EUROGRAPHICS 2024 / A. Bermano and E. Kalogerakis
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 43 (2024), Number 2

Neural Semantic Surface Maps

Luca Morreale[1] and Noam Aigerman[2,3] and Vladimir G. Kim[3] and Niloy J. Mitra[1,3]

[1]Unversity College London
[2]University of Montreal
[3]Adobe Research

Appendix A: Pseudocode

We provide pseudocode for our semantic homeomorphic map extraction framework in Algorithm 1.

Algorithm 1: Semantic Surface Homeomorphism

Data: source **A**, target **B**
$\mathbf{R} \leftarrow \text{COALIGN}(\text{DinoViT}(), \mathbf{A}, \mathbf{B})$;
`fuzzyMatches` $\leftarrow$
$\text{COMPUTEMATCHES}(\text{DinoViT}(), \mathbf{A}, \mathbf{B}, \mathbf{R})$;
$A_{disk}, B_{disk} \leftarrow \text{ASYNCCUT}(\mathbf{A}, \mathbf{B}, \texttt{fuzzyMatches})$;
$A_{NSM} \leftarrow \text{OVERFITNSM}(A_{disk})$;
$B_{NSM} \leftarrow \text{OVERFITNSM}(B_{disk})$;
`map` $\leftarrow \text{DISTILMAP}(A_{NSM}, B_{NSM}, \texttt{fuzzyMatches})$;
return `map`

Appendix B: Rendering Details

In all cases, we render images of the same size, i.e., 1344×1344 with Mitsuba [JSR*22] using $spp = 150$ and a path integrator. When extracting semantic matches, we limit to rotations around the up-axis (y) - 20 steps between $[0, 2\pi)$ - and forward-axis (z) - 10 steps between $[\frac{-\pi}{2}, \frac{\pi}{2})$ - obtaining 200 images for each shape. Similarly, to align shapes, we rotate around the up-axis - 12 steps - with fixed increments. To uplift 2D pixels to 3D for the matches, we use ray-triangle intersection. On average, we get 328 correspondences per view, totaling 65k correspondences across the 200 views.

Appendix C: Computing rendering correspondences

As discussed in the main manuscript, we render the two surfaces from a given viewpoint to get two renderings, $R_V^{\mathbf{A}}$ and $R_V^{\mathbf{B}}$. We leverage DinoV2 [ODM*23] to extract semantic features in the image space, thus obtaining $\lambda_i^{\mathbf{A}}$ and $\lambda_i^{\mathbf{B}}$ as features of rendering of $R_V^{\mathbf{A}}$ and $R_V^{\mathbf{B}}$, respectively. Then, to segment foreground/background we rely on PCA's first component of these features as it naturally groups them in opposite half-spaces.

Finally, we match features with the cosine similarity between all feature pairs from the same viewpoint, as score S_{ij}. We define the match of patch $i \in R_V^{\mathbf{A}}$ as the patch $j \in R_V^{\mathbf{B}}$ with the highest cosine similarity, and vice versa, the match of patch $j \in R_V^{\mathbf{B}}$ as the patch $i \in R_V^{\mathbf{A}}$ with the highest cosine similarity. In summary, the pair $(i, j), i \in R_V^{\mathbf{A}}, j \in R_V^{\mathbf{B}}$ is a match, if

$$S_{ij} = \max_k S_{ik} \text{ or } S_{ij} = \max_l S_{lj}. \quad (1)$$

Patch generation, feature extraction, and PCA

Images are split into (non-overlapping) patches of 14 pixels. Then, Dinov2 [ODM*23] embeds these patches in a forward pass. Following [AGBD21], we use *keys* as feature vectors.

To segment foreground/background we rely on PCA's first component of the features. As discussed in [ODM*23], the features' sign naturally groups them in opposite half-spaces. As the sign is appointed randomly, we use the attention mask from the last layer to select the correct half-space: we average the first PCA component of the features and take the half-space which agrees with the positive attention mask. Matches are estimated only between foreground patches.

Finally, to unproject a match to 3D, we first translate a patch to a pixel using the known patch size, and then identify the 3D point on each shape (ray casting).

Appendix D: Comparison Details

We discuss the main considerations for/against the competing algorithms we compare against.

