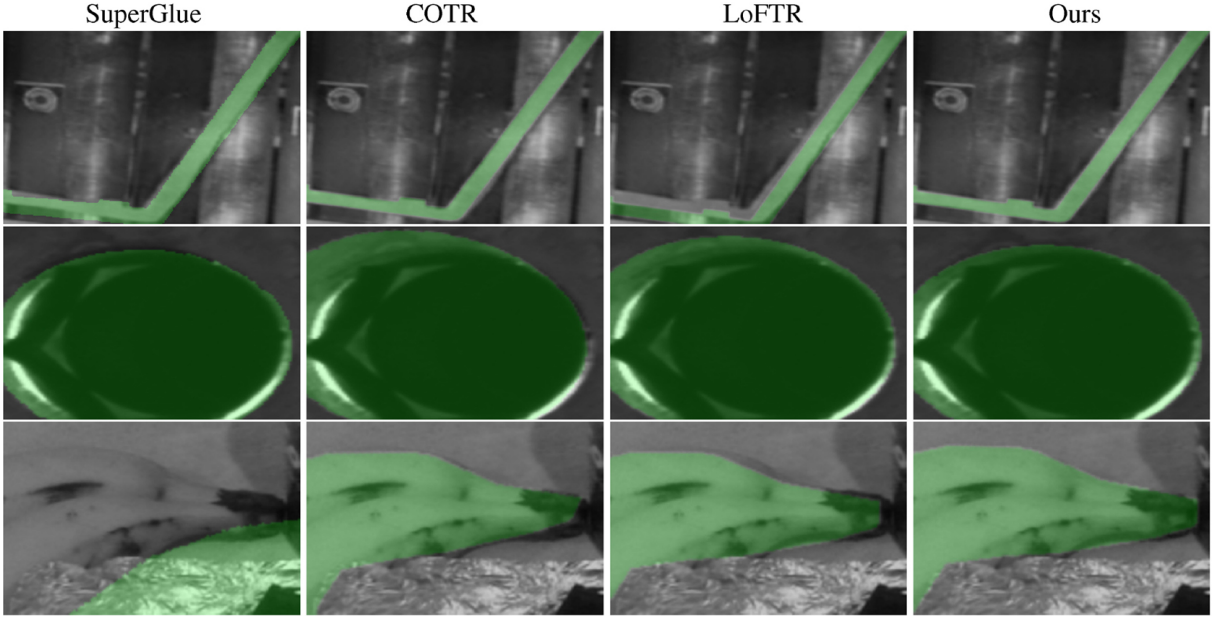

Fig. 7 Close-ups of registration results from SuperGlue, COTR, LoFTR, and our method. Our method accurately focuses on the contours of objects.

Table 3 Homography estimation on the CoCo dataset. The AUC of the measurement point error is reported as a percentage. SuperGlue−− and SuperGlue use the pre-trained SuperPoint and our fine-tuned SuperPoint for keypoint detection, respectively

Category	Method	Homography est. AUC ↑		
		@3 px	@5 px	@10 px
Overall similarity measure	Linemod-2D	26.2	47.5	64.2
	GHT	1.8	4.5	10.1
Keypoints + MNN	SURF + RANSAC	0.1	0.1	0.3
	SURF + MAGSAC	0.1	0.2	0.8
	D2Net + RANSAC	0.5	2.4	3.7
	D2Net + MAGSAC	1.3	3.5	7.2
	ASLFeat + RANSAC	1.3	3.4	7.6
	ASLFeat + MAGSAC	2.5	5.3	10.8
	SuperPoint + RANSAC	0.1	1.4	1.2
	SuperPoint + MAGSAC	0.5	1.8	4.4
Learning matchers	SuperGlue−− + RANSAC	2.7	6.5	11.9
	SuperGlue−− + MAGSAC	4.8	9.4	14.6
	SuperGlue + RANSAC	14.5	21.7	31.3
	SuperGlue + MAGSAC	14.7	22.2	32.1
	COTR + RANSAC	19.1	33.5	47.4
	COTR + MAGSAC	22.4	36.3	48.6
	LoFTR + RANSAC	26.9	47.2	62.8
	LoFTR + MAGSAC	28.0	48.5	64.0
	Ours	**32.4**	**51.5**	**66.2**

5.3.4 *Baselines using edge maps*

As extracting edges of input images may reduce the impact of modality differences on initial feature extraction, we performed further experiments on the Mechanical Parts dataset to evaluate competitive learning-based baseline methods using edge detection as pre-processing. For a fair comparison, we use PiDiNet to extract edge maps from the template and source images for all methods. Training settings remained the same as for the training process without edge extraction. As Fig. 8 shows, edge detection preprocessing worsens the results of these baseline methods, especially SuperGlue and COTR. We note that these methods tend to provide correspondences with lower accuracy for low-texture scenes, and edge detection results in images with little texture.

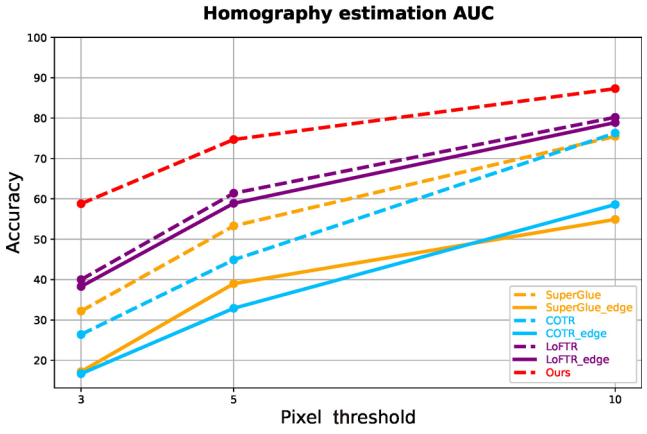

Fig. 8 Accuracy of various methods with (solid lines) and without (dashed lines) edge detection preprocessing of the input images. The homography estimation accuracy is reported for pixel thresholds [3, 5, 10].

