

# A Scalable Combinatorial Solver for Elastic Geometrically Consistent 3D Shape Matching

## – APPENDIX –

Paul Roetzer<sup>1,2</sup>    Paul Swoboda<sup>3</sup>    Daniel Cremers<sup>1</sup>    Florian Bernard<sup>2</sup>  
 TU Munich<sup>1</sup>    University of Bonn<sup>2</sup>    MPI Informatics<sup>3</sup>

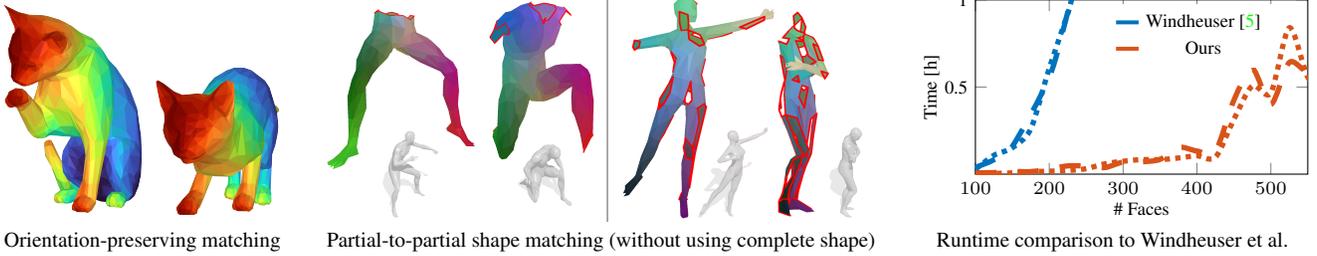


Figure A1. We propose a **novel combinatorial solver for the non-rigid matching of 3D shapes** based on discrete orientation-preserving diffeomorphisms [5] (left). For the first time we utilize an orientation-preserving diffeomorphism to constrain the challenging problem of non-rigidly matching a pair of partial shapes without availability of complete shapes (center). Our solver scales significantly better compared to existing solvers and can thus handle shapes with practically relevant resolutions (right).

### A1. Product Triangles to Explore

In the following we explain the neighborhood between product triangles. For the  $f$ -th product triangle  $(a_1b_1, a_2b_2, a_3b_3)$  the set of neighbors  $F_{\mathcal{N}(f)}$  is defined as

$$F_{\mathcal{N}(f)} = \left\{ \begin{array}{l} (a'_1b'_1 = a_2b_2 \wedge a'_2b'_2 = a_1b_1) \vee \\ (a'_1b'_1 = a_3b_3 \wedge a'_2b'_2 = a_2b_2) \vee \\ (a'_1b'_1 = a_1b_1 \wedge a'_2b'_2 = a_3b_3) \vee \\ \left( \begin{array}{l} a'_1b'_1 \\ a'_2b'_2 \\ a'_3b'_3 \end{array} \right) \in F : \left. \begin{array}{l} (a'_2b'_2 = a_2b_2 \wedge a'_3b'_3 = a_1b_1) \vee \\ (a'_2b'_2 = a_3b_3 \wedge a'_3b'_3 = a_2b_2) \vee \\ (a'_2b'_2 = a_1b_1 \wedge a'_3b'_3 = a_3b_3) \vee \\ (a'_3b'_3 = a_2b_2 \wedge a'_1b'_1 = a_1b_1) \vee \\ (a'_3b'_3 = a_3b_3 \wedge a'_1b'_1 = a_2b_2) \vee \\ (a'_3b'_3 = a_1b_1 \wedge a'_1b'_1 = a_3b_3) \end{array} \right\} \end{array} \right.$$

In words, every product triangle which shares an opposite oriented edge with the  $f$ -th product triangle is neighboring to the  $f$ -th product triangle. The union of all sets  $F_{\mathcal{N}(f)}$  yields the set of exploration candidates

$$F_{\text{expl}} = \bigcup_{f \text{ part of solution}} F_{\mathcal{N}(f)}. \quad (\text{A1})$$

When searching for new matchings with our primal heuristic, we only iterate over the product triangles in  $F_{\text{expl}}$ . If none of the product triangles in  $F_{\text{expl}}$  is feasible, the current partial solution cannot be rounded to a feasible solution and previously added matchings are removed.