Blended Intrinsic Maps (BIM) [KLF11] is a classic method that uses geometric priors without any learning component. Namely, it picks a subset of self-consistent and low-distortion conformal maps

Table 1: ***Dino ViT pose ablation:*** *DinoV2 [ODM*23] matches are significantly more accurate than DinoV1 [CTM*21] in case of pose variation, with no significant difference between features from L9 and L11.*

	FAUST		SCAPE		TOSCA	
Layer	9	11	9	11	9	11
DinoV1	0.16	0.16	0.38	0.40	0.27	0.29
DinoV2	0.09	0.09	0.18	0.18	0.27	0.25

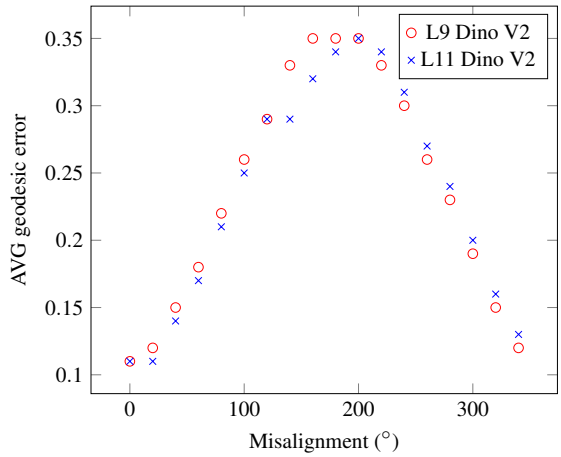

Figure 1: ***Robustness to misalignment:*** *the quality of matches depends on the quality of alignment. In the case of severe misalignment (60° or more), we observe poor correspondence.*

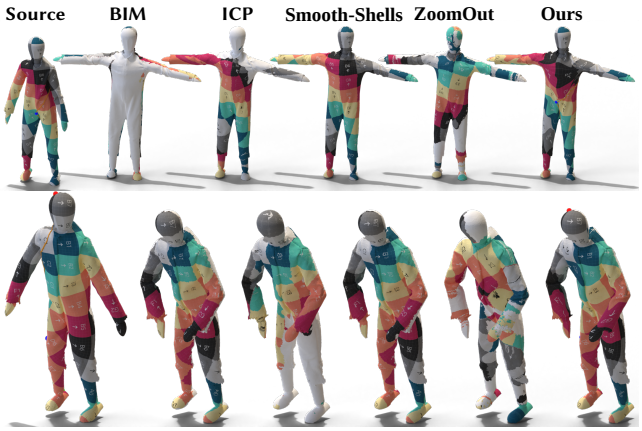

Figure 2: ***Qualitative comparison SHREC19:*** *Functional maps-based methods produce good maps, although often being discontinuous. Ours explicitly encourages continuity and bijectivity.*

and then blends them using weighted averages. Individual conformal maps can handle very non-isometric surfaces, however, they can produce high isometric distortion even in near-isometric cases. Note also that the resulting blended map is not a homeomorphism nor even continuous.

Zoomout [MRR*19] and Smooth-shells [ELC20] are both functional maps-based methods. Zoomout starts with a small functional correspondence matrix and iterative upsamples it in the spectral domain. Smooth-shells follow a similar coarse-to-fine scheme, relying on shells as a proxy for functional basis. To handle self-symmetries, Eisenberger et al. [ELC20] incorporate MCMC to evaluate multiple possible functional maps.

We initialize Zoomout's map (C_{21}) as an identity of size 4 as by official implementation. Then, we refine it until it contains 50 eigenvectors. Similarly, for Smooth-shells we follow the official implementation and use MCMC to bootstrap the map using $K_{min} = 6$ and $K_{max} = 20$. and evaluating $N_{prop} = 500$ proposal. In both cases, no landmarks are used. Finally, for ICP we first align the two input shapes as described in Sec. 3.1, and then estimate the nearest neighbor for each vertex.

We depict maps for the different methods on SHREC19 in Figure 2. State-of-the-art methods work well as they exploit geometric cues, although they are susceptible to self-symmetries (see BIM [KLF11] first row). Conversely, "Ours" relies purely on visual cues, with no isometric regularization, thus being less accurate on average.

Appendix E: Differences with Neural Surface Maps

Neural Surface Maps [MAKM21] defines the general mapping framework used to optimize maps. Following the original work, the input two shapes must be homomorphic to a disk with their boundary in correspondence. As this constraint is impossible to satisfy automatically, this work relies on seamless maps, thus relaxing this constraint to 3 corresponding points which are extracted automatically. Furthermore, we define a soft correspondence term to handle inaccurate correspondences, while NSM enforces exact correspondences with an L2 loss over all correspondences.

Appendix F: Metrics

In all experiments, all shapes are automatically normalized and centered.

