Supplementary for Shape-guided Configuration-aware Learning for Endoscopic-image-based Pose Estimation of Flexible Robotic Instruments

Yiyao Ma*[1], Kai Chen*[1], Hon-Sing Tong[2], Ruofeng Wei[1], Yui-Lun Ng[2], Ka-Wai Kwok†[2,3,4], and Qi Dou†[1]

[1] Department of Computer Science and Engineering, The Chinese University of Hong Kong
[2] Agilis Robotics Limited
[3] Department of Mechanical Engineering, The University of Hong Kong
[4] Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong

Overview

In this supplementary material, we provide additional contents that are not included in the main paper due to the space limit:

- A. Details of Configuration-aware Shape Deformation.
- B. Depth maps from pre-trained DPT.
- C. Details of keypoint-based baseline.
- D. Details of skeleton-based baseline.
- E. Implementation Details.
- F. Comparison with different pose representations.
- G. Pose prediction with uncertainty.
- H. More qualitative comparisons with baselines.
- I. General effectiveness for pose refinement.

A. Details of Configuration-aware Shape Deformation.

In the main paper, we claim that we can parameterize each control point $\mathbf{c_i^{b_j}}$ within part_j in the base coordinate system. In this section, we provide a more detailed explanation of how the coordinates of each segment's skeleton curve are parameterized specifically for the flexible robot employed in this paper. Let N_S, N_P, and N_D represent the number of control points for segments O_rS, SP, and PD, respectively. We can parameterize the control points relative to (α, β, γ):

$$\mathbf{c_{i_s}^{r_j}} = (x_{i_s}, y_{i_s}, z_{i_s}) = (\frac{l_S}{N_S} i_s, 0, \frac{l_S}{N_S} i_s \tan\gamma), \tag{1}$$

* indicates equal contributions.
† indicates corresponding authors.

$$\mathbf{c}_{\mathbf{i_p}}^{\mathbf{r_j}} = (x_{i_p}, y_{i_p}, z_{i_p}) = (l_S + \frac{l_P}{\alpha}\sin\alpha_{i_p}, \frac{2l_P}{\alpha}\sin^2(\frac{\alpha_{i_p}}{2}), \frac{l_P}{\alpha}\sin\alpha_{i_p}\tan\gamma), \tag{2}$$

$$\mathbf{c}_{\mathbf{i_d}}^{\mathbf{r_j}} = (x_{i_d}, y_{i_d}, z_{i_d}) \tag{3}$$

$$= R_P \cdot (\frac{l_D}{\beta}\sin\beta_{i_d}, -\frac{2l_D}{\beta}\sin^2(\frac{\beta_{i_d}}{2}), \frac{l_D}{\beta}\sin\beta_{i_d}\tan\gamma) + T_P, \tag{4}$$

where i_s, i_p, i_d are the index of points within N_S, N_P, N_D, respectively. For simplicity, we omit the superscript r_j in the above equation. $\alpha_{i_p} \sim U(0, \alpha)$, $\beta_{i_d} \sim U(0, \beta)$ are sampled from uniform distribution. R_P and T_P represent the rotation matrix and translation vector from the coordinate determined by P to base coordinate:

$$R_P = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0, \\ 0 & 0 & 1 \end{bmatrix}, T_P = (l_S + \frac{l_P}{\alpha}\sin\alpha, \frac{2l_P}{\alpha}\sin^2(\frac{\alpha}{2}), 0)^T. \tag{5}$$

After that, each point on skeleton will to be rotated by ($\mathbf{e}$, δ) based on Rodrigues' rotation formula, which has been already discussed in our main paper.