### A2. Recomputation of Min-Marginals

In order to obtain a rounded primal solution we repeatedly recompute the min-marginals after a certain number of calls of the primal heuristic. To this end, we make a trade-off between computation time and quality of min-marginals and achievable solutions. For  $k$  being the total number of calls of the primal heuristic, we compute the threshold  $\alpha = 0.2 \cdot \min(|F_X|, |F_Y|)$ , and whenever we have added  $k \cdot \alpha$  product triangles to the solution, we re-compute the min-marginals. For that, we first fix variables in (ILP-SM), then dualize (ILP-SM), and eventually solve the dual problem again.

### A3. Comparison to Windheuser et al.

**Reimplementation of Windheuser et al.'s approach.** Windheuser et al. tackle the ILP formulation (ILP-SM) through an LP-relaxation, for which variables are gradually

rounded to binary values and then kept fixed. This process is repeated until all constraints are fulfilled. The LP-relaxation reads

$$\min_{\Gamma \in [0,1]^{|F|}} \mathbb{E}^\top \Gamma \text{ s.t. } \begin{pmatrix} \pi_X \\ \pi_Y \\ \partial \end{pmatrix} \Gamma = \begin{pmatrix} \mathbf{1}_{|F_X|} \\ \mathbf{1}_{|F_Y|} \\ \mathbf{0}_{|E|} \end{pmatrix}. \quad (\text{A2})$$

In Algorithm 1 we sketch our re-implementation of the approach by Windheuser et al., which follows the procedure explained in [5]. For solving the LP-relaxation we use the state-of-the-art LP-solver Gurobi [3].

---

**Algorithm 1:** Solving (ILP-SM) according to Windheuser et al.

---

**Input:** (A2)

**Output:** Solution  $\Gamma \in \{0, 1\}^{|F|}$

- ```

1 while Constraints not fulfilled do
2   Solve LP A2 (while keeping already set
   elements of  $\Gamma$  fixed);
3   if  $\Gamma_i > 0.5$  then
4     Fix  $\Gamma_i = 1$ ;
5   end
6 end

```
- 

**Test setup.** In Fig. 6 of the main paper we quantitatively compare our solver to the approach by Windheuser et al. We complement these results with Fig. A2, where we show the corresponding curves for the individual classes. In this experiment for each shape matching instance we first run our method, and afterwards run the approach by Windheuser et al. with a fixed time budget. This is implemented by tracking the total time of Algorithm 1 (including within the LP-solver itself), and once the limit is reached the algorithm is terminated and the current (possibly partial) solution is used as matching.

In all experiments we allow the method of Windheuser et al. to use  $10\times$  more time than ours – even with this generous time budget the method by Windheuser et al. is often not able to produce a complete matching.

#### A4. Partial Shape Matching

While the original formalism (ILP-SM) assumes that the shapes do not have a boundary, we propose a simple yet effective way of dealing with partial shapes. To this end, we simply close existing holes with triangular patches and then compute our energy for each product triangle that involves a ‘hole-triangle’. In Fig. A3 we show an example of shapes with closed holes and their corresponding shapes with holes. We note that in all other visualizations we do not show the closed holes but highlight all shape boundaries.

