Supplementary Materials: Smooth Non-Rigid Shape Matching via Effective Dirichlet Energy Optimization

Robin Magnet
LIX, École Polytechnique, IP Paris
rmagnet@lix.polytechnique.fr

Jing Ren
ETH Zurich
jing.ren@inf.ethz.ch

Olga Sorkine-Hornung
ETH Zurich
sorkine@inf.ethz.ch

Maks Ovsjanikov
LIX, École Polytechnique, IP Paris
maks@lix.polytechnique.fr

1. Smoothness Reformulation

In this section, we give details on the reformulation of smoothness methods provided in the main manuscript.

1.1. Non-Rigid ICP

Non-Rigid ICP (nICP) [1] deforms a source shape $\mathcal{S}_1$ into a target shape $\mathcal{S}_2$ using a per-vertex affine deformation $\mathbf{D}$. Global energy reads, trying to fit a pointwise map Π_{12}

$$E_{\text{nicp}}(\Pi_{12}, \mathbf{D}) = \|\mathbf{D}\|^2_{W_1} + \beta \|\mathbf{D} \circ X_1 - \Pi_{12} X_2\|^2_{A_1} \quad (1)$$

with $\|\mathbf{D}\|^2_{W_1} = \sum_{i\sim j} w_{ij} \|D_i - D_j\|^2_F$ and $\mathbf{D} \circ X_1$ the deformed vertex coordinates. Given a point-wise map Π_{12}, one can directly incorporate this energy in our algorithm, where solving for Y_{12} is replaced by solving for $\mathbf{D}$, and then setting $Y_{12} = \mathbf{D} \circ X_1$. Solving for $\mathbf{D}$ reduces to a simple linear system, as explained in [1]. Note that in the original work, nICP algorithm uses graph Laplacian instead of cotan Laplacian, but we find that using cotan weights is more stable in the case of triangle meshes. We furthermore ignored landmarks preservation terms, borders skipping heuristic, normals preservation and self-intersection verification procedures for simplicity.

1.2. As-Rigid-As-Possible

As-Rigid-As-Possible (ARAP) [12] promotes *local rigidity* of the deformation of a shape $\mathcal{S}_1$ using per-vertex rotations $\mathbf{R}$, which results in minimizing the following energy:

$$E_{\text{arap}}(\mathbf{R}, Y) = \sum_{i\sim j} w_{ij} \|(y_i - y_j) - R_i(x_i - x_j)\|^2_F \quad (2)$$

where y_i are the expected vertex coordinates and x_i the undeformed coordinates.

We observe that the ARAP energy can be decomposed into two main components including a smoothness term and a rigidity term:

$$E_{\text{arap}}(\mathbf{R}, Y) = E^{\text{smooth}}_{\text{arap}}(Y) - 2E^{\text{rigid}}_{\text{arap}}(\mathbf{R}, Y) + \text{const.} \quad (3)$$

$$E^{\text{smooth}}_{\text{arap}}(Y) = \sum_{(x_i, x_j)\in\mathcal{E}(\mathcal{S}_1)} w_{ij} \|y_i - y_j\|^2_F \quad (4)$$

$$E^{\text{rigid}}_{\text{arap}}(\mathbf{R}, Y) = \sum_{(x_i, x_j)\in\mathcal{E}(\mathcal{S}_1)} w_{ij} (y_i - y_j)^T R_i (x_i - x_j) \quad (5)$$

with $E^{\text{smooth}}_{\text{arap}} = \|Y\|^2_{W_1} = E_D(Y)$. Note, however, that the default ARAP energy does not have a coupling term to ensure that Y remains on the surface of $\mathcal{S}_2$. Therefore, to avoid a trivial solution, such an energy must rely on pre-existing landmarks to make sure that the deformation maps onto the target shape. In our algorithm, given a pointwise map Π_{12}, we instead decide to add a coupling term between the expected coordinates Y_{12} and transferred coordinates $\Pi_{12} X_1$, which slightly modifies the linear system to solve when minimizing over Y_{12}, but doesn't involve the rotations $\mathbf{R}$. Therefore, given a pointwise map Π_{12}, one first needs to compute local rotations $\mathbf{R}$ and can then obtain the expected coordinates Y_{12} by solving a linear system.