Bijectivity We estimate the map's bijectivity of the shape vertices for all baselines. For ICP, BIM, Zoomout, and Smooth-shells we map all vertices forward ($A \rightarrow B$) and then backward ($A \leftarrow B$), using the forward and backward map respectively. Then, we compute the geodesic distance between the starting vertex and its forward-backward map.

Similarly, for consistency we evaluate "Ours" bijectivity only for the shape vertices. In particular, we map a vertex in A onto B's 2D domain through h, and then, we use the piecewise linear map for 2D-3D. For B to A, we pullback vertices through barycentric coordinates after mapping forward all A's triangles. Empirically, for "Ours" *we never observe flips*; while for baselines, correspondences are always given, thus, no ambiguity arises. In the case of a non-bijective map, we would consider the first triangle.

Appendix G: Ablation

On Dinov2 features

As aforementioned, we deem a match if the cosine similarity S_{ij} between patch features - $\lambda_i^{\mathbf{A}}$ and $\lambda_i^{\mathbf{B}}$ - is the highest. While this is a common similarity measure, it is important to acknowledge its inherent limitations. Specifically, one notable challenge is that similarity scores derived from different images may not be directly comparable. For example, for two correspondences with scores 0.9 and 0.8, the former match pair is not necessarily better than the latter. In essence, features extracted from one view may be extremely dissimilar to those extracted from another view, even for the same

Table 2: ***Dino ViT ablation:*** *DinoV2 [ODM*23] works better than its predecessor [CTM*21], with no significant difference between features from L9 and L11. The use of colored lights (rows DinoV1 and DinoV2) offers better visual cues to extract matches than white lights. Although counter-intuitive, the use of* simple *texture reduces the visual cues available to Dino ViT.*

	FAUST		SHREC15		3DBiCar	
Layer	9	11	9	11	9	11
DinoV1	0.10	0.12	0.32	0.32	0.36	0.49
DinoV2	0.11	0.11	0.24	0.24	0.33	0.33
white lights (V1)	0.20	0.18	0.27	0.35	0.38	0.38
white lights (V2)	0.11	0.11	0.24	0.24	0.30	0.31
texture (V1)	-	-	-	-	0.26	0.26
texture (V2)	-	-	-	-	0.29	0.29

shape. This arises from the inherent variation in image structure across different views and how features are generated from them. This inherent variability hinders consistency in cross-image feature comparisons. Consequently, the process of aggregating features across different views can potentially yield unexpected outcomes, leading to either incorrect matching or highly inaccurate results.

Experimentally, sampling the top $k = 100$ correspondences based on the similarity produced far worse results than uniform sampling or uniform weighting, see Figure 4 for qualitative comparison. In both cases, we optimize maps following the proposed algorithm: *Ours* uses all correspondences, while *TopK* is limited to $k = 100$ correspondences with the highest similarity score. Visibly, some of these correspondences are incorrect and bias the map towards incorrect minima, thus their similarity score is not representative of their quality. Indeed, the use of all correspondences prevents the map from falling into such a degenerate solution, as the majority of correspondences are reasonably correct.

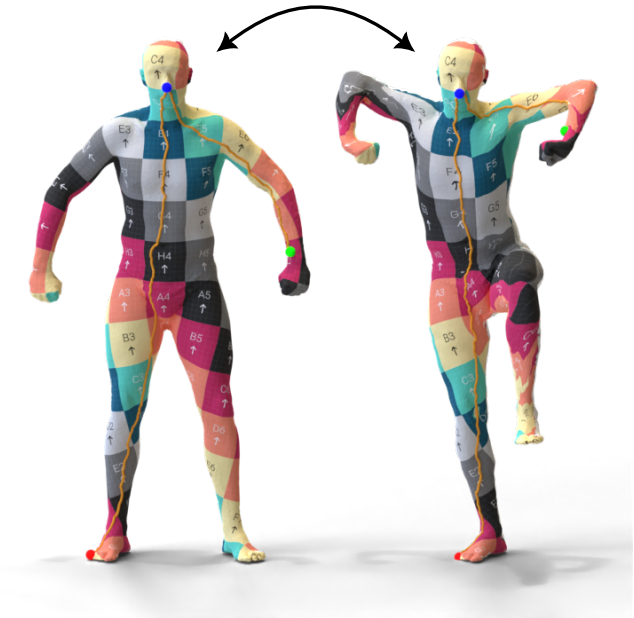

Figure 3: ***Pose variation:*** *we assess the ability of DinoV2 [ODM*23] to establish matches between shapes in different poses, as those in the figure. Experimentally, DinoV2 yields correspondences able to guide our pipeline to a proper solution. Colored landmarks and paths show* automatically *selected cones and cuts by our method.*