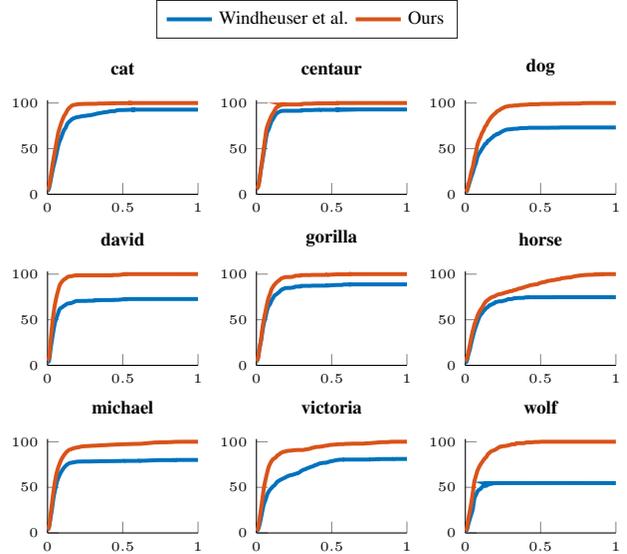


Figure A2. Comparison of the percentage of correct matchings for different shape classes of the TOSCA dataset. The horizontal axis shows the geodesic error threshold, and the vertical axis shows the percentage of matches that are smaller than or equal to this error. We reduce all shapes to 175 triangles. For Windheuser et al. we allow the solver to take  $10\times$  more time than our method needed. The curves by Windheuser et al. are lower because often it only finds only few matchings within the given time budget.

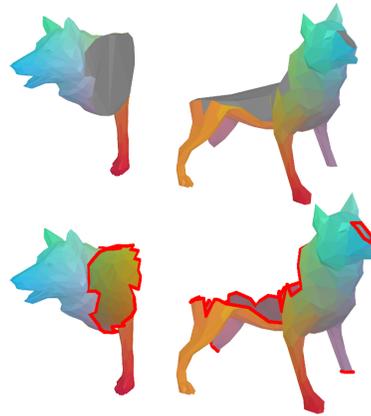


Figure A3. Matching of partial wolf shapes with our approach. At the top, we additionally show the closed shapes (hole-closing triangles are plotted in gray).

#### A5. Additional Results

**Additional qualitative results.** In Fig. A4 we show additional qualitative matchings of the method by Eisenberger et al. [2], Ren et al. [4] and ours. The experimental setting corresponds to Fig. 5 of the main paper.