5.4 Application

We now describe a challenging application of our method to real industrial lines, illustrated in Fig. 9. For each batch of industrial parts, the task is to select the correct template from a set of candidates templates for each part and to calculate its accurate pose. This is now an N-to-N template matching problem. We first pre-process the original scene using a real-time object detection network [73] to roughly locate each part and crop it into a separate image. For each candidate template, we first conduct coarse matching to select the optimal template: we use the correspondences with weighted confidences obtained by coarse matching to get an initial homography. Based on that homography, the template containing the most inlier correspondences is regarded as optimal. We then apply fine matching to accurately obtain the pose of the object using the selected optimal template.

To quantitatively evaluate our algorithm in multi-template scenarios, besides the correct template, we randomly add extra 9 noisy ones to the candidate template set. We tested 284 scenarios with 2445 test samples and achieved a recognition accuracy of 98.8%, when taking an inlier rate of more than 80% as correct recognition using the estimation matrix. In addition, our method runs at a competitive speed since we adopt the strategy of only using coarse matching for template selection. Further details of runtimes are presented in Appendix A.1.

We further note that our model generalizes well to

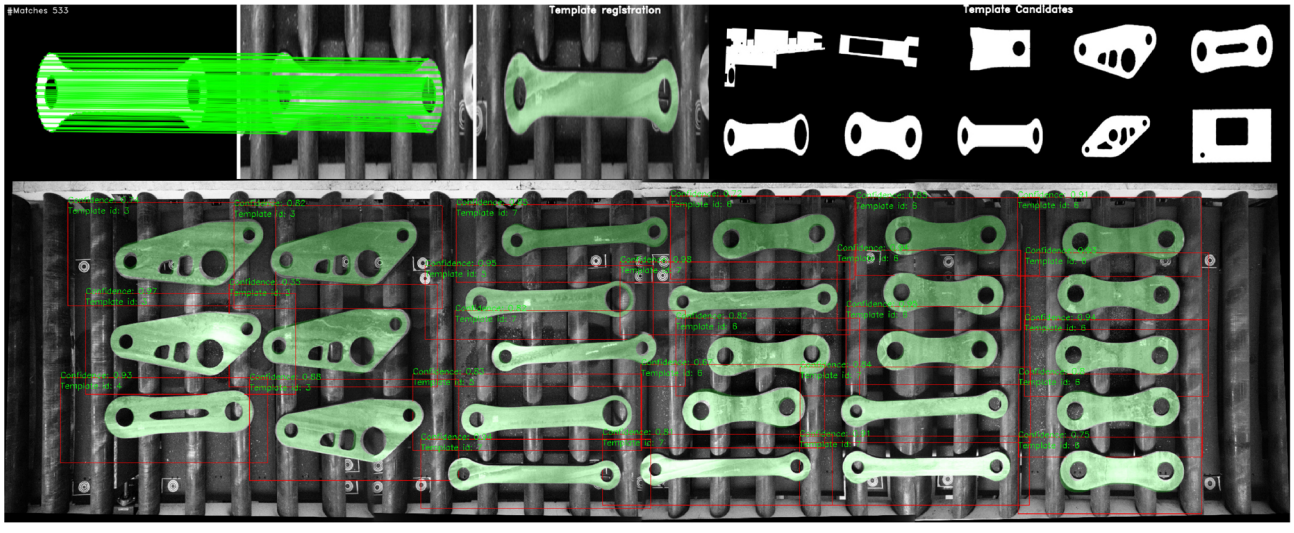

Fig. 9 Application of our method in an industrial line. Above left: best matching template for an object. Above right: set of candidate templates. Below: final matches to selected templates. The coarse matching inlier rate using every template is used as a basis for template selection.

unseen real scenarios after training only on synthetic data. We provide the link to video demonstrations in the Availability of data and materials section.

5.5 Analysis and discussion

5.5.1 Design study

To better understand our proposed method, we conducted seven comparative experiments on different modules, using the Mechanical Parts dataset. The quantitative results in Tables 4 and 5 validate our design decisions and show that they have a significant effect on performance. The choices considered are

