3D Congealing: 3D-Aware Image Alignment in the Wild

– Supplementary Materials –

Yunzhi Zhang[1], Zizhang Li[1], Amit Raj[2], Andreas Engelhardt[3], Yuanzhen Li[2], Tingbo Hou[4], Jiajun Wu[1], and Varun Jampani[5]

[1] Stanford University
[2] Google DeepMind
[3] University of Tübingen
[4] Meta GenAI
[5] Stability AI

1 Additional Qualitative Results

The complete set of input images used for the sculpture dataset from Figure 1 (main text) and the corresponding results are shown in Figure 1. Images come from a personal photo collection.

2 Implementation Details

Feature Extractors. We use the ViT-G/14 variant of DINO-V2 [6] as the feature extractor f_ζ from Sec 3.2 and extract tokens from its final layer for all quantitative experiments. For qualitative results from Figure 1, following [8], we use features from the first upsampling block of the UNet from Stable Diffusion 2.1 with diffusion timestep 261 as f_ζ, as these features are similar for semantically-similar regions [3, 8, 9] but are locally smoother compared to DINO, which is consistent with the observations from [9].

Smoothness Loss. The smoothness loss from Eq. (10) is specified as follows. Following [5], we define

$$\mathcal{L}_{\text{rigidity},\nabla}(T) := \|J_\nabla(T)^T J_\nabla(T)\|_F + \|(J_\nabla(T)^T J_\nabla(T))^{-1}\|_F, \tag{1}$$

where T jointly considers all neighboring coordinates u, instead of only one coordinate u at once. We define $[T]_u = \tilde{u} - u$, where $\tilde{u}$ is the optimization variable for input u from Eq. (10), and $J_\nabla(T)$ computes the Jacobian matrix of T approximated with finite differences with pixel offset ∇. Following [7], we denote huber loss with $\mathcal{L}_{\text{huber}}$ and define the total variation loss as

$$\mathcal{L}_{\text{TV}}(T) = \mathcal{L}_{\text{huber}}(\nabla_x T) + \mathcal{L}_{\text{huber}}(\nabla_y T), \tag{2}$$

where ∇_x and ∇_y are partial derivatives *w.r.t.* x and y coordinates, approximated with finite differences. The final smoothness loss is defined as

$$\mathcal{L}_{\text{smooth}} = \lambda_{\text{rigidity},\nabla=10} + 0.1\mathcal{L}_{\text{rigidity},\nabla=1} + 10\mathcal{L}_{\text{TV}}. \tag{3}$$

3 Feature Visualizations

Matching features independently for each pixel gives noisy similarity heatmaps (Figure 3 (b)), due to the noise of feature maps (Figure 3 (d-e)), and the lack of geometric reasoning in the matching process. Our method is robust to such noises as it seeks to align the input with a posed rendering considering all pixel locations in the input altogether.

4 Semantic Correspondence Matching

We provide additional quantitative evaluation of our method on the task of semantic correspondence matching. Given a pair of source and target image $(x_{\text{source}}, x_{\text{target}})$, and given a keypoint in the source image u_{source}, the goal of this task is to find its most semantically similar keypoint u_{target} in the target image.

The matching process using our method is specified as follows. We first map the 2D keypoints being queried to the 3D coordinates in the canonical space, and then project these 3D coordinates to the 2D image space of the target image. Formally, given an image pair $(x_{\text{source}}, x_{\text{target}})$ and a 2D keypoint u_{source}, the corresponding keypoint u_{target} is computed with

$$u_{\text{target}} = \Phi^{\text{rev}}_{x_{\text{target}}} \circ \Phi^{\text{fwd}}_{x_{\text{source}}}(u_{\text{source}}), \tag{4}$$

with notations defined in Sec. 3.3.

For all experiments in this section, for Eq. (10), we set $\lambda_{\ell_2} = 10$ and for simplicity set $\lambda_{\text{smooth}} = 0$.

Dataset. We use SPair-71k [4], a standard benchmark for semantic correspondence matching for evaluation. We evaluate our method on 9 rigid, non-cylindrical-symmetric categories from this dataset. The images for each category may contain a large diversity in object shape, texture, and environmental illumination.

Following prior works [5, 7], we report the Percentage of Correct Keypoints (PCK@α) with $\alpha = 0.1$, a standard metric that evaluates the percentage of keypoints correctly transferred from the source image to the target image with a threshold α. A predicted keypoint is correct if it lies within the radius of $\alpha \cdot \max(H_{\text{bbox}}, W_{\text{bbox}})$ of the ground truth keypoint in the object bounding box in the target image with size $H_{\text{bbox}} \times W_{\text{bbox}}$.