1.2.1 Smooth Shells

Smooth Shells [5] models the deformation $\mathbf{D}$ as a simple per-vertex translation seen as a function $\mathcal{S}_1 \to \mathbb{R}^3$, which is restricted to lie in the *spectral* basis of size K, i.e., $\mathbf{D} \in \mathbb{R}^{K\times 3}$. In addition smooth shells uses the ARAP energy to enforce the smoothness of the deformation, which therefore adds additional local rotation $\mathbf{R}$. Specifically, $X_1 + \Phi_1 \mathbf{D}$ would give the updated vertex positions and the smoothness

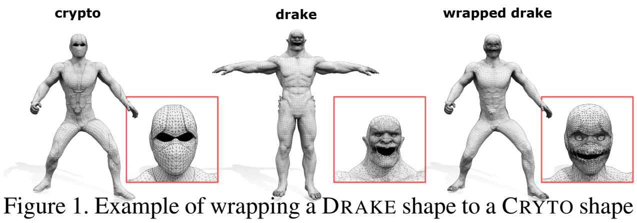

Figure 1. Example of wrapping a DRAKE shape to a CRYTO shape to establish cross-category correspondences.

is then defined as:

$$E_{\text{shells}}^{\text{smooth}}(\mathbf{D},\mathbf{R}) = E_{\text{arap}}(\mathbf{R}, X_1 + \Phi_1\mathbf{D}) \tag{6}$$

The smoothness energy is again associated with a coupling term which ensures the deformed shape remains close to the current correspondences $\|X_1 + \Phi_1\mathbf{D} - \Pi_{12}X_2\|_{A_1}^2$. Note that in the original work, vertices X_1 and X_2 are also projected to a spectral basis, and extra feature and normal preservation terms are added. In practice, solving for $\mathbf{D}$ reduces to solving a $K \times K$ linear system, compared to the $n \times n$ linar system obtained with standard ARAP.

1.2.2 Reversible Harmonic Maps

Reversible Harmonic Maps (RHM) [6] directly minimizes the Dirichlet energy of a map without manipulating deformation fields. To avoid making the map collapse the authors look for bijective maps with the lowest possible Dirichlet energy. Vertices of the pull-back shape $\Pi_{ij}X_j$ for $(i,j) \in \{(1,2),(2,1)\}$ are again estimated via an auxiliary variable Y_{ij} and the energy reads as the sum in both directions of $E_{\text{rhm}}^{\text{half}}$ with :

$$\begin{aligned} E_{\text{rhm}}^{\text{half}}(\Pi_{ij},\Pi_{ji},Y_{ij},Y_{ji}) = E_D(Y_{ij}) + E_{\text{rhm}}^{\text{bij}}(\Pi_{ji},Y_{ij}) \\ + E_{\text{rhm}}^{\text{couple}}(\Pi_{ij},Y_{ij}). \end{aligned} \tag{7}$$

Here, again, we recognize the Dirichlet energy of the estimated map $E_D(Y_{ij})$, and two terms $E_{\text{rhm}}^{\text{bij}}$ and $E_{\text{rhm}}^{\text{couple}}$ which respectively enforce bijectivity and coupling:

$$E_{\text{rhm}}^{\text{bij}}(\Pi_{ji},Y_{ij}) = \left\|\Pi_{ji}Y_{ij} - X_j\right\|_{A_j}^2 \tag{8}$$

$$E_{\text{rhm}}^{\text{couple}}(\Pi_{ij},Y_{ij}) = \left\|Y_{ij} - \Pi_{ij}X_j\right\|_{A_i} \tag{9}$$

This formulation leads to a computationally expensive iterative solver, which can obtain to great results given an already good initialization. Additionally, the authors use a high-dimensional embedding obtained via MDS [4] which mimics the geodesic distance, instead of directly using the embedding coordinates.