- *Matching-layer: dual-softmax vs. optimal transport*. Both the dual-softmax operator and optimal transport achieve similar scores, and either would provide effective matching layers in our method.
- *Position-encoding: absolute vs. relative*. Replacing relative positional encoding by absolute positional encoding results in a significant drop in AUC. Relative position information is important in template matching.
- *Homography estimation: RANSAC vs. consistency confidence*. Since our method provides high-quality correspondences with confidence weights based on consistency, the differentiable DLT outperforms MAGSAC. An example is shown to demonstrate the advantages of DLT with consistency weights over RANSAC in Fig. 10. Inliers and outliers are explicitly distinguished by the RANSAC estimator, so correspondences with insufficient accuracy are directly discarded or fully adopted to estimate the final transformation matrix. Instead, our consistency module provides confidence weights, and we observe that the confidence weights estimated by the proposed method are consistent with ground-truth reprojection errors. Our method effectively assigns higher weights to more accurate correspondences and suppresses outliers. Therefore, in the case of high-quality matches, our consistency module can efficiently utilize correspondence information and so outperforms RANSAC.
- *Value & position*. Multiplying the value token by the positional embedding in the transformer module provides better results.
- *Translation module: Canny* [74] *vs. translation network*. Accuracy using the translation network is better than using Canny edge detection.
- *Feature fusion*. In the refinement stage, deep fusion of local features and global features leads to a noticeable performance improvement.
- *One stage vs. coarse-to-fine*. The coarse-to-fine module contributes to the estimation accuracy significantly by finding more matches and refining them to a sub-pixel level.
- *Self-supervision loss*. Using self-supervision loss ($L2$ similarity loss) brings a significant performance boost in fine-level training.
- *Maximum number of sample patches*. See Table 5. As the maximum number of samples based on the contour increases, accuracy of our method tends to improve. However, without sampling and using the entire template image as input, performance

Table 4 Evaluation of design choices using the Mechanical Parts dataset. Strategies marked with ⋆ are the ones adopted in our method

Design aspect	Method	Homography est. AUC ↑		
		@3 px	@5 px	@10 px
Matching-layer	Dual-Softmax	58.7	**74.7**	**87.3**
	Optimal transport ⋆	**58.8**	**74.7**	**87.3**
Position-encoding	Absolute	53.0	70.9	85.4
	Relative ⋆	**58.8**	**74.7**	**87.3**
Homography estimation	MAGSAC	53.8	71.3	85.6
	Consistency ⋆	**58.8**	**74.7**	**87.3**
Value & position	w/o position	57.9	74.1	86.9
	w position ⋆	**58.8**	**74.7**	**87.3**
Translation module	Canny	57.6	73.9	86.8
	Translation network ⋆	**58.8**	**74.7**	**87.3**
Feature fusion	Local feature	51.2	69.4	84.6
	Local–global feature fusion ⋆	**58.8**	**74.7**	**87.3**
One stage vs. coarse-to-fine	One stage	45.9	65.8	82.8
	Coarse-to-fine ⋆	**58.8**	**74.7**	**87.3**
Self-supervision loss	w/o self-supervision	55.0	72.2	86.0
	w self-supervision ⋆	**58.8**	**74.7**	**87.3**

Table 5 Effects on speed and accuracy of varying the number of patches sampled. ⋆ indicates the number used in our method

Number of patches	Homography est. AUC ↑			Runtime (ms) ↓
	@3 px	@5 px	@10 px	
8	22.0	40.9	66.0	99.3
16	37.7	58.0	78.6	99.8
32	50.5	69.0	84.2	107.6
64	57.4	73.7	86.8	112.2
⋆ 128	58.8	74.7	87.3	115.8
256	60.0	75.5	87.7	125.1
512	60.1	75.5	87.7	175.3
Unsampled	52.0	70.2	85.0	218.4

is somewhat lower than achieved by sampling with the best number of patches. We believe that edge-based sampling allows our method to more efficiently perceive the template structure and aggregate local features. We set the maximum number of patches to 128 as a trade-off between accuracy and runtime.

5.5.2 Understanding attention

To better understand the role of attention in our method, we visualize transformed features with t-SNE [75], and self- and cross-attention weights in Fig. 11. The visualization shows that our method learns a position-aware feature representation. The visualized attention weights reveal that the query point can aggregate global information dynamically and focus on meaningful locations. Self-attention may focus anywhere in the same image, especially regions with obvious differences, while cross-attention focuses on regions with a similar appearance in the other image.

5.5.3 Limitations and future work

Our method utilizes an existing edge detection network to eliminate the domain gap between templates and images, which is convenient for our approach. However, we believe that jointly training the translation network is a promising avenue for further improving performance. Another interesting follow-up is to design a one-to-many template matching algorithm that does not rely on any pre-processing.

6 Conclusions

We have presented a differentiable pipeline for accurate correspondence refinement for industrial template matching. With efficient feature extraction and feature aggregation by transformers, we obtain high-quality feature correspondences between the template mask and the grayscale image in a coarse-to-fine manner. The correspondences are then used to get a precise pose or transformation for the target object. To eliminate the domain gap between the template mask and grayscale image, we exploit a translation network. Based on the properties of the cross-modal template matching problem, we design a structure-aware strategy to improve robustness and efficiency. Furthermore, two valuable datasets from industrial scenarios have been collected, which we expect to benefit future work on industrial template matching. Our experiments show that our method significantly improves the accuracy and robustness of