Baselines. We compare with a 2D-correspondence matching baseline. Formally, for this baseline, for each querying keypoint u_{query}, we compute the keypoint prediction with

$$u_{\text{target}} = \arg\min_{u} d^{u,u}_{\zeta}(x_{\text{target}}, x_{\text{source}}), \tag{5}$$

where the distance metric d_ζ is defined in Eq. (6) and is induced from a pre-trained features extractor f_ζ. We use the same DINO feature extractor for our method and this baseline.

We further compare with previous congealing methods, GANgealing [7], which uses pre-trained GAN for supervision, and Neural Congealing [5] and ASIC [1], which are both self-supervised.

	Aero	Bike	Boat	Bus	Car	Chair	Motor	Train	TV	Mean
GANgealing. [7]	-	37.5	-	-	-	-	-			
Neural Congealing [5]	-	29.1	-	-	-	-	-	-	-	-
ASIC [1]	57.9	25.2	24.7	28.4	30.9	21.6	26.2	49.0	24.6	32.1
DINOv2-ViT-G/14 [6]	**72.5**	67.0	**45.5**	54.6	53.5	40.7	71.8	**53.5**	36.3	55.0
Ours	70.0	**70.3**	40.0	**65.8**	**72.1**	**50.1**	**77.0**	26.1	**43.1**	**57.2**

Table 1: Semantic Correspondence Evaluation on SPair-71k [4]. Our method achieves an overall better keypoint transfer accuracy compared to prior 2D congealing methods and a 2D-matching baseline using the same semantic feature extractor as ours.

Results. Results are shown in Tab. 1. The performance gain over the DINOv2 baseline, which uses the same semantic feature extractor backbone as ours, suggests the effectiveness of 3D geometric consistency utilized by our framework.

Qualitative results are shown in Figure 2. Our method is the only one that performs correspondence matching via reasoning in 3D among all baselines. Such 3D reasoning offers an advantage especially when the relative rotation between the objects from the source and target image is large. Our method transforms the 3D coordinate from source to target in the canonical frame, where the 3D shape guarantees the 3D consistency. In comparison, as shown on the right of Figure 2, the baseline performs 2D matching and incorrectly matches the front of a plane with its rear, and incorrectly matches the front wheel of a bicycle with its back wheel.

5 Other Image Editing Tasks

Given an input image, in Figure 4 we show results on rendering the template under a novel view, which is the image's assigned pose shifted by 30° in azimuth, and show texture transfer results below. We use Zero-1-to-3 [2] as the texture source for novel views, which itself does not guarantee 3D consistency in the view synthesis result but serves as a proxy source for the pixel values of novel views.

6 Failure Modes

We have identified two failure modes of the proposed method: (1) incorrect shapes from the generative model distillation process, *e.g.*, the incorrect placement of the water gun handle from Figure 5 (a), and (2) incorrect poses due to feature ambiguity, *e.g.*, the pumpkin is symmetric and DINO features cannot disambiguate sides from Figure 5 (b).

References

1. Gupta, K., Jampani, V., Esteves, C., Shrivastava, A., Makadia, A., Snavely, N., Kar, A.: Asic: Aligning sparse in-the-wild image collections. arXiv preprint arXiv:2303.16201 (2023)

2. Liu, R., Wu, R., Van Hoorick, B., Tokmakov, P., Zakharov, S., Vondrick, C.: Zero-1-to-3: Zero-shot one image to 3d object. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9298–9309 (2023)
3. Luo, G., Dunlap, L., Park, D.H., Holynski, A., Darrell, T.: Diffusion hyperfeatures: Searching through time and space for semantic correspondence. NeurIPS **36** (2024)
4. Min, J., Lee, J., Ponce, J., Cho, M.: Spair-71k: A large-scale benchmark for semantic correspondence. arXiv preprint arXiv:1908.10543 (2019)
5. Ofri-Amar, D., Geyer, M., Kasten, Y., Dekel, T.: Neural congealing: Aligning images to a joint semantic atlas. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19403–19412 (2023)
6. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: Dinov2: Learning robust visual features without supervision. arXiv preprint arXiv:2304.07193 (2023)
7. Peebles, W., Zhu, J.Y., Zhang, R., Torralba, A., Efros, A.A., Shechtman, E.: Gan-supervised dense visual alignment. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13470–13481 (2022)
8. Tang, L., Jia, M., Wang, Q., Phoo, C.P., Hariharan, B.: Emergent correspondence from image diffusion. arXiv preprint arXiv:2306.03881 (2023)
9. Zhang, J., Herrmann, C., Hur, J., Cabrera, L.P., Jampani, V., Sun, D., Yang, M.H.: A tale of two features: Stable diffusion complements dino for zero-shot semantic correspondence. arXiv preprint arXiv:2305.15347 (2023)