2. DEFORMTHINGS4D-MATCHING Dataset

Here we discuss in details how we construct our dataset from the DEFORMTHINGS4D [7] for shape matching task:

1. **Select Models.** We first pick models in DEFORMTHINGS4D that are close to watertight. Specifically, we only keep the models where the number of vertices in the largest connected components is more than 75% of the total number of vertices. Then the largest connected component is taken if the model is disconnected. As a result, we get 56 animal models and 8 humanoid models.

2. **Select Poses.** For each watertight model, we collect all motion clips in DEFORMTHINGS4D and select poses from all the frames that are sufficiently different from each other. Specifically, we first pick a base pose that is close to an A-pose: we find the pose that has large range in z-axis and has relatively small range in xy-axis. We then recursively find new pose from the collection that have the largest difference in vertex positions to the chosen ones, until we get 50 poses or all the poses are included. We then manually check each chose pose and remove unrealistic poses with large distortion or self-intersection. As a result, the number of poses for each model has a range from 30 to 50.

3. **Remeshing.** The chosen poses in each model are in the same triangulation, which can lead to overfitting issues for some shape matching methods [11]. We therefore apply a geometry-aware remeshing algorithm, LRVD [13], to independently remesh all the poses to the resolution of around 8K vertices. The correspondences between the remeshed shapes are propagated by nearest-neighbor searching between the remeshed shapes and the original shapes. To fix the potential topological errors in the nearest neighbor map, we apply spectral ICP [8] at dimension 500 of the Laplace-Beltrami Basis.

4. **Wrapping.** We also provide cross-category correspondences for the 8 humanoid models. Specifically, we use the commercial software R3DS to wrap the rest 7 models (ZLORP, MANNEQUIN, DRAKE, NINJA, PRISONER, PUMPKINHULK, SKELETONZOMBIE) to the chosen model (CRYPTO, the left-most shapes in Fig. 2 in the main paper). For each pair, we manually select 50-80 landmarks on shapes for wrapping. Note here we wrap the original models and propagate the correspondences to the remeshed shapes afterwards. Specifically, the cross-category correspondences among the original poses can be established by nearest-neighbor searching between the wrapped shape and the target shape (see Fig. 1 for an example of a wrapped shape), which are then propagated to the remeshed poses similar to step 3. Note that, since some shapes are far from isometry or even incomplete, the wrapped results are not perfect, and hence the established correspondences via map compositions can be inaccurate. In general, as illustrated in

Table 1. **Accuracy** on DEFORMTHINGS4D-MATCHING

methods	*near-isometric*				*partial*	*non-iso*
	ZLORP	DRAKE	MANNEQUIN	NINJA	PRISONER	ZOMBIE
Init	11.49	9.59	8.62	10.43	20.78	15.33
Ours w/ ARAP	11.22	9.04	8.10	9.88	19.91	14.83
Ours w/ nICP	7.29	7.07	4.61	5.25	21.18	11.95
Ours w/ Shells	3.25	7.78	4.11	4.73	20.27	10.32
ZO	3.43	5.74	3.33	4.61	20.59	13.71
DO	3.26	5.95	3.64	5.10	19.59	16.53
Ours w/ D	3.72	6.93	4.18	4.80	19.81	9.71
Ours w/ RHM	3.70	5.63	3.94	5.46	18.85	11.00

Table 2. **Bijectivity** on DEFORMTHINGS4D-MATCHING

methods	*near-isometric*				*partial*	*non-iso*
	ZLORP	DRAKE	MANNEQUIN	NINJA	PRISONER	ZOMBIE
Init	11.69	7.17	6.58	10.69	22.53	11.52
Ours w/ ARAP	11.93	7.25	7.69	10.42	21.71	11.18
Ours w/ nICP	3.63	2.73	2.58	2.49	7.17	4.71
Ours w/ Shells	1.67	2.16	2.22	2.23	3.56	3.71
ZO	2.14	4.05	1.37	3.99	21.19	10.11
DO	1.27	1.55	1.63	1.46	2.26	2.52
Ours w/ D	1.77	2.12	2.30	2.25	3.60	3.74
Ours w/ RHM	1.42	1.84	1.82	1.94	2.81	3.24