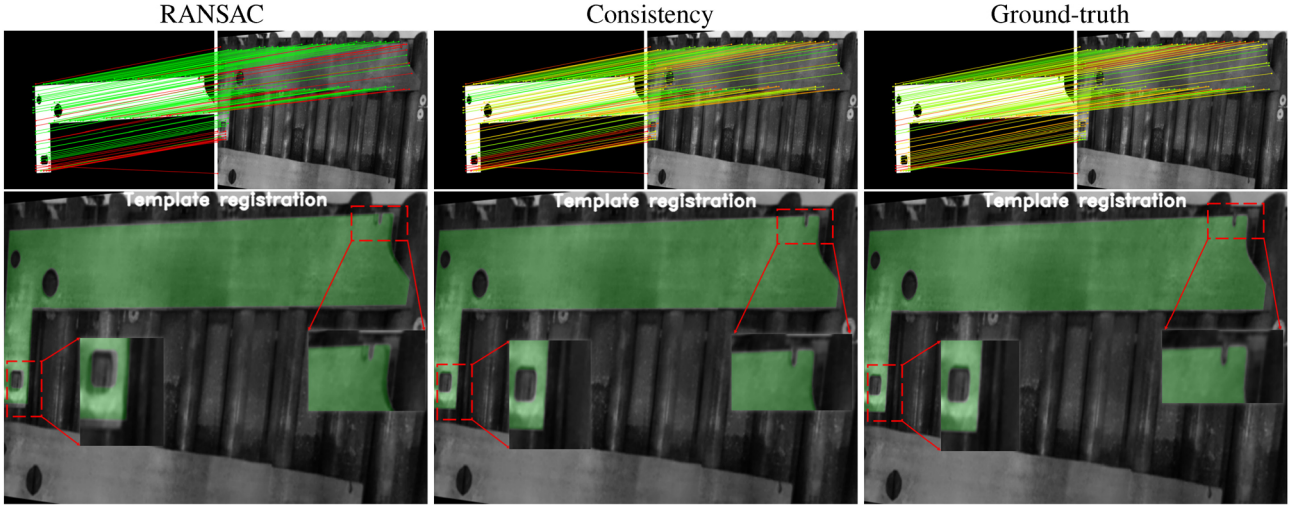

Fig. 10 Comparison of using RANSAC, or consistency, for homography estimation. Above: correspondences provided by coarse matching. Below: template registration results. Confidence is indicated by line colour from green (1) to red (0). In RANSAC, inliers have a confidence of 1, and outliers, 0. For the ground-truth, the reprojection error represents confidence.

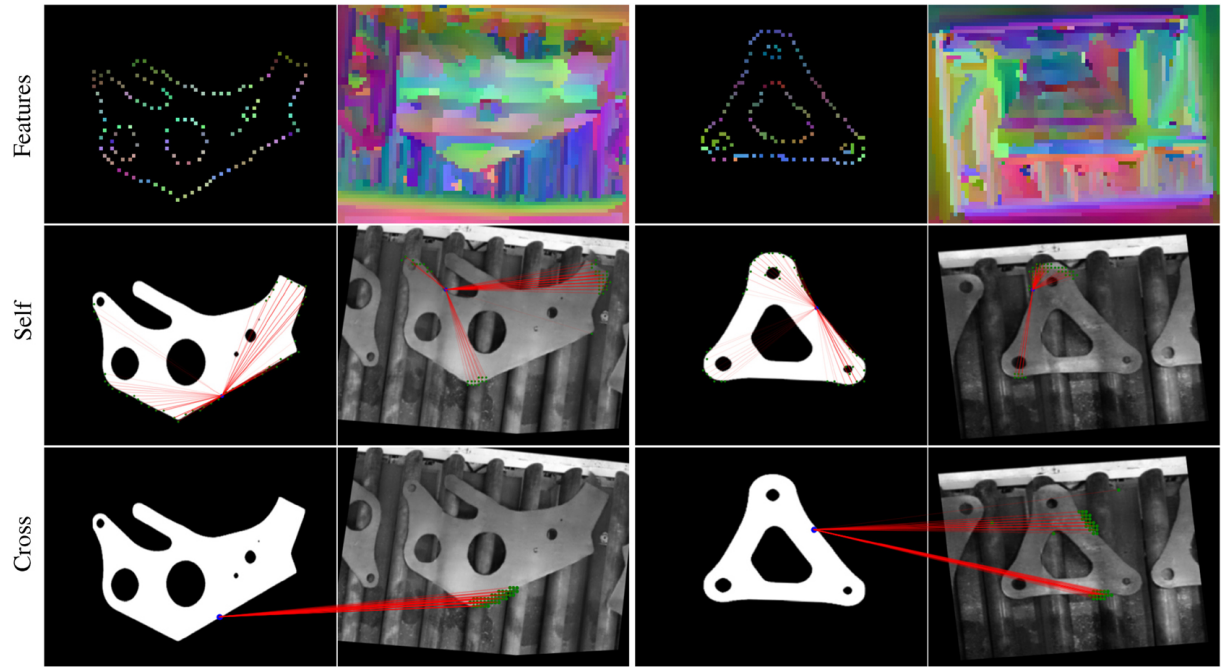

Fig. 11 Transformed features using t-SNE, and self- and cross-attention weights during coarse matching.

template matching relative to multiple state-of-the-art methods and baselines. Video demos of N-to-N template matching in real industrial lines show the effectiveness and good generalization of our method.

Appendix

A.1 Speed

We have tested the runtime of our method and other baselines on the Assembly Holes dataset, and report average values using an NVIDIA RTX 3080Ti. Coarse matching in our method takes 63 ms to match one pair; full matching takes 105 ms. LoFTR takes 87 ms, while COTR is much slower at 17 s. GHT and Linemod-2D take 4.9 and 2.2 s respectively: using multiple templates for different scales and poses is time-consuming. For a scene with 10 objects and 10 candidate templates, our method takes about 6.7 s to locate and identify all objects, and provide accurate poses.