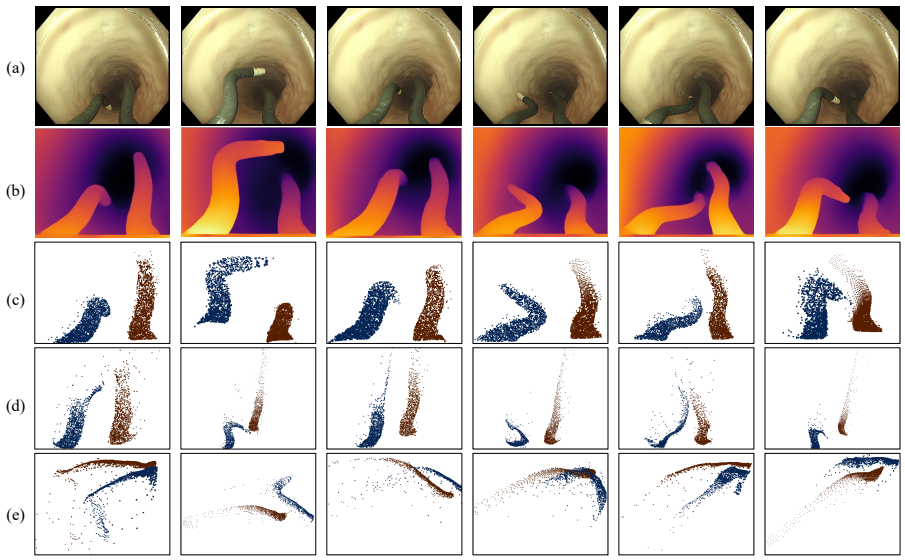

Fig. 1: Illustration of the depth map and reconstructed point cloud. (a) is the RGB input; (b) is the depth map predicted by DPT [6]; (c)(d)(e) are the front view, top view, and left view of reconstructed point cloud.

B. Depth map from pre-trained DPT

DPT leverages vision transformers as a backbone for dense depth prediction. The input image is first transformed into tokens by extracting non-overlapping patches followed by a linear projection of their flattened representation. Afterwards, the tokens from various stages of the vision transformer are assembled into image-like representations at various resolutions and then progressively combined into full-resolution final depth map using a convolutional decoder. DPT is pre-trained by a large dataset MIX and has better generalizability for monocular depth estimation [6]. We directly use the pre-trained DPT to predict the depth maps of flexible robots. As shown in Fig. 1, although the pre-trained model can predict the depth map with rich boundaries and details, the recovered flexible robot point cloud still suffer severe shape distortions and have relatively large noise, which limit the pose accuracy.

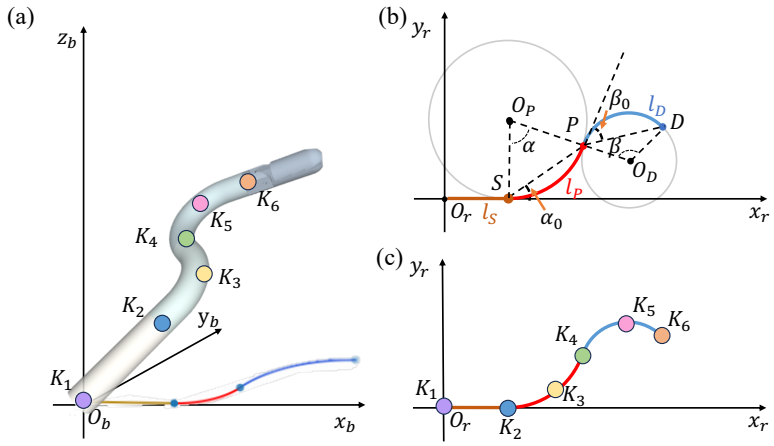

Fig. 2: Illustration of pre-defined keypoints. (a) represents the keypoints in 3D space; (b)(c) exhibit the keypoints in $x_r o_r y_r$ space.

C. Details of keypoint-based baseline

Fig. 2 shows the pre-defined six keypoints on the flexible robot body. K_1, K_2, K_4, K_6 are on the boundary of each robot part, K_3 and K_5 are on the middle of the arc SP and arc PD. For each keypoint $K_i = (u_i, v_i)$, we can compute its 3D coordinate in the image coordinate system with the known camera intrinsic parameters and the depth value d_i:

$$[x_i, y_i, z_i] = [\frac{u_i - c_x}{f_x} \times d_i, \frac{v_i - c_y}{f_y} \times d_i, d_i],$$