Table 3. **Coverage** on DEFORMTHINGS4D-MATCHING

methods	*near-isometric*				*partial*	*non-iso*
	ZLORP	DRAKE	MANNEQUIN	NINJA	PRISONER	ZOMBIE
Init	22%	34%	35%	28%	8%	20%
Ours w/ ARAP	28%	37%	36%	34%	22%	29%
Ours w/ nICP	38%	50%	53%	53%	20%	30%
Ours w/ Shells	61%	55%	56%	57%	39%	43%
ZO	72%	70%	71%	71%	59%	60%
DO	68%	66%	65%	66%	55%	55%
Ours w/ D	59%	55%	54%	56%	37%	41%
Ours w/ RHM	64%	60%	60%	60%	45%	47%

Table 4. **Smoothness** via Conformal Distortion on DEFORMTHINGS4D-MATCHING

methods	*near-isometric*				*partial*	*non-iso*
	ZLORP	DRAKE	MANNEQUIN	NINJA	PRISONER	ZOMBIE
Ours w/ ARAP	2.33	2.99	2.21	2.24	3.02	2.10
Ours w/ nICP	4.14	5.15	2.59	2.90	10.49	4.58
Ours w/ Shells	3.22	4.68	3.77	4.56	14.04	7.13
ZO	3.05	5.03	2.51	4.22	24.80	15.76
DO	3.23	5.30	3.77	4.69	21.10	16.09
Ours w/ D	2.85	3.70	2.81	3.05	9.89	4.54
Ours w/ RHM	2.88	3.92	2.78	3.07	10.05	4.72

Fig. 2 in the main paper, the established correspondences are in reasonable accuracy.

3. FAUST dataset

The FAUST dataset [3] consists in 100 meshes of 10 individuals in 10 different poses.

This dataset is used as a standard benchmark for most shape-matching algorithms. However as all shapes are near-isometric, many methods achieve smooth and accurate results for this dataset. This therefore gives very little room for improvement regarding the smoothness.

We provide results on a random subset of 200 pairs in the main manuscript, where pairs we selected so that only cross-individual ones are considered.

4. Additional Results

We evaluate different methods using accuracy, bijectivity, coverage, and smoothness of the maps as metrics. We also report runtime to compare the efficiency. Specifically, we compute the geodesic distances between the obtained maps T_{ij} and the ground-truth maps (if available) to measure the *accuracy* (see Tab. 1). Similarly, we compute the geodesic distances between the composite maps $T_{ij} \circ T_{ji}$ and the identity map I_{n_i} to measure the *bijectivity* of the pointwise maps (see Tab. 2). We compute the Dirichlet energy on the obtained pointwise maps to evaluate the smoothness (defined in Eq. (3) in the main paper) as shown in Tab. 1 in the main paper. Here we additionally evaluate the conformal distortion [6, 9], another popular smoothness metric as shown in Tab. 4. We finally compute coverage of a pointwise map T, i.e., the area ratio of the target shape that is covered by map T, which evaluates the map surjectivity (see Tab. 3). This metric must be considered in pair with smoothness to detect degenerate case of trivial maps with perfect smoothness. For example, a trivial map where are vertices on the source are mapped to the same vertex on the target, is perfectly smooth w.r.t. the Dirichlet energy, but its coverage is close to zero. Therefore, in the ideal case, the best map is the one with zero Dirichlet energy and 100% coverage. All metrics are reported as an average over all the tested shape pairs.

In Fig. 2, we show some qualitative results on the TOSCA non-isometric dataset.