A.2 Training details

Our network was trained on 2 NVIDIA RTX 3090 GPUs using a batch size of 16. Although end-to-end training is feasible, we found that a two-stage training strategy yielded better results. The first stage trained coarse-level matching using the loss term $\mathcal{L}_c$, until the validation loss converged. The second stage trained the whole pipeline using both $\mathcal{L}_c$ and $\mathcal{L}_f$ until the validation loss converged. Using the Mechanical Parts/Assembly Holes/COCO datasets, we trained our network for 15/15/30 epochs respectively for the first stage using Adam, with an initial rate of 10^{-3}, and 18/15/12 epochs for the second stage using Adam, with an initial rate of 10^{-4}. We loaded pre-trained weights for the translation network and local feature CNN provided by Refs. [33, 46], and fixed the local feature CNN parameters in the second stage.

We also loaded pre-trained parameters for other learning-based baseline methods and retrained them until the validation loss converged. Numbers of training epochs used for the different learning-based baseline methods are shown in Table 6 for each dataset. For better performance, for the keypoints methods (D2Net, ASLFeat, and SuperPoint), we only

Table 6 Number of training epochs used for different learning-based baseline methods, for the three datasets Mechanical Parts (MP), Assembly Holes (AH), and COCO. For SuperGlue−− and SuperGlue, we respectively used the pre-trained SuperPoint and our fine-tuned SuperPoint in the keypoint detection phase

Category	Method	MP	AH	COCO
Keypoint	D2Net	20	20	20
	ASLFeat	15	15	15
	SuperPoint	10	10	10
Matching	SuperGlue−−	30	30	30
	SuperGlue	10 + 30	10 + 30	10 + 30
	COTR	100	100	90
	LoFTR	12	12	35

used the edge points on the template to construct the ground-truth matching pairs when training the network. For COTR, we followed its three-stage training strategy to fine-tune the network. Since there is no recommended training method for SuperGlue, we trained it based on the code at `https://github.com/gouthamvgk/SuperGlue_training`.

A.3 Data and ground-truth generation

A.3.1 Mechanical Parts dataset

We used GauGAN to generate further image pairs for various shape parts. The manually adjusted image pairs provided by Linemod-2D served as training data for GauGAN. We discovered that GauGAN can learn well even from somewhat noisy data, and can provide high-quality ground-truth. Using arbitrarily distributed mask images (templates), we used GauGAN to generate synthetic industrial part images of size 480×640 for our network training. For each pair of generated images, we first moved the template mask to the center of the image, then randomly scaled the synthetic image by a factor in the range [0.8, 1.2] and rotated it through an angle in the range $[-15°, 15°]$.

A.3.2 Assembly Holes dataset

The ground-truth of the Assembly Holes dataset was annotated by humans: we segmented the outer circle of the part to give the mask for the image. The scaling range was [0.75, 1.25].

A.3.3 COCO dataset

We used the instance mask as the template T and the original image as the search image I for the COCO dataset [20]. We first filtered out masks near the image boundary because these masks tend to be the image background. Among the remaining masks, we chose the mask with the largest area as the template. The selected image pairs were resized to 480×640, the scaling range set to [0.9, 1.1], and the rotation range to $[-30°, 30°]$.

A.3.4 All datasets

To simulate possible object deformation or camera disturbance, we randomly perturbed the four corners of the image by values within the range $[-32, 32]$ pixels for all datasets.

Availability of data and materials

The well-known CoCo dataset is available from https://cocodataset.org/. Our two industrial datasets can be freely downloaded from https://drive.google.com/drive/folders/1Mu9QdnM5WsLccFp0Ygf7ES7mLV-64wRL?usp=sharing. Our code and video demos are available at https://github.com/zhirui-gao/Deep-Template-Matching.

Funding

This work is supported in part by the National Key R&D Program of China (2018AAA0102200) and the National Natural Science Foundation of China (62002375, 62002376, 62325221, 62132021).

Author contributions

Zhirui Gao: Methodology, Writing Draft, Visualization, Results Analysis; Renjiao Yi: Methodology, Supervision, Writing Draft, Results Analysis; Zheng Qin: Supervision, Results Analysis; Yunfan Ye: Supervision, Results Analysis; Chenyang Zhu: Methodology, Supervision; Kai Xu: Methodology, Supervision.

Acknowledgements

We thank Lintao Zheng and Jun Li for their help with dataset preparation and discussions.

Declaration of competing interest

The authors have no competing interests to declare that are relevant to the content of this article. The author Kai Xu is the Area Executive Editor of this journal.

References

[1] Hinterstoisser, S.; Cagniart, C.; Ilic, S.; Sturm, P.; Navab, N.; Fua, P.; Lepetit, V. Gradient response maps for real-time detection of textureless objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 34, No. 5, 876–888, 2012.

[2] Ballard, D. H. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition* Vol. 13, No. 2, 111–122, 1981.

[3] Muja, M.; Rusu, R. B.; Bradski, G.; Lowe, D. G. REIN – A fast, robust, scalable REcognition INfrastructure. In: Proceedings of the IEEE International Conference on Robotics and Automation, 2939–2946, 2011.

[4] Hinterstoisser, S.; Lepetit, V.; Ilic, S.; Fua, P.; Navab, N. Dominant orientation templates for real-time detection of texture-less objects. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2257–2264, 2010.

[5] Cheng, J. X.; Wu, Y.; AbdAlmageed, W.; Natarajan, P. QATM: Quality-aware template matching for deep learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 11545–11554, 2019.