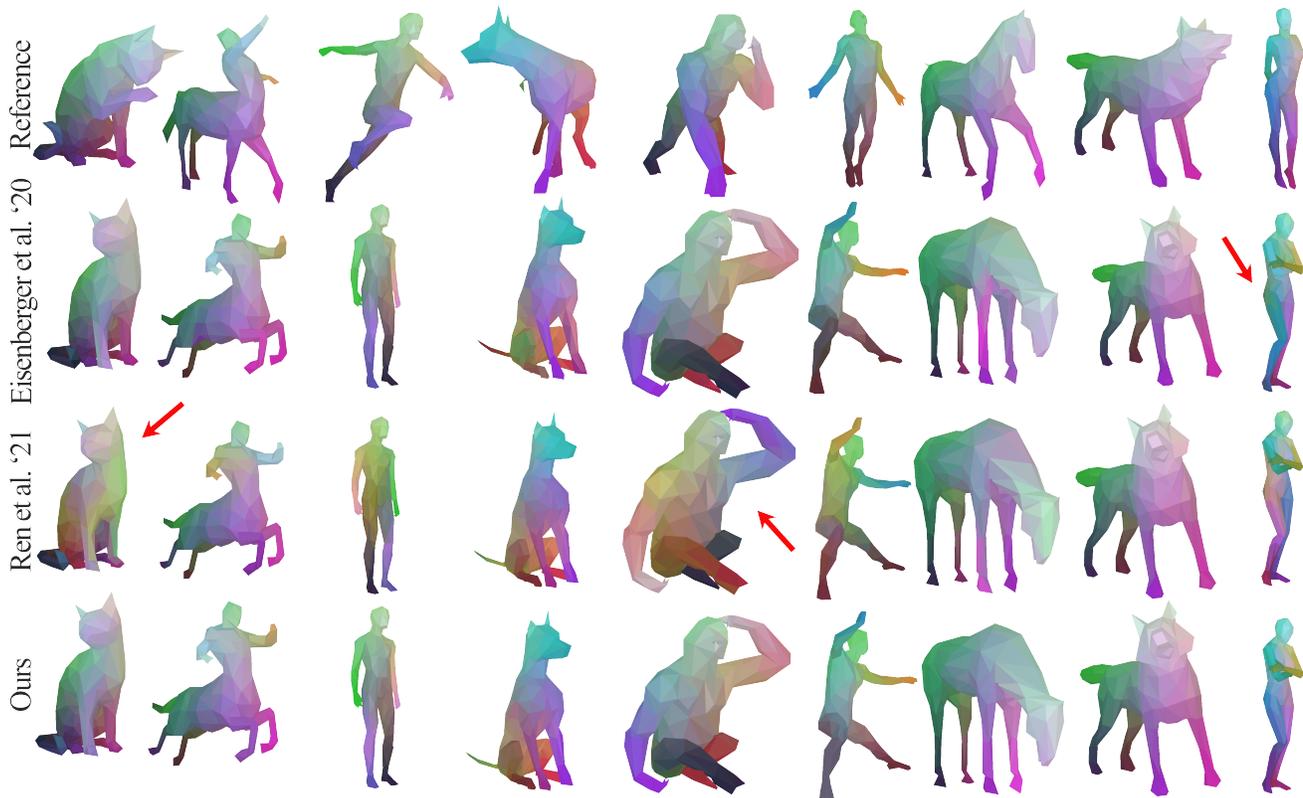


Figure A4. **Qualitative comparison** of the method by Eisenberger et al. [2] (second row), Ren et al. [4] (third row) and Ours (last row) on the TOSCA dataset.

**Additional quantitative results.** In Fig. A8 we show error curves for individual shape classes of TOSCA and KIDS dataset.

**Shape resolution and discretization.** A strong advantage of the utilized discrete matching model is that by allowing for degenerate matchings (triangle-vertex and triangle-edge matchings) it can handle different shape resolutions and discretizations. In Fig. A5 we show this for shapes with different discretisation (left), as well as for substantially varying resolution (right, factor of  $\approx 3\times$  more triangles).



Figure A5. Matching shapes with different discretisation for two shape pairs.

In Fig. A6 we show additional partial-to-partial results with shapes of different mesh resolution (factor of  $\approx \sqrt{2}$ ) for different levels of partiality.

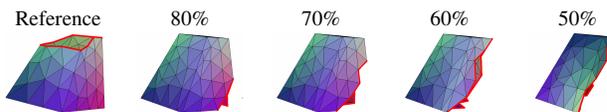


Figure A6. Matching shapes with different levels of partiality with different mesh resolution. At the partiality level of around 50% of the original shape no plausible matching can be found anymore.

**Non-isometries.** The discrete diffeomorphism implemented by the *constraints* in (ILP-SM) can also handle non-isometries, which we show in Fig. A7 on the SHREC'20 dataset [1].



Figure A7. Matching two different non-isometric shape pairs of the SHREC'20 dataset.

