NeRF Analogies: Example-Based Visual Attribute Transfer for NeRFs
– Supplemental Materials –

Michael Fischer[1]* Zhengqin Li[2] Thu Nguyen-Phuoc[2] Aljaž Božič[2] Zhao Dong[2]
Carl Marshall[2] Tobias Ritschel[1]
[1]University College London [2]Meta Reality Labs Research

In this supplemental, we will detail additional details on related work, training, ViT setup and experiment protocol that could not be included in the main paper for reasons of brevity. We encourage the reader to also view the electronic supplemental where we show animated versions of our method and the baselines. Our project page is `mfischer-ucl.github.io/nerf_analogies`.

1. Extended Related Work

Inter-Surface Mappings, in pursuit of a similar goal as Neural Radiance Field (NeRF) analogies, try to establish relations between two shapes by comparing their geometric [4, 12] or, more recently, semantic [1, 9] features. However, most surface mapping methods either rely on manual annotations (*i.e.*, are non-automatic) [12], are non-robust to geometry imperfections [4], introduce discontinuous partitions [1, 8] or are limited to objects of the same topology (*e.g.*, genus-zero surfaces [9]) and hence are currently - without further adaption - not suitable for the task of creating NeRF analogies, but might provide an interesting direction for future research.

2. Implementation Details

2.1. Training

We use the standard NeRF architecture presented in [7]: a fully-connected MLP with 8 layers a 256 neurons, followed by a single layer of 128 neurons and an output layer activated by a Sigmoid function. We use the Adam optimizer [5] with a learning rate of 1×10^{-4} and a batchsize of 512. We found that some of the correspondences that DiNO produces are noisy, i.e., two points on the target geometry might map to two different points in the source NeRF. We alleviate this by training with the L1 loss, which encourages sparsity. Our total loss thus is a weighted combination of the color loss $\mathcal{L}_c$ (cf. the main text) and the DoG loss $\mathcal{L}_G$

$$\mathcal{L} = \mathcal{L}_c + \lambda\,\mathcal{L}_G,$$

where we set λ to be zero for the first 20,000 training iterations , and then gradually fade in the edge-loss by increasing λ up to 50. We train for a total of 60,000 iterations and are able to create a NeRF analogy, including the feature extraction process on 100 source- and target-images, respectively, in less than two hours on a single GPU.

2.2. ViT Setup

We use DiNO-ViT [3] with the vision transformer (ViT)-8B backbone, with a standard patch size of 8, a stride of 4 pixels and increased resolution, leading to overlapping patches and smoother feature maps. For our application, we found it important to be able to produce dense correspondences at pixel granularity, which is why we abstain from using DiNO-v2, as it uses a larger patch size and hence coarser feature granularity. To further increase the spatial resolution of the feature maps, we query DiNO on vertically and horizontally translated versions of the image (four subsequent translations by one pixel in -x and -y direction, respectively). For images of size 400p, this leads to per-image feature maps of resolution 392, with 384 features per pixel. We also experimented with diffusion (hyper-) features [6] and tried replacing, fusing and concatenating them to our DiNO-setup. This did not significantly improve the correspondence quality, but doubled the required computations (both during feature extraction and cosine-similarity computation), which is why we decided to stick with our high-resolution DiNO features. Research on ViT features has shown the positional bias to decrease with layer depth, while the semantic information increases [2]. As we do not necessarily expect semantically related regions to occupy similar image positions, we thus use the output of the deepest (11th) attention layer, specifically, the key-component of the attention maps, which has been shown to correlate well with semantic similarity [2, 13].

*Corresponding author. Work done during an internship at Meta Reality Labs Research.

2.3. Evaluation Details

For the real-world scenes, we use NeRFStudio [14] and train their Instant-NGP model on the scenes provided in the main text. For all 2D methods that are lifted to 3D, we train an Instant-NGP [10] network with standard hyperparameters for 10,000 iterations, by which point convergence has long been achieved. Our setup for all metrics and methods is 200 images, sampled randomly from a sphere around the object, and split into 100 images for training, 20 for validation and 80 for testing. We evaluate on unseen test-views. For the CLIP direction consistency metric we rendered 80 images in a circular trajectory around the object, with a constant elevation of 30°. The metrics in Tab. 1 are averaged across the set of the seven synthetic object pairs shown in Fig. 6, which were also presented to the participants of the user study. We show NeRF analogies on additional object pairs in the electronic supplemental.