[6] Gao, B.; Spratling, M. W. Robust template matching via hierarchical convolutional features from a shape biased CNN. In: *The International Conference on Image, Vision and Intelligent Systems. Lecture Notes in Electrical Engineering, Vol. 813*. Yao, J.; Xiao, Y.; You, P.; Sun, G. Eds. Springer Singapore, 333–344, 2022.

[7] Ren, Q.; Zheng, Y. B.; Sun, P.; Xu, W. Y.; Zhu, D.; Yang, D. X. A robust and accurate end-to-end template matching method based on the Siamese network. *IEEE Geoscience and Remote Sensing Letters* Vol. 19, Article No. 8015505, 2022.

[8] Wu, Y.; Abd-Almageed, W.; Natarajan, P. Deep matching and validation network: An end-to-end solution to constrained image splicing localization and detection. In: Proceedings of the 25th ACM international conference on Multimedia, 1480–1502, 2017.

[9] Rocco, I.; Arandjelovic, R.; Sivic, J. Convolutional neural network architecture for geometric matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 41, No. 11, 2553–2567, 2019.

[10] Efe, U.; Ince, K. G.; Aydin Alatan, A. DFM: A performance baseline for deep feature matching. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 4279–4288, 2021.

[11] Jiang, W.; Trulls, E.; Hosang, J.; Tagliasacchi, A.; Yi, K. M. COTR: Correspondence transformer for matching across images. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 6187–6197, 2021.

[12] Sarlin, P. E.; DeTone, D.; Malisiewicz, T.; Rabinovich, A. SuperGlue: Learning feature matching with graph neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 4937–4946, 2020.

[13] Sun, J. M.; Shen, Z. H.; Wang, Y. A.; Bao, H. J.; Zhou, X. W. LoFTR: Detector-free local feature matching with transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 8918–8927, 2021.

[14] Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł; Polosukhin, I. Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing System, 6000–6010, 2017.

[15] Wu, K.; Peng, H. W.; Chen, M. H.; Fu, J. L.; Chao, H. Y. Rethinking and improving relative position encoding for vision transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 10013–10021, 2021.

[16] Jaderberg, M.; Simonyan, K.; Zisserman, A.; Kavukcuoglu, K. Spatial transformer networks. In: Proceedings of the 28th International Conference on Neural Information Processing Systems, Vol. 2, 2017–2025, 2015.

[17] Fischler, M.; Bolles, R. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM* Vol. 24, No. 6, 381–395, 1981.

[18] Katharopoulos, A.; Vyas, A.; Pappas, N.; Fleuret, F. Transformers are RNNs: Fast autoregressive transformers with linear attention. In: Proceedings of the 37th International Conference on Machine Learning, 5156–5165, 2020.

[19] Park, T.; Liu, M. Y.; Wang, T. C.; Zhu, J. Y. Semantic image synthesis with spatially-adaptive normalization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2332–2341, 2019.

[20] Lin, T. Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C. L. Microsoft COCO: Common objects in context. In: *Computer Vision – ECCV 2014. Lecture Notes in Computer Science, Vol. 8693*. Fleet, D.; Pajdla, T.; Schiele, B.; Tuytelaars, T. Eds. Springer Cham, 740–755, 2014.

[21] Lowe, D. G. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* Vol. 60, No. 2, 91–110, 2004.

[22] Bay, H.; Tuytelaars, T.; Van Gool, L. SURF: Speeded up robust features. In: *Computer Vision – ECCV 2006. Lecture Notes in Computer Science, Vol. 3951*. Leonardis, A.; Bischof, H.; Pinz, A. Eds. Springer Berlin Heidelberg, 404–417, 2006.

[23] Rublee, E.; Rabaud, V.; Konolige, K.; Bradski, G. ORB: An efficient alternative to SIFT or SURF. In: Proceedings of the International Conference on Computer Vision, 2564–2571, 2011.

[24] Barath, D.; Matas, J.; Noskova, J. MAGSAC: Marginalizing sample consensus. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 10189–10197, 2019.

[25] Brachmann, E.; Rother, C. Neural-guided RANSAC: Learning where to sample model hypotheses. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 4321–4330, 2019.

[26] Brachmann, E.; Krull, A.; Nowozin, S.; Shotton, J.; Michel, F.; Gumhold, S.; Rother, C. DSAC—Differentiable RANSAC for camera localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2492–2500, 2017.

[27] Lucas, B. D.; Kanade, T. An iterative image registration technique with an application to stereo vision. In: Proceedings of the 7th International Joint Conference on Artificial Intelligence, 674–679, 1981.

[28] DeTone, D.; Malisiewicz, T.; Rabinovich, A. Deep image homography estimation. *arXiv preprint* arXiv:1606.03798, 2016.

[29] Hartley, R.; Zisserman, A. *Multiple View Geometry in Computer Vision*, 2nd edn. Cambridge, UK: Cambridge University Press, 2003.

[30] Nguyen, T.; Chen, S. W.; Shivakumar, S. S.; Taylor, C. J.; Kumar, V. Unsupervised deep homography: A fast and robust homography estimation model. *IEEE Robotics and Automation Letters* Vol. 3, No. 3, 2346–2353, 2018.

[31] Zhang, J. R.; Wang, C.; Liu, S. C.; Jia, L. P.; Ye, N. J.; Wang, J.; Zhou, J.; Sun, J. Content-aware unsupervised deep homography estimation. In: *Computer Vision – ECCV 2020. Lecture Notes in Computer Science, Vol. 12346*. Vedaldi, A.; Bischof, H.; Brox, T.; Frahm, J. M. Eds. Springer Cham, 653–669, 2020.