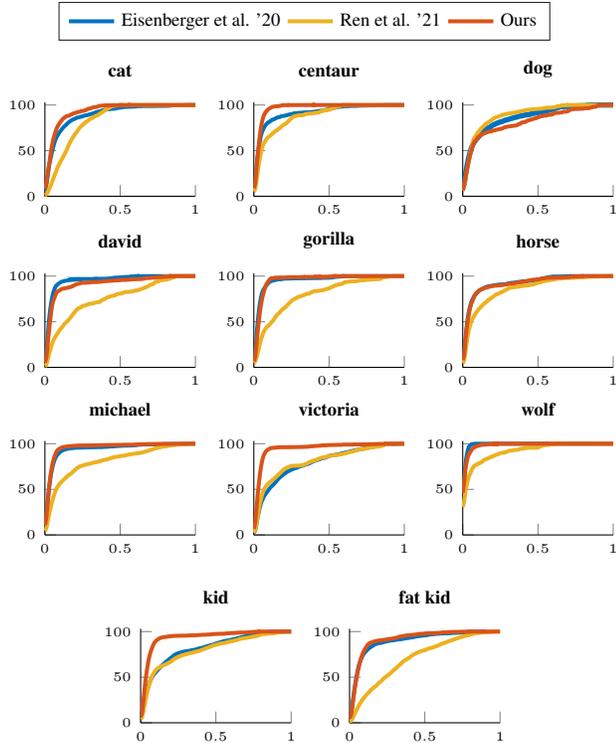


Figure A8. PCK curves for individual shape classes of **TOSCA dataset** (first three rows) and **KIDS dataset** (last row). The horizontal axis shows the geodesic error threshold, and the vertical axis shows the percentage of matches smaller than or equal to this error.

**Texture transfer.** In Fig. A9 we illustrate that the matchings computed with our method can be used for texture transfer.



Figure A9. Texture transfer based on correspondences computed with our method.

**Error cases.** In Fig. A10 we show some failure modes of our method. For the partial-partial dog shown in Fig. A10a ours was correctly initialized (correct matching head), but was not able to determine the remaining part of matching appropriately. This could possibly be accounted for by considering a tighter relaxation. For the partial-partial matching of the ‘Victoria’ shape in Fig. A10b, the overlapping areas of both shapes are too small, so that finding a proper

matching is extremely challenging. The matching of the dog in Fig. A10c failed due to a wrong initialization. This may happen if the total min-marginals do not clearly indicate which matching of initial matchings are best suited. As illustrated by these examples, there are some cases in which our method may fail. Nevertheless, as the quantitative experiments in the main paper indicate, overall our obtained matchings improve upon several existing state-of-the-art methods for non-rigid shape matching.

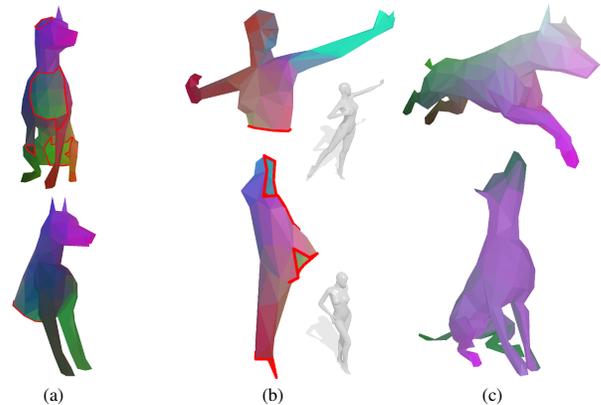


Figure A10. Some failure modes of our approach.

## References

- [1] Roberto M. Dyke, Yu-Kun Lai, Paul L. Rosin, Stefano Zappalà, Seana Dykes, Daoliang Guo, Kun Li, Riccardo Marin, Simone Melzi, and Jingyu Yang. SHREC’20: Shape correspondence with non-isometric deformations. *Computers & Graphics*, 92:28–43, 2020. 3
- [2] Marvin Eisenberger, Zorah Lahner, and Daniel Cremers. Smooth Shells: Multi-Scale Shape Registration With Functional Maps. In *CVPR*, 2020. 2, 3
- [3] Gurobi Optimization, LLC. Gurobi optimizer reference manual, 2021. 2
- [4] Jing Ren, Simone Melzi, Peter Wonka, and Maks Ovsjanikov. Discrete Optimization for Shape Matching. *Computer Graphics Forum*, 40(5):81–96, Aug. 2021. 2, 3
- [5] Thomas Windheuser, Ulrich Schlickewei, Frank R. Schmidt, and Daniel Cremers. Geometrically consistent elastic matching of 3D shapes: A linear programming solution. In *ICCV*, 2011. 1, 2