5. Parameters

In all experiments, we use the same set of parameters, where those of each smoothness energy were tuned independently. Parameters can also be found in the released implementation at https://github.com/RobinMagnet/smoothFM.

Spectral Energy. For all experiments we weighted the spectral bijectivity term by 1 and the coupling term by

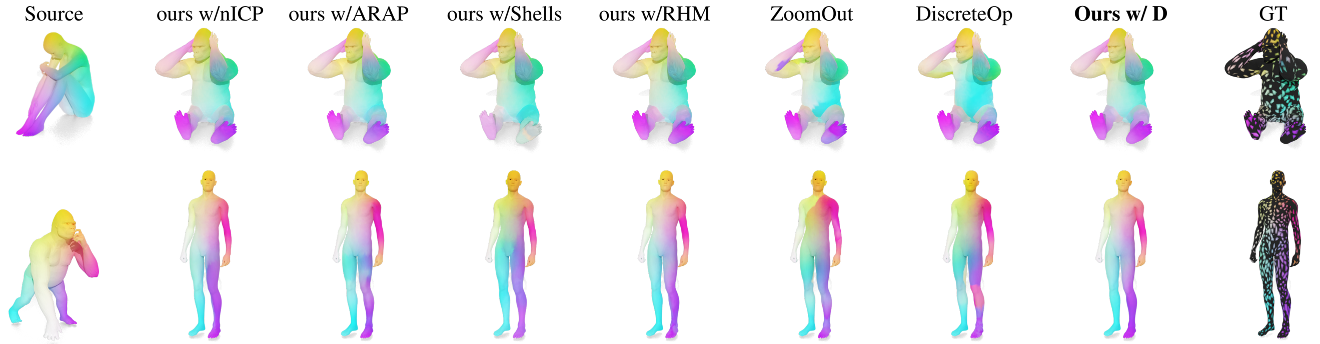

Figure 2. We show two non-isometric shape pairs from TOSCA dataset can compare pointwise maps obtained from different methods via color transfer. Note that TOSCA non-isometric dataset only provide *sparse* ground-truth correspondences. We therefore color the vertices that do not have GT correspondences in *black*.

Table 5. Results on TOSCA nonIsometric using WKS initialization

methods	*accuracy*	*bijectivity*	*smoothness*	*coverage*
Init	56.56	39.50	93.24	15.48 %
Zo	54.61	43.23	19.27	52.48 %
DO	53.65	2.33	16.47	50.04 %
Ours w/ D	51.38	22.30	2.46	16.72 %
Ours w/ RHM	54.07	4.18	3.92	35.29 %

10^{-1}, as advocated in the Discrete Optimization implementation [10].

Smoothness Energy. Each smoothness energy required its own set of parameters. The Dirichlet energy was weighted by 1 for all of them for consistency. In particular, for RHM energy, we used a coupling weight of 1 and a bijectivity weight of 10^4. We used a coupling weight of 10^{-1} for ARAP, 10^{-2} for nICP and 10^{-3} for Shells.

Coupling. We globally reweighted the smoothness energy by a parameter γ, gradually increasing from 10^{-1} to 1 across iterations.

6. Initialization

For all datasets, we obtain initial dense correspondences by computing a 5×5 functional map using 5 landmarks.

We chose this kind of initialization as standard shape descriptors like WKS [2] could not provide meaningful correspondences in the presence of high levels of non-isometry.

Indeed, Table 5 provides results using WKS descriptor as initialization for all methods. Note that the accuracy is unable to significantly go down from initialization. It thus becomes difficult to read into these results in a meaningful manner.

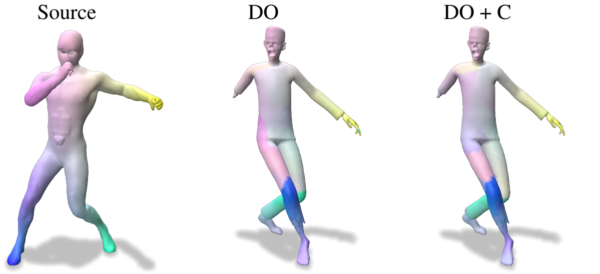

Figure 3. Example of correspondences without (enter) and with (right) the conformal term of Discrete Optimization. While some parts are more smoother, the overall effect is marginal.