[32] Koguciuk, D.; Arani, E.; Zonooz, B. Perceptual loss for robust unsupervised homography estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 4269–4278, 2021.

[33] DeTone, D.; Malisiewicz, T.; Rabinovich, A. SuperPoint: Self-supervised interest point detection and description. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 337–33712, 2018.

[34] Dusmanu, M.; Rocco, I.; Pajdla, T.; Pollefeys, M.; Sivic, J.; Torii, A.; Sattler, T. D2-net: A trainable CNN for joint description and detection of local features. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 8084–8093, 2019.

[35] Yi, K. M.; Trulls, E.; Lepetit, V.; Fua, P. LIFT: Learned invariant feature transform. In: *Computer Vision – ECCV 2016. Lecture Notes in Computer Science, Vol. 9910*. Leibe, B.; Matas, J.; Sebe, N.; Welling, M. Eds. Springer Cham, 467–483, 2016.

[36] Luo, Z. X.; Zhou, L.; Bai, X. Y.; Chen, H. K.; Zhang, J. H.; Yao, Y.; Li, S. W.; Fang, T.; Quan, L. ASLFeat: Learning local features of accurate shape and localization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 6588–6597, 2020.

[37] Chen, H. K.; Luo, Z. X.; Zhang, J. H.; Zhou, L.; Bai, X. Y.; Hu, Z. Y.; Tai, C. L.; Quan, L. Learning to match features with seeded graph matching network. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 6281–6290, 2021.

[38] Jiang, B.; Sun, P. F.; Luo, B. GLMNet: Graph learning-matching convolutional networks for feature matching. *Pattern Recognition* Vol. 121, 108167, 2022.

[39] Shi, Y.; Cai, J. X.; Shavit, Y.; Mu, T. J.; Feng, W. S.; Zhang, K. ClusterGNN: Cluster-based coarse-to-fine graph neural network for efficient feature matching. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 12507–12516, 2022.

[40] Roessle, B.; Nießner, M. End2End multi-view feature matching with differentiable pose optimization. *arXiv preprint* arXiv:2205.01694, 2022.

[41] Suwanwimolkul, S.; Komorita, S. Efficient linear attention for fast and accurate keypoint matching. In: Proceedings of the International Conference on Multimedia Retrieval, 330–341, 2022.

[42] Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In: *Computer Vision – ECCV 2020. Lecture Notes in Computer Science, Vol. 12346*. Vedaldi, A.; Bischof, H.; Brox, T.; Frahm, J. M. Eds. Springer Cham, 213–229, 2020.

[43] Kitaev, N.; Kaiser, Ł; Levskaya, A. Reformer: The efficient transformer. *arXiv preprint* arXiv:2001.04451, 2020.

[44] Tay, Y.; Dehghani, M.; Bahri, D.; Metzler, D. Efficient transformers: A survey. *ACM Computing Surveys* Vol. 55, No. 6, 109, 2022.

[45] Lan, Y. Q.; Duan, Y.; Liu, C. Y.; Zhu, C. Y.; Xiong, Y. S.; Huang, H.; Xu, K. ARM3D: Attention-based relation module for indoor 3D object detection. *Computational Visual Media* Vol. 8, No. 3, 395–414, 2022.

[46] Su, Z.; Liu, W. Z.; Yu, Z. T.; Hu, D. W.; Liao, Q.; Tian, Q.; Pietikäinen, M.; Liu, L. Pixel difference networks for efficient edge detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 5097–5107, 2021.

[47] Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint* arXiv:1409.1556, 2014.

[48] Jau, Y. Y.; Zhu, R.; Su, H.; Chandraker, M. Deep keypoint-based camera pose estimation with geometric constraints. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 4950–4957, 2020.

[49] Wang, Q.; Zhou, X.; Hariharan, B.; Snavely, N. Learning feature descriptors using camera pose supervision. In: *Computer Vision – ECCV 2020. Lecture Notes in Computer Science, Vol. 12346*. Vedaldi, A.; Bischof, H.; Brox, T.; Frahm, J. M. Eds. Springer Cham, 757–774, 2020.

[50] Zhou, Q.; Agostinho, S.; Ošep, A.; Leal-Taixé, L. Is geometry enough for matching in visual localization? In: *Computer Vision – ECCV 2022. Lecture Notes in Computer Science, Vol. 13670*. Avidan, S.; Brostow, G.; Cissé, M.; Farinella, G. M.; Hassner, T. Eds. Springer Cham, 407–425, 2022.

[51] Qi, C. R.; Yi, L.; Su, H.; Guibas, L. J. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, 5105–5114, 2017.

[52] Li, Y.; Harada, T. Lepard: Learning partial point cloud matching in rigid and deformable scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 5544–5554, 2022.

[53] Su, J. L.; Lu, Y.; Pan, S. F.; Murtadha, A.; Wen, B.; Liu, Y. F. RoFormer: Enhanced transformer with rotary position embedding. *arXiv preprint* arXiv:2104.09864, 2021.

[54] Cuturi, M. Sinkhorn distances: Lightspeed computation of optimal transport. In: Proceedings of the 26th International Conference on Neural Information Processing Systems, Vol. 2, 2292–2300, 2013.

[55] Rocco, I.; Cimpoi, M.; Arandjelovi? R.; Torii, A.; Pajdla, T.; Sivic, J. Neighbourhood consensus networks. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems, 1658–1669, 2018.