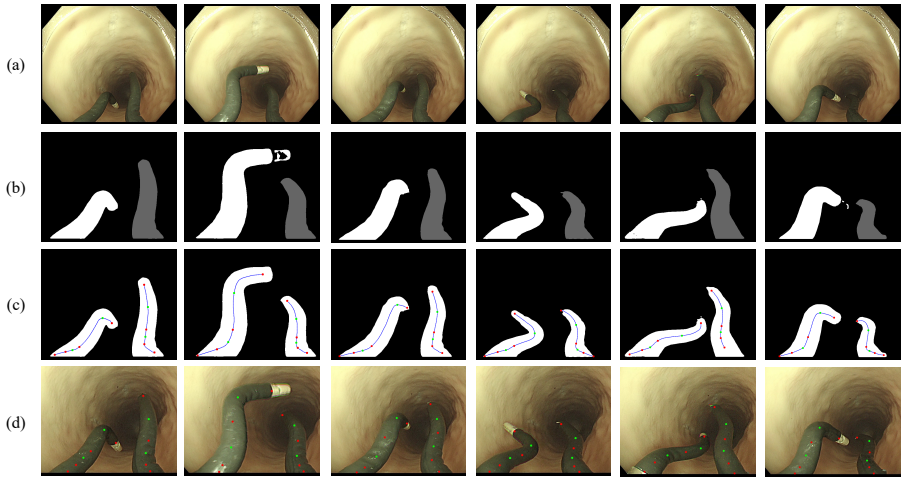

Fig. 3: Illustration of intermediate results of baselines. (a) is the RGB input; (b) is the robot mask; (c) is the extracted skeleton line; (d) is the 2D keypoints tracked by [10].

where f_x, f_y are camera focal length along x and y axes, c_x, c_y are the coordinates of the principal point, all of them can be obtained by offline calibration.

Without loss of generality, we suppose that in image coordinate system, the x_b-axis points to the right, the y_b-axis points downward, and the z_b-axis points away from the camera (towards the scene). We denote $\mathbf{v_{K_iK_j}}$ as the vector from point K_i to point K_j. Therefore, the value of γ is determined by the angle between the plane $y_b o_b z_b$ and the vector $\mathbf{v_{K_1K_2}}$. By utilizing the $3D$ coordinates of the above keypoints, we can establish a robot plane $x_r o_r y_r$, allowing us to derive δ by computing the angle between $x_r o_r y_r$ and $y_b o_b z_b$. Moreover, the central angle α is twice the size of α_0, which represents the angle between $\mathbf{v_{K_2K_4}}$ and $\mathbf{v_{K_1K_2}}$. Similarly, β is twice the size of β_0, which can be determined by the tangent line T_{K_4} at point K_4 and the vector $\mathbf{v_{K_4K_6}}$. Utilizing the values of l_S, l_P, α, γ, and δ, we can calculate the coordinates of the center O_P, thereby obtaining T_{K_4} by determining the line perpendicular to $\mathbf{v_{O_PK_4}}$ and passing through point K_4.

Since our data is derived from continuous image sequences, we manually label keypoints on the first frame, and then employ a tracking method [10] to track these keypoints consistently across subsequent frames. To maintain accuracy, we periodically reposition the keypoints every 100 frames.

D. Details of skeleton-based baseline

To extract a flexible robot skeleton from an image, we first identify the largest contour within the binary mask. Subsequently, we utilize the fast skeletonization method to generate the skeleton line. However, given the large shape variance of the flexible robot, the resulting skeleton may contain noticeable noise. To address

Table 1: Comparison of different pose representations on G_3.

Methods	α		β		γ		δ			
	Mean↓	Med.↓	Mean↓	Med.↓	Mean↓	Med.↓	Mean↓	Med.↓	5° ↑	10° ↑
Euler Angle	10.72	7.73	14.30	14.96	3.20	**0.93**	101.70	87.18	0	0
Quaternion	10.72	7.45	7.01	5.75	2.57	1.82	23.21	15.18	23.1	38.8
Rot6d [12]	**8.62**	5.94	**4.64**	**3.68**	**1.67**	1.33	16.23	8.13	31.4	58.6
PoseEst. (*ours*)	8.76	**4.78**	4.79	3.90	2.05	1.58	**14.58**	**5.94**	**45.9**	**59.5**

this, we employ a Cubic-Bezier curve fitting approach to refine the skeleton line. With the fitted curve parameters, we can determine the turning and boundary points, which are instrumental in subsequent pose calculations.

E. Implementation Details

We utilized off-the-shelf segmentation algorithms [3, 5] to crop the image region of the flexible robot arm. Specifically, we applied the Segment-Anything-Model (SAM) [3] to assist in annotating the ground truth masks of the instruments. These masks were subsequently used to train a lightweight AttU-Net [5] model for efficient robot arm segmentation. Then, the image patch was resized to 224×224 before it was fed into the image encoder for image feature extraction. For the shape prior of the flexible robot arm, we first uniformly sampled $N = 1024$ points for the arm. Furthermore, we sampled $M = 64$ center points via the farthest point sampling algorithm. For each center point, we collected its $K = 32$ nearest neighbors for point cloud tokenization. The point cloud tokens were fed into 12 sequential Transformer encoder blocks for shape feature extraction. We further enhanced the image-based flexible robot representation with the shape guidance via a single layer Transformer with 128 hidden dimensions and 4 self-attention heads. During training, we adopted data augmentation of color jittering, random Gaussian noises, and background randomization to enhance the model robustness. We use Adam optimizer [2] with a base learning rate of 2×10^{-4}, annealed at 60% of the training epoch using a cosine schedule. The total training epoch is set to 100 with a batch size of 24.