7. Discrete Optimization

The discrete optimization framework [10] proposes a large set of spectral energies, along which the conformal energy promoting functional maps associated to conformal pointwise correspondences. While this energy does help smoothness, we did not notice significant improvements regarding discontinuities in the correspondences.

On Figure 3, we display an example of correspondences obtained by the standard Discrete Optimization (center) and by adding the conformal term (right). While some parts have been made smoother, the effect remain quite marginal.

In practice, this term can provide meaningful regularization in some cases but appears quite hard to tune to obtain a consistent effect.

References

[1] Brian Amberg, Sami Romdhani, and Thomas Vetter. Optimal step nonrigid icp algorithms for surface registration. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2007. 1

[2] Mathieu Aubry, Ulrich Schlickewei, and Daniel Cremers. The wave kernel signature: A quantum mechanical approach to shape analysis. In *2011 IEEE international conference on computer vision workshops (ICCV workshops)*, pages 1626–1633. IEEE, 2011. 4

[3] Federica Bogo, Javier Romero, Matthew Loper, and Michael J. Black. FAUST: Dataset and evaluation for 3D mesh registration. In *Computer Vision and Pattern Recognition (CVPR)*, pages 3794–3801, Columbus, Ohio, 2014. IEEE. 3

[4] Michael AA Cox and Trevor F Cox. Multidimensional scaling. In *Handbook of data visualization*, pages 315–347. Springer, 2008. 2

[5] Marvin Eisenberger, Zorah Lahner, and Daniel Cremers. Smooth shells: Multi-scale shape registration with functional maps. In *Computer Vision and Pattern Recognition (CVPR)*, pages 12265–12274, 2020. 1

[6] Danielle Ezuz, Justin Solomon, and Mirela Ben-Chen. Reversible harmonic maps between discrete surfaces. *ACM Transactions on Graphics (TOG)*, 38(2):15:1–15:12, 2019. 2, 3

[7] Yang Li, Hikari Takehara, Takafumi Taketomi, Bo Zheng, and Matthias Nießner. 4dcomplete: Non-rigid motion estimation beyond the observable surface. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12706–12716, 2021. 2

[8] Maks Ovsjanikov, Mirela Ben-Chen, Justin Solomon, Adrian Butscher, and Leonidas Guibas. Functional maps: a flexible representation of maps between shapes. *ACM Transactions on Graphics (TOG)*, 31(4):30:1–30:11, 2012. 2

[9] Jing Ren, Simone Melzi, Maks Ovsjanikov, and Peter Wonka. Maptree: Recovering multiple solutions in the space of maps. *ACM Transactions on Graphics (TOG)*, 39(6), Nov. 2020. 3

[10] Jing Ren, Simone Melzi, Peter Wonka, and Maks Ovsjanikov. Discrete optimization for shape matching. In *Computer Graphics Forum*, volume 40, pages 81–96. Wiley Online Library, 2021. 4

[11] Jing Ren, Adrien Poulenard, Peter Wonka, and Maks Ovsjanikov. Continuous and orientation-preserving correspondences via functional maps. *ACM Transactions on Graphics (TOG)*, 37(6), 2018. 2

[12] Olga Sorkine and Marc Alexa. As-rigid-as-possible surface modeling. In *Computer Graphics Forum*, volume 4, pages 109–116. Wiley Online Library, 2007. 1

[13] Dong-Ming Yan, Guanbo Bao, Xiaopeng Zhang, and Peter Wonka. Low-resolution remeshing using the localized restricted voronoi diagram. *IEEE Transactions on Visualization and Computer Graphics*, 20(10):1418–1427, 2014. 2