DIFFUSION 3D FEATURES (DIFF3F)
Decorating Untextured Shapes with Distilled Semantic Features

Supplementary Material

Niladri Shekhar Dutt[1,2] Sanjeev Muralikrishnan[1] Niloy J. Mitra[1,3]
[1]University College London [2]Ready Player Me [3]Adobe Research
https://diff3f.github.io/

1. Implementation Details

Rendering. The multi-view rendering is implemented using PyTorch 3D [7] with Hard Phong [6] shading and appropriate lighting. We place the light source in the direction of the camera to ensure coherent lighting across views and set a bin size of 1 to capture enough details during rasterization. To render the shape from all angles, we vary the elevation (θ) and azimuthal (ϕ) angles in linearly spaced intervals between 0°-360°.

Diffusion. We use simple text prompts such as 'human' for SHREC'19 or animal name for TOSCA and SHREC'20 (e.g., 'dog', 'cat', etc.) to guide the diffusion model to produce textured renderings. All text prompts are appended with 'best quality, highly detailed, photorealistic' to serve as positive prompt. We further use the negative prompt, 'lowres, low quality, monochrome' to prevent the diffusion model from producing poor quality textures.

Point Clouds. When applying our method on point clouds, DIFF3F requires point clouds to have enough density to produce a (mostly) smooth and continuous depth map to accurately condition the diffusion model. For the SHREC'19 dataset, this would require about 8000 points. We replace normal maps ($\mathcal{D}$) in G from Equation 3 with Canny edge maps ($\mathcal{E}$) since accurate estimation of normals from a point cloud can be challenging. We compute the edge map from the depth map instead of the point cloud render because the depth map is smooth and hence less noisy.

$$G := \{\mathcal{E}(I_j^S), \mathcal{D}(I_j^S)\}, \tag{13}$$

Code. Our implementation can be found at https://github.com/niladridutt/Diffusion-3D-Features

Evaluation. To have a shared baseline and maintain parity with previous works [1, 4, 9], we downsample the original point cloud to 1024 points by random sampling. For our method, we use the complete mesh for rendering and compute descriptors for only 1024 points during the unprojection process.

2. Semantic Correspondence

To showcase the semantic nature of DIFF3F descriptors, we present heatmap visualizations in our project webpage. For a query point in the source, we see semantically similar regions being highlighted in the target shape. For example, in the teddy → table pair, we see the legs of the teddy correspond with the legs of the table. We additionally showcase correspondence results on highly non-isometric pairs such as octopus → ant. We see that the head of the octopus maps to the head and tail of the ant, whereas the three legs of the octopus map to the two antennas and six legs of the ant.

3. Evaluation on SHREC'07

We evaluate DIFF3F and baseline methods on the SHREC Watertight 2007 [2] dataset to show performance on diverse shape classes outside commonly tested human and animal shapes. The dataset comprises of 400 3D shapes of 20 classes spanning glasses, fish, plane, plier, etc. with an average of 15 annotated correspondence points for each class. We skip all non human and four legged animals as these classes have been already evaluated (SHREC'19, TOSCA, SHREC'20). We additionally do not evaluate on spring and vase as these shapes lack annotations. The resulting dataset comprises of 320 shapes and we pair shapes from the same class to form 3040 test pairs, considerably larger than previous evaluations. Similar to previous evaluations, we choose DPC [4] and SE-ORNet [1] models trained on the SURREAL [3] dataset as it contains the largest number of training shapes. The weak performance of DPC in Table 1 highlights the importance of a general purpose correspondence method. We were unable to evaluate SE-ORNet on SHREC'07 as the method failed on shape classes with a low number of annotated points.

Table 1. **Evaluation on SHREC'07**. DIFF3F (ours) attains 3.5x lower error compared to DPC. This highlights the importance of a general purpose correspondence method to adeptly work on a wide range of classes.

Method	$acc \uparrow$	$err \downarrow$
DPC [4]	35.63	4.91
SE-ORNet [1]	✗	✗
DIFF3F (ours)	**52.73**	**1.41**

3.1. Robustness to Rotation

We randomly rotate shapes in SHREC'19 and TOSCA datasets, sampling X and Y direction values from a uniform distribution in $U(-180°, 180°]$. Our method's performance is compared against DPC [4] and SE-ORNet [1] in Table 2. SE-ORNet aligns shapes before computing correspondences for robustness, but this is ineffective for large rotations on SHREC'19 and TOSCA. In contrast, our method proves highly robust to substantial rotations, maintaining accuracy due to rotation invariance in the multi-view rendering and unprojection process, unlike baseline methods.

Table 2. **Robustness to rotation.** Our method being based on multi-view rendering is invariant to rotation and sees no degradation in performance.

Dataset → ↓ Method	SHREC'19		TOSCA	
	$acc \uparrow$	$err \downarrow$	$acc \uparrow$	$err \downarrow$
DPC [4]	8.83	9.03	3.89	17.87
SE-ORNet [1]	8.77	7.86	4.23	28.23
DIFF3F (ours)	**26.41**	**1.69**	**20.27**	**5.69**

4. Robustness to Text Prompt

We evaluated our method on SHREC'19 [5] using the prompt 'human' against manually labeled 'man' or 'woman.' We do not see a significant change in performance, indicating our method is robust to prompt variations. We attribute this to additional channels like normal maps and depth maps that give hints about the object. We do not see a significant change in performance when using varied text prompts for different renderings either.

5. Compute Time

DIFF3F distills 2D features from StableDiffusion [8] to 3D and hence does not require any further training. Our method takes about 2-3 minutes to compute descriptors of a shape for number of views (n) = 100 on a single Nvidia RTX 4090 GPU. The descriptors only need to be computed once and can be used without further optimization.

References

[1] Jiacheng Deng, Chuxin Wang, Jiahao Lu, Jianfeng He, Tianzhu Zhang, Jiyang Yu, and Zhe Zhang. Se-ornet: Self-ensembling orientation-aware network for unsupervised point cloud shape correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5364–5373, 2023. 1, 2

[2] Daniela Giorgi, Silvia Biasotti, and Laura Paraboschi. Shape retrieval contest 2007: Watertight models track. *SHREC competition*, 8(7):7, 2007. 1

[3] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. 3d-coded: 3d correspondences by deep deformation. In *Proceedings of the european conference on computer vision (ECCV)*, pages 230–246, 2018. 1

[4] Itai Lang, Dvir Ginzburg, Shai Avidan, and Dan Raviv. Dpc: Unsupervised deep point correspondence via cross and self construction. In *2021 International Conference on 3D Vision (3DV)*, pages 1442–1451. IEEE, 2021. 1, 2

[5] Simone Melzi, Riccardo Marin, Emanuele Rodolà, Umberto Castellani, Jing Ren, Adrien Poulenard, Peter Wonka, and Maks Ovsjanikov. Shrec 2019: Matching humans with different connectivity. In *Eurographics Workshop on 3D Object Retrieval*, page 3. The Eurographics Association, 2019. 2

[6] Bui Tuong Phong. Illumination for computer generated pictures. *Commun. ACM*, 18(6):311–317, 1975. 1

[7] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv preprint arXiv:2007.08501*, 2020. 1

[8] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 2

[9] Yiming Zeng, Yue Qian, Zhiyu Zhu, Junhui Hou, Hui Yuan, and Ying He. Corrnet3d: Unsupervised end-to-end learning of dense correspondence for 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6052–6061, 2021. 1