3. Additional Experiments

In addition to the experiments in the main manuscript, we here investigate a range of other design decisions. Firstly, we try replacing the compute-heavy DiNO-descriptors by more lightweight SIFT features, computed densely across the image with Kornia [11]. We re-run our birdhouse test-case with SIFT- instead of DiNO-descriptors and find that they do not perform well, presumably due to SIFT not capturing semantic similarities.

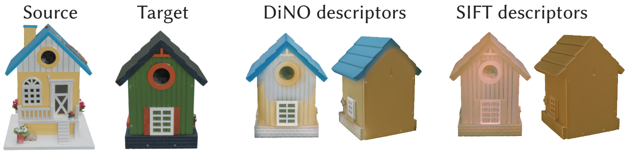

Figure 1. Comparison between DiNO- and SIFT-features.

Moreover, we note that our method can work on any input or output modality that can represent color in 3D. We thus repeat our experiments with signed distance fields (SDFs) and transfer the appearance between two SDFs fitted with NeuS2 [15].

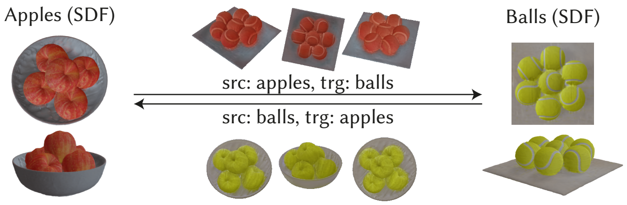

Figure 2. A semantic transfer between a bowl of apples and a set of tennis balls, both encoded as SDFs.

Additionally, we create a NeRF analogy on semantically unrelated, but similarly shaped objects. We transfer the appearance of an avocado onto an armchair of similar form and see that, while not working perfectly, our method produces a plausible outcome.

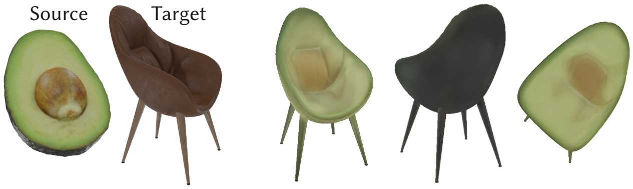

Figure 3. Transfer between semantically unrelated objects.

References

[1] Ahmed Abdelreheem, Abdelrahman Eldesokey, Maks Ovsjanikov, and Peter Wonka. Zero-shot 3d shape correspondence. *arXiv preprint arXiv:2306.03253*, 2023. 1

[2] Shir Amir, Yossi Gandelsman, Shai Bagon, and Tali Dekel. Deep vit features as dense visual descriptors. *arXiv preprint arXiv:2112.05814*, 2(3):4, 2021. 1

[3] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 1

[4] Marvin Eisenberger, Zorah Lahner, and Daniel Cremers. Smooth shells: Multi-scale shape registration with functional maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12265–12274, 2020. 1

[5] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 1

[6] Grace Luo, Lisa Dunlap, Dong Huk Park, Aleksander Holynski, and Trevor Darrell. Diffusion hyperfeatures: Searching through time and space for semantic correspondence. *arXiv preprint arXiv:2305.14334*, 2023. 1

[7] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 1

[8] Luca Morreale, Noam Aigerman, Vladimir G Kim, and Niloy J Mitra. Neural surface maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4639–4648, 2021. 1

[9] Luca Morreale, Noam Aigerman, Vladimir G Kim, and Niloy J Mitra. Neural semantic surface maps. *arXiv preprint arXiv:2309.04836*, 2023. 1

[10] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–15, 2022. 2

[11] E. Riba, D. Mishkin, D. Ponsa, E. Rublee, and G. Bradski. Kornia: an open source differentiable computer vision library for pytorch. In *Winter Conference on Applications of Computer Vision*, 2020. 2

[12] Patrick Schmidt, Dörte Pieper, and Leif Kobbelt. Surface maps via adaptive triangulations. In *Computer Graphics Forum*, volume 42. Wiley Online Library, 2023. 1

[13] Prafull Sharma, Julien Philip, Michaël Gharbi, Bill Freeman, Fredo Durand, and Valentin Deschaintre. Materialistic: Selecting similar materials in images. *ACM Transactions on Graphics (TOG)*, 42(4):1–14, 2023. 1

[14] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, et al. Nerfstudio: A modular framework for neural radiance field development. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–12, 2023. 2

[15] Yiming Wang, Qin Han, Marc Habermann, Kostas Daniilidis, Christian Theobalt, and Lingjie Liu. Neus2: Fast learning of neural implicit surfaces for multi-view reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3295–3306, 2023. 2