F. Comparison with different pose representations

In the main paper, we discussed our adoption of the matrix Fisher distribution [1] to construct a probabilistic model for representing rotation matrices and improving pose estimation. While other representations such as Euler angles, quaternions, or rot6d [12] can also be used as regression targets, they have limitations that affect their performance. As demonstrated in Table 1, Euler angles and quaternions produce unsatisfactory results due to their discontinuity, making it challenging for neural networks to learn effectively. For example, in rotation space, values like 0.1 and $2\pi - 0.1$ are very close, yet the loss in Euler angles

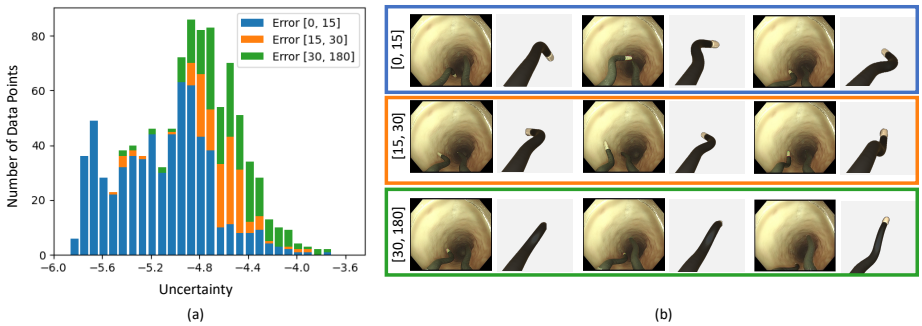

Fig. 4: Indication ability of uncertainty with model performance. (a) The x axis is the uncertainty value, and the y axis is the number of data points. Different colors represent different error ranges, in degree. (b) Qualitative results of the data with different error ranges and uncertainty values. This result comes from the model w / o shape guidance for predicting δ.

can be significantly large. On the other hand, rot6d performs comparably to the matrix Fisher distribution by addressing the issue of rotation discontinuity. However, the matrix Fisher distribution offers the advantage of providing both the pose estimation and the reliance of the prediction, which will be demonstrated in Section G. This additional information can potentially be valuable for informing subsequent robot manipulation tasks.

G. Pose Prediction with Uncertainty

In the main paper, we highlighted the significance of the estimated pose as a critical visual feedback for robot control. However, relying solely on the pose result without considering its uncertainty can lead to an incomplete understanding of the reliability of the pose estimation. Therefore, accounting for uncertainty is vital in close-loop control systems, as it enables informed decision-making and enhances the overall control performance.

In this subsection, we showcase how the utilization of matrix Fisher distribution as a pose representation enables us to extract the uncertainty associated with pose predictions. Following [11], we employ entropy as a metric for quantifying uncertainty. Lower uncertainty values typically indicate a more focused distribution with higher prediction confidence. Fig. 4 illustrates the correlation between prediction errors and uncertainty values obtained from the model w / o shape guidance when estimating δ. Notably, the results suggest that data with greater uncertainty are more likely to have larger errors, verifying that the uncertainty can be an effective indicator to reflect the pose quality.

H. More qualitative comparisons with baselines

In the main paper, we provided a qualitative comparison with a skeleton-based method. This visualization was obtained by first deforming the robot CAD model

Fig. 5: Qualitative comparison with more baselines. (a) is input RGB image; (b) exhibits the extracted robot skeleton; (c)(d)(e)(f)(g)(h) represent the pose estimation results from SKL [8], KP [4], SimPS [7], DR [9], PoseEst. (ours), and PoseRefine. (ours). For concise illustration, only the result for the left arm is shown.

with the estimated poses and then projecting the deformed model onto 2D image using the camera extrinsic and intrinsic parameters. In this section, we extend the qualitative comparisons to various baselines, including SKL [8], KP [4], SimPS [7], and DR [9]. As shown in Fig. 5, our methods, PoseEst. and PoseRefine., demonstrate superior performance in estimating the flexible robot pose that closely aligns with the input image.