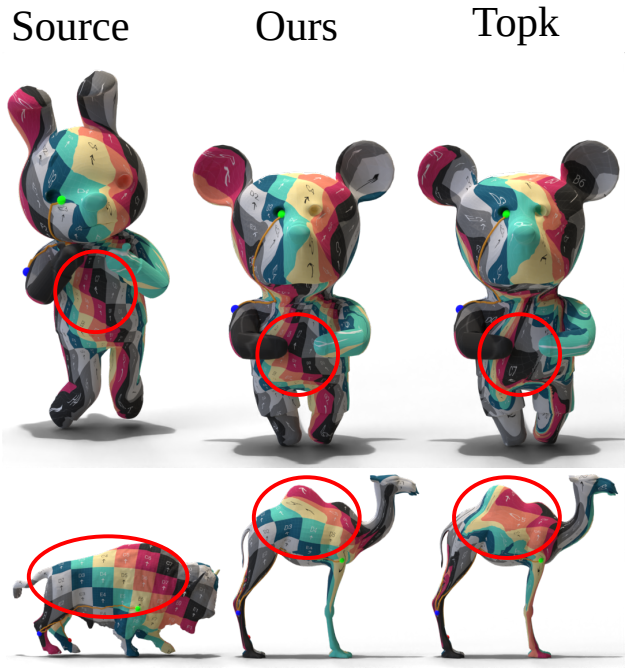

Figure 4: ***Similarity scores. Right****: a map optimized with the top $k = 100$ correspondences based on the similarity score.* ***Middle****: map optimized using all correspondences.* ***Left****: source mesh. The map optimized with matches with the highest similarity score matches shows several incorrectness, highlighted with a red circle. This is the result of several incorrect matches which bias the map towards an incorrect energy minimum. Differently, using all correspondences prevents this behavior, as the optimization process automatically filters out wrong matches.*

Tuning DinoVit Matches

We ablate the quality of matches based on DinoViT's degrees of freedom - layer features - in different contexts: pose variation, presence of texture, lights, and misalignment. We conduct our analysis on three distinct datasets: FAUST [BRLB14], 3DBiCar [LCD*23], and SHREC15 [LZEE*15] each with dense or sparse ground truth.

We select 12 shape pairs, 4 for each dataset, to ablate texture and misalign concerning the choice of Dino ViT feature layer, as discussed in [AGBD21]. Similarly, we assess the effect of pose variation for the same model with a single instance of FAUST, SCAPE, and TOSCA mapped onto all the other provided poses. We report

the quantitative results in Table 2 and show shape pairs examples and qualitative optimization results in Figure 3.

We assess the quality of the aggregated correspondences in terms of the normalized average geodesic distance [KLF11]. We follow the procedure described in the main paper to aggregate the fuzzy correspondences, thus, obtaining a face-wise map M from one mesh onto the other. Finally, the geodesic distance is computed on the target mesh between the centroid of the mapped face to the centroid of the ground truth target face.

In general, DinoV2 [ODM*23] outperforms its predecessor V1 [CTM*21], offering more accurate and robust matches. The depth at which features are extracted (9 vs 11) does not impact the matches of DinoV2, while it plays a significant role for DinoV1, as discussed in [AGBD21]. The presence of texture is beneficial to DinoV1, while it only offers a minor improvement for DinoV2. This is reassuring as our method can only assume access to untextured models. The choice of colored lights offers additional shading and visual features for DinoV1, but it is less relevant for DinoV2 as white lights perform equally with the base case.

Effect of Initial Alignment

We ablate the effect and robustness to misalignment for correspondences quality, see Figure 1. We start from a correct alignment with 12 shape pairs and incrementally misalign one shape - step of 20° around the up axis. We report the quality of correspondences in terms of geodesic error, i.e., accuracy. The quality sensibly decreases with severe misalignment - more than 40° - reaching a peak with opposite orientation - 180°. We additionally compare the quality of correspondences for the last two layers of Dino-ViT and show that, for such a case, a deeper level (L11) seems to encode slightly better semantic information than the previous layer (L9).