[56] Tyszkiewicz, M. J.; Fua, P.; Trulls, E. DISK: Learning local features with policy gradient. In: Proceedings of the 34th International Conference on Neural Information Processing Systems, 14254–14265, 2020.

[57] Barath, D.; Matas, J. Graph-cut ransac. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 6733–6741, 2018.

[58] Chum, O.; Matas, J.; Kittler, J. Locally optimized RANSAC. In: *Pattern Recognition. Lecture Notes in Computer Science, Vol. 2781*. Michaelis, B.; Krell, G. Eds. Springer Berlin Heidelberg, 236–243, 2003.

[59] Leordeanu, M.; Hebert, M. A spectral technique for correspondence problems using pairwise constraints. In: Proceedings of the 10th IEEE International Conference on Computer Vision, 1482–1489, 2005.

[60] Bai, X. Y.; Luo, Z. X.; Zhou, L.; Chen, H. K.; Li, L.; Hu, Z. Y.; Fu, H. B.; Tai, C. L. PointDSC: Robust point cloud registration using deep spatial consistency. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 15854–15864, 2021.

[61] Chen, Z.; Sun, K.; Yang, F.; Tao, W. B. SC2-PCR: A second order spatial compatibility for efficient and robust point cloud registration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 13211–13221, 2022.

[62] Quan, S. W.; Yang, J. Q. Compatibility-guided sampling consensus for 3-D point cloud registration. *IEEE Transactions on Geoscience and Remote Sensing* Vol. 58, No. 10, 7380–7392, 2020.

[63] Yang, J. Q.; Xian, K.; Wang, P.; Zhang, Y. N. A performance evaluation of correspondence grouping methods for 3D rigid data matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 43, No. 6, 1859–1874, 2021.

[64] Qin, Z.; Yu, H.; Wang, C.; Guo, Y. L.; Peng, Y. X.; Xu, K. Geometric transformer for fast and robust point cloud registration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 11133–11142, 2022.

[65] Mises, R. V.; Pollaczek-Geiringer, H. Praktische verfahren der gleichungsauflösung. *ZAMM - Zeitschrift Für Angewandte Mathematik Und Mechanik* Vol. 9, No. 1, 58–77, 1929.

[66] Mok, T. C. W.; Chung, A. C. S. Affine medical image registration with coarse-to-fine vision transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 20803–20812, 2022.

[67] Parihar, U. S.; Gujarathi, A.; Mehta, K.; Tourani, S.; Garg, S.; Milford, M.; Krishna K. M. RoRD: Rotation-robust descriptors and orthographic views for local feature matching. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 1593–1600, 2021.

[68] Shen, X.; Darmon, F.; Efros, A. A.; Aubry, M. RANSAC-flow: Generic two-stage image alignment. In: *Computer Vision – ECCV 2020. Lecture Notes in Computer Science, Vol. 12349*. Vedaldi, A.; Bischof, H.; Brox, T.; Frahm, J. M. Eds. Springer Cham, 618–637, 2020.

[69] Truong, P.; Danelljan, M.; Timofte, R. GLU-net: Global-local universal network for dense flow and

correspondences. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 6257–6267, 2020.

[70] Lee, M. C. H.; Oktay, O.; Schuh, A.; Schaap, M.; Glocker, B. Image-and-spatial transformer networks for structure-guided image registration. In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019. Lecture Notes in Computer Science, Vol. 11765*. Shen, D., et al. Eds. Springer Cham, 337–345, 2019.

[71] Shu, C.; Chen, X.; Xie, Q. W.; Han, H. An unsupervised network for fast microscopic image registration. In: Proceedings of the SPIE 10581, Medical Imaging 2018: Digital Pathology, 105811D, 2018.

[72] Riba, E.; Mishkin, D.; Ponsa, D.; Rublee, E.; Bradski, G. Kornia: An open source differentiable computer vision library for PyTorch. In: Proceedings of the IEEE Winter Conference on Applications of Computer Vision, 3663–3672, 2020.

[73] Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 779–788, 2016.

[74] Canny, J. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. PAMI-8, No. 6, 679–698, 1986.

[75] Van der Maaten, L.; Hinton, G. Visualizing data using t-SNE. *Journal of Machine Learning Research* Vol. 9, No. 86, 2579–2605, 2008.

Zhirui Gao received his B.E. degree in computer science and technology from the Chinese University of Geosciences, Wuhan, in 2021. He is now a master student at the National University of Defense Technology (NUDT). His research interests include image matching and 3D vision.

Renjiao Yi is an assistant professor in the School of Computing, NUDT. She is interested in 3D vision problems such as inverse rendering and image-based relighting.

Zheng Qin received his B.E. and M.E. degrees in computer science and technology from NUDT in 2016 and 2018, respectively, where he is currently pursuing a Ph.D. degree. His research interests focus on 3D vision, including point cloud registration, pose estimation, and 3D representation learning.

Yunfan Ye is a Ph.D. candidate in the School of Computing, NUDT. His research interests include computer vision and graphics.

Chenyang Zhu is an assistant professor in the School of Computing, NUDT. His current directions of interest include data-driven shape analysis and modeling, 3D vision, robot perception, and robot navigation.

Kai Xu is a professor in the School of Computing, NUDT, where he received his Ph.D. degree in 2011. He serves on the editorial board of *ACM Transactions on Graphics*, *Computer Graphics Forum*, *Computers & Graphics*, etc.