I. General effectiveness for pose refinement

In this experiment, we studied the general effectiveness of the proposed pose refinement model in different scenarios. We evaluated the model in two different scenarios. First, we used it to refine the pose prediction from other baseline methods. Fig. 6 depicts the experimental results on top of the two regression-based baselines. As we can see, the proposed refinement module significantly reduces the prediction error for all pose parameters. Second, we take the pose prediction from the previous frame as the initial robot pose for the current frame, and then use the pose refinement module to refine the prediction for the current frame. It is a workflow similar to robot pose tracking. Fig. 7 presents the experimental results. The pose refinement model is very robust. It can significantly improve the average accuracy as well as the prediction robustness within the whole sequence.

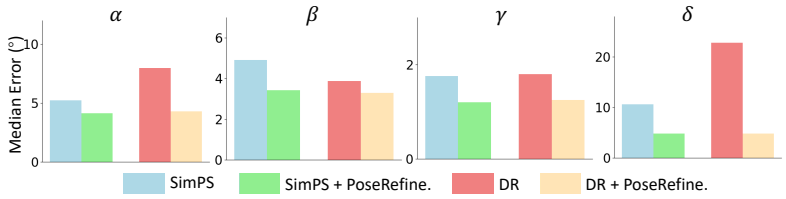

Fig. 6: Effectiveness of pose refinement. We compare the median error of the estimation with or without PoseRefine.

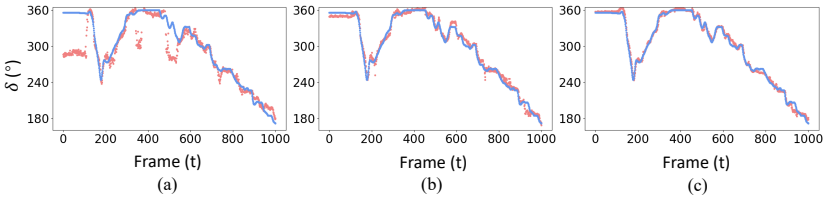

Fig. 7: Comparative results of flexible robot pose tracking. (a) Model w/o shape guidance: averaged error 16.74°. (b) Model with PoseEst.: averaged error 4.77°. (c) Model with PoseRefine.: averaged error 3.47°. Blue and red points represent ground truth and model predictions, respectively.

References

1. Khatri, C., Mardia, K.V.: The von mises–fisher matrix distribution in orientation statistics. Journal of the Royal Statistical Society Series B: Statistical Methodology **39**(1), 95–106 (1977)
2. Kingma, D.P., Ba, J.L.: Adam: Amethod for stochastic optimization. In: Proc. 3rd Int. Conf. Learn. Representations. pp. 1–15 (2014)
3. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., Dollar, P., Girshick, R.: Segment anything. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 4015–4026 (October 2023)
4. Lee, T.E., Tremblay, J., To, T., Cheng, J., Mosier, T., Kroemer, O., Fox, D., Birchfield, S.: Camera-to-robot pose estimation from a single image. In: 2020 IEEE International Conference on Robotics and Automation (ICRA). pp. 9426–9432. IEEE (2020)
5. Oktay, O., Schlemper, J., Le Folgoc, L., Lee, M., Heinrich, M., Misawa, K., Mori, K., McDonagh, S., Hammerla, N.Y., Kainz, B., et al.: Attention u-net: Learning where to look for the pancreas. In: Medical Imaging with Deep Learning (2022)
6. Ranftl, R., Bochkovskiy, A., Koltun, V.: Vision transformers for dense prediction. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 12179–12188 (2021)
7. Souipas, S., Nguyen, A., Laws, S.G., Davies, B.L., y Baena, F.R.: Simps-net: Simultaneous pose & segmentation network of surgical tools. IEEE Transactions on Medical Robotics and Bionics (2023)

8. Tanaka, K., Minami, Y., Tokudome, Y., Inoue, K., Kuniyoshi, Y., Nakajima, K.: Continuum-body-pose estimation from partial sensor information using recurrent neural networks. IEEE Robotics and Automation Letters **7**(4), 11244–11251 (2022)
9. Valassakis, E., Dreczkowski, K., Johns, E.: Learning eye-in-hand camera calibration from a single image. In: Conference on Robot Learning. pp. 1336–1346. PMLR (2022)
10. Yang, J., Gao, M., Li, Z., Gao, S., Wang, F., Zheng, F.: Track anything: Segment anything meets videos. arXiv preprint arXiv:2304.11968 (2023)
11. Yin, Y., Cai, Y., Wang, H., Chen, B.: Fishermatch: Semi-supervised rotation regression via entropy-based filtering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11164–11173 (2022)
12. Zhou, Y., Barnes, C., Lu, J., Yang, J., Li, H.: On the continuity of rotation representations in neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5745–5753 (2019)