Handling Noise and Holes

Raw scans present noise or holes, thus inhibiting the applicability of our method since it assumes watertight genus zero meshes. Intuitively the presence of large holes, and missing limbs such as arms, may severely mislead DinoViT and thus our pipeline. On the other hand, small holes can be dealt with by applying a simple hole-filling approach. In Figure 5, we use our method to map a raw scan to the SMPL template [LMR*15]. We prefill small holes with Meshlab [CCC*08] and then apply our pipeline.

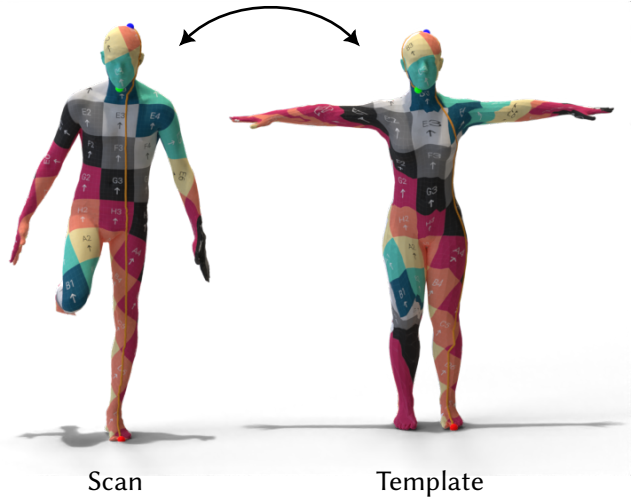

Figure 5: ***Scan to SMPL:*** *we first close holes in the raw scan (left) with Meshlab [CCC*08], then we map it onto the template SMPL model [LMR*15] and mask out the surfaced introduced to fill holes. Colored landmarks and paths show* automatically *selected cones and cuts by our method.*

References

[AGBD21] AMIR S., GANDELSMAN Y., BAGON S., DEKEL T.: Deep vit features as dense visual descriptors. *arXiv preprint arXiv:2112.05814* (2021). 1, 3, 4

[BRLB14] BOGO F., ROMERO J., LOPER M., BLACK M. J.: Faust: Dataset and evaluation for 3d mesh registration. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2014), pp. 3794–3801. 3

[CCC*08] CIGNONI P., CALLIERI M., CORSINI M., DELLEPIANE M., GANOVELLI F., RANZUGLIA G.: MeshLab: an Open-Source Mesh Processing Tool. In *Eurographics Italian Chapter Conference* (2008), Scarano V., Chiara R. D., Erra U., (Eds.), The Eurographics Association. 4

[CTM*21] CARON M., TOUVRON H., MISRA I., JÉGOU H., MAIRAL J., BOJANOWSKI P., JOULIN A.: Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 9650–9660. 1, 3, 4

[ELC20] EISENBERGER M., LAHNER Z., CREMERS D.: Smooth shells: Multi-scale shape registration with functional maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), pp. 12265–12274. 2

[JSR*22] JAKOB W., SPEIERER S., ROUSSEL N., NIMIER-DAVID M., VICINI D., ZELTNER T., NICOLET B., CRESPO M., LEROY V., ZHANG Z.: Mitsuba 3 renderer, 2022. https://mitsuba-renderer.org. 1

[KLF11] KIM V. G., LIPMAN Y., FUNKHOUSER T.: Blended intrinsic maps. *Transactions on Graphics (Proc. of SIGGRAPH) 30*, 4 (2011). 1, 2, 4

[LCD*23] LUO Z., CAI S., DONG J., MING R., QIU L., ZHAN X., HAN X.: Rabit: Parametric modeling of 3d biped cartoon characters with a topological-consistent dataset. *arXiv preprint arXiv:2303.12564* (2023). 3

[LMR*15] LOPER M., MAHMOOD N., ROMERO J., PONS-MOLL G., BLACK M. J.: Smpl: A skinned multi-person linear model. *ACM transactions on graphics (TOG) 34*, 6 (2015), 1–16. 4

[LZEE*15] LIAN Z., ZHANG Z., EL ELNAGHY H., EL SANA J., FURUYA T., GIACHETTI A., GÜLER A., LAI L., LI C., LI H., ET AL.: Shrec 15 track non rigid 3d shape retrieval. 3

[MAKM21] MORREALE L., AIGERMAN N., KIM V. G., MITRA N. J.: Neural surface maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 4639–4648. 2

[MRR*19] MELZI S., REN J., RODOLA E., SHARMA A., WONKA P., OVSJANIKOV M.: Zoomout: Spectral upsampling for efficient shape correspondence. *arXiv preprint arXiv:1904.07865* (2019). 2

[ODM*23] 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). 1, 3, 4