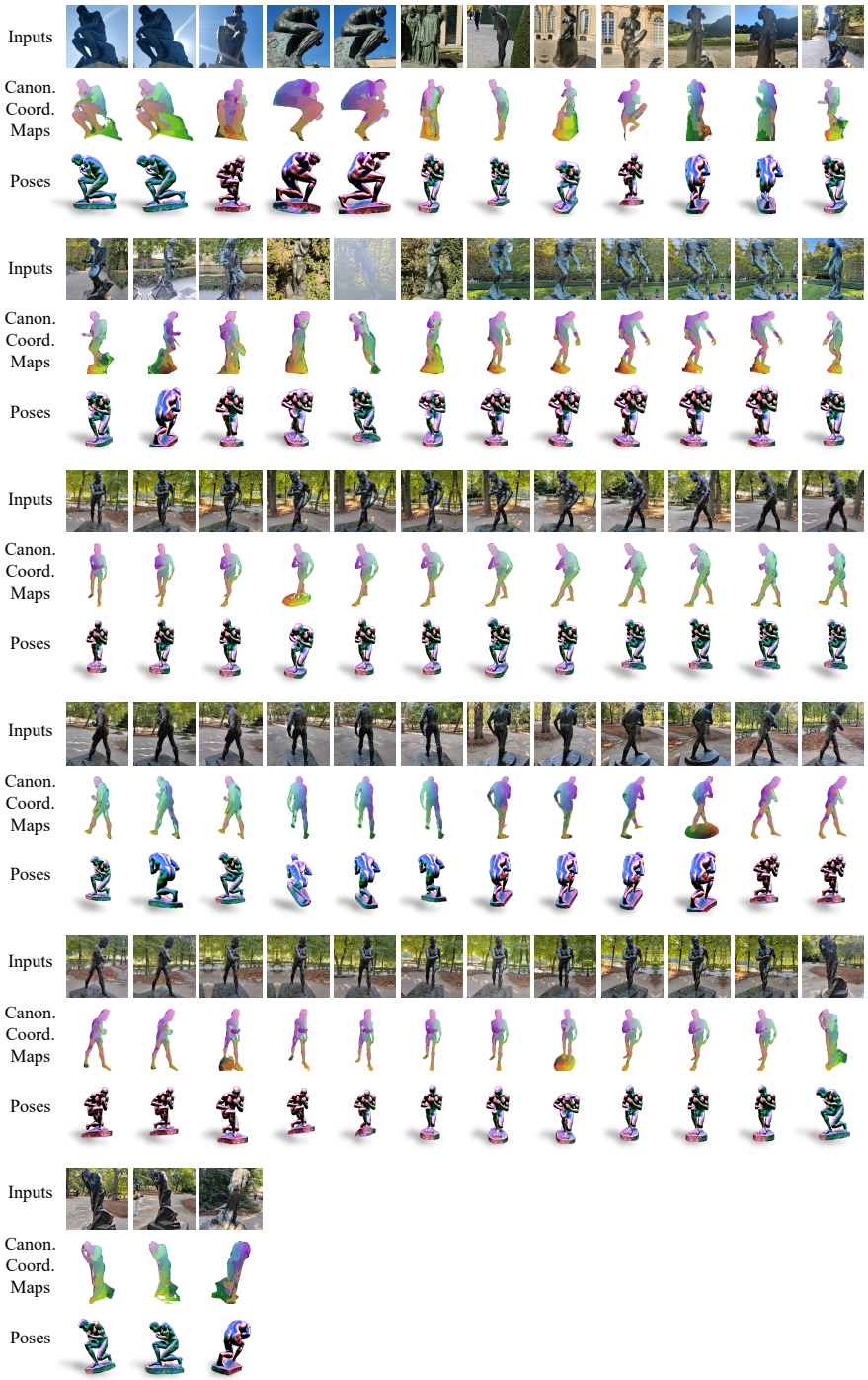


Fig. 1: Results on the Sculpture Dataset.

