Robust Shape Collection Matching and Correspondence from Shape Differences

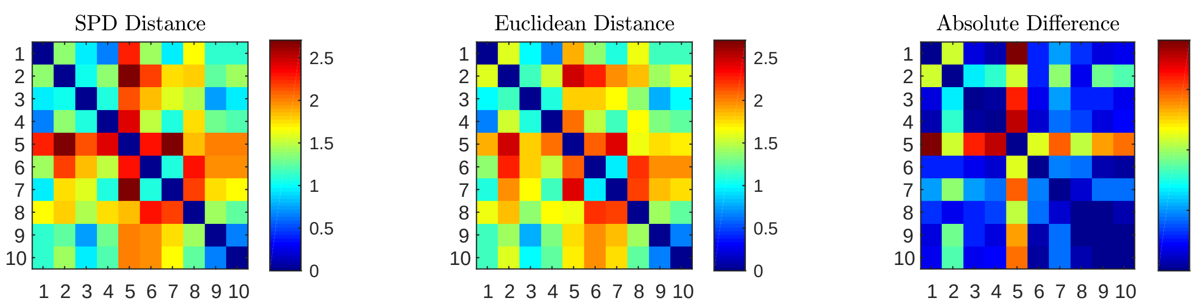
Supplemental Material

Aharon Cohen Mirela Ben-Chen

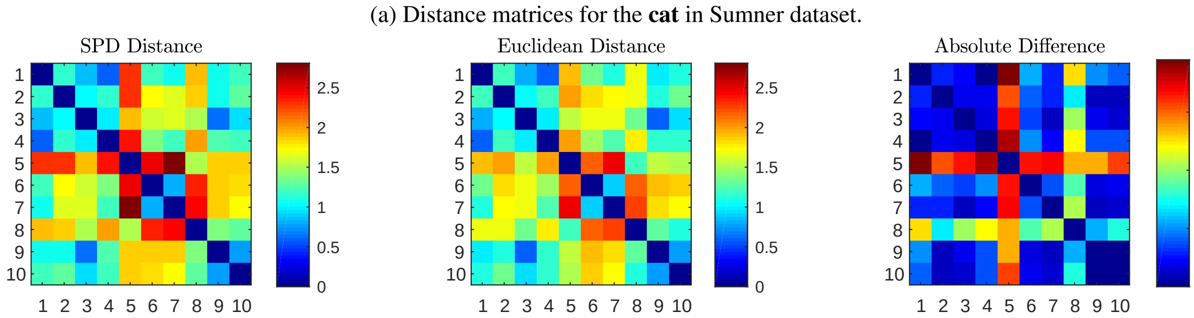
1 SPD Riemannian Distance vs. Euclidean Distance

We first explain why it is beneficiary to take advantage of the fact that the area-based shape difference operators are Symmetric Positive-Definite (SPD) matrices, by using the expression for the Riemannian distance on this manifold (Equation (3) in the main paper, [Bhatia, 2009, Eq. (6.14)]).