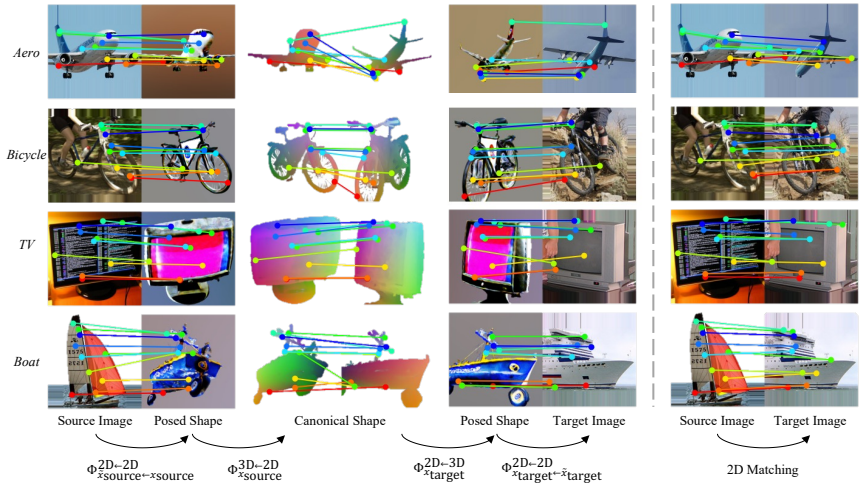


Fig. 2: Semantic Correspondence Matching. The figure shows results on 4 example categories from SPair-71k [4]. To match a given keypoint from the source image, our method first warps the keypoint to the rendered image space (2D-to-2D), then identifies the warped coordinate's location in the canonical frame in 3D (2D-to-3D), then projects the *same* 3D location to the rendering corresponding to the target image (3D-to-2D), and finally warps the obtained coordinate to the target image space (2D-to-2D). The learned 3D canonical shape serves as an intermediate representation that aligns the source and target images, and it better handles scenarios when the viewpoint changes significantly compared to matching features in 2D.

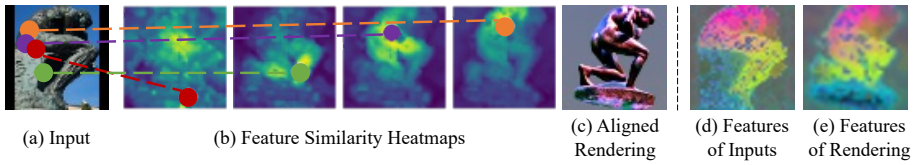


Fig. 3: Feature Visualizations. Despite that DINO features tend to be noisy, our approach assigns a plausible pose to the input, as shown in the aligned rendering.

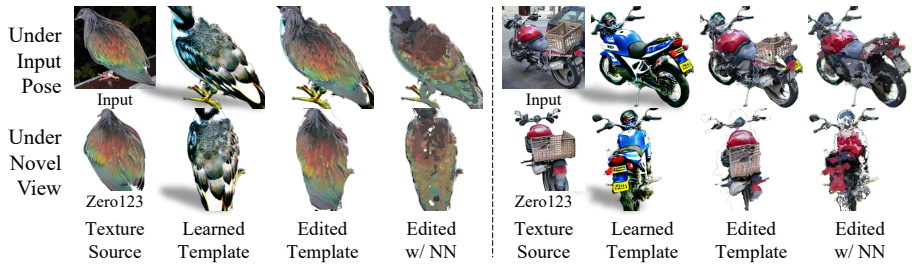
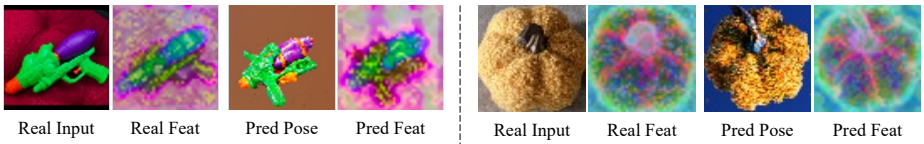


Fig. 4: Texture Propagation under Novel Views. Our method can be used in conjunction with Zero-1-to-3 to render a shape template under a viewpoint different from the input image with source texture from the input.



(a) Incorrect Canonical Shape (b) Ambiguous Features

Fig. 5: Failure Modes. Our method inherits the failure from (a) canonical shape optimization and (b) pre-trained feature extractors.