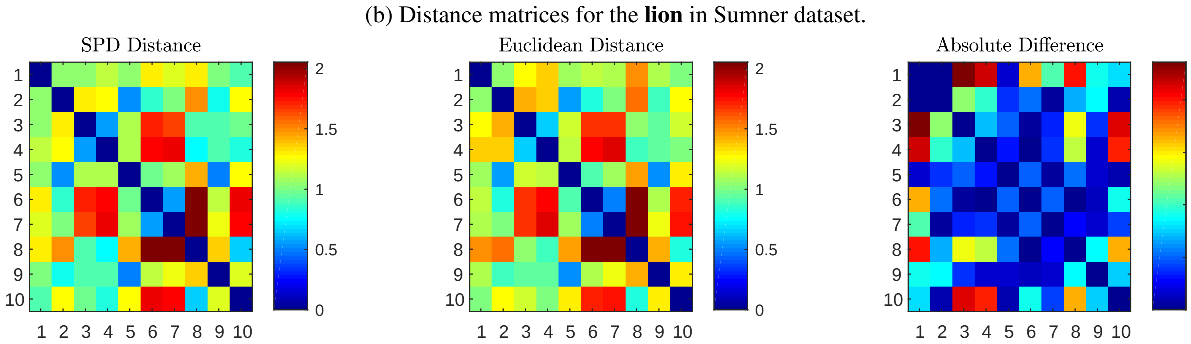
Figure 1 shows the distance matrices computed using SPD distance vs. Euclidean distance (using the same color-bar) for various collections. The difference matrix (absolute value) between the two distance matrices is shown as well. It is clear that we get different results for the two distance types. Note that the largest difference between the two matrices is around 15% of the distance. Therefore, we conclude that Euclidean distance is an *approximation* since it does not take into account the manifold curvature. Thus, we would rather use the SPD distance in order to avoid approximations. It is important to mention that it is easy to compute and does not increase our algorithm's timing. Also note that we get different distance matrices for the cat and lion from Sumner dataset (Figure 1a, 1b) compared to those in the main paper (Figure 4) since here we use the area-based shape differences rather than the conformal.



(a) Distance matrices for the **cat** in Sumner dataset.



(b) Distance matrices for the **lion** in Sumner dataset.



(c) Distance matrices for **various people in the same pose** from FAUST dataset.

Figure 1: Distance matrices computed using SPD Riemannian distance, using Euclidean distance and their difference (absolute value).

Due to the reason that a collection of different people in the same pose (Figure 8 in the main paper) has mainly non area preserving variations, we could test the effect of distance type choice on the algorithm's performance. Figure 2 demonstrates the *matching accuracy* and *normalized alignment error* (as in Figure 9 in the main paper) for matching collections of various people in the same pose. The result was averaged over all possible collection combinations (45 in total) and plotted vs. the shape space dimension. Using SPD distance leads to higher matching accuracy and lower alignment error. Thus, we can indeed see that using SPD distance gives slightly, however consistent, better performance than the Euclidean distance approximation.

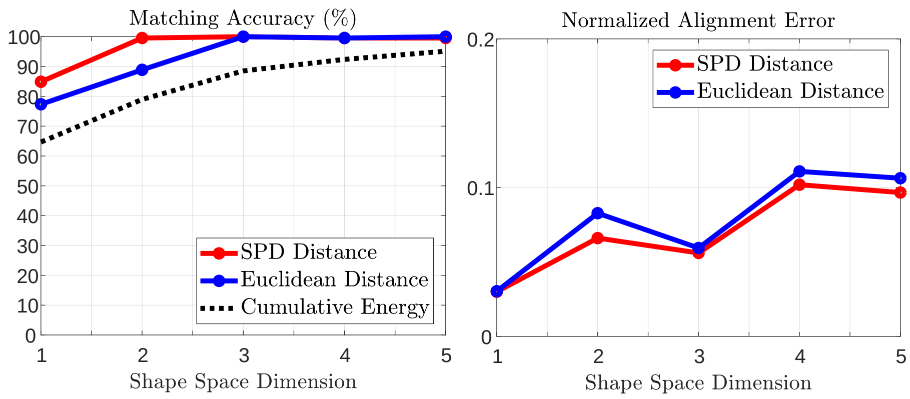


Figure 2: Matching accuracy and normalized alignment error vs. shape space dimension for SPD and Euclidean distance for various people in the same pose from FAUST dataset.

2 Comparison to SBC14

The goal of this section is to give further comparison of our method and [Shapira and Ben-Chen, 2014] (SBC14).

2.1 Algorithm Outline Comparison

We first give an outline of each method and emphasize the differences (in **bold**).

Our Algorithm:

1. Use the input intra-maps to construct the shape differences between the input shapes in the same collection.
2. Use the shape differences to construct a low-dimensional shape space embedding for both collections:
   (a) Choose a base shape **randomly**.
   (b) Compute the shape difference operator for each shape with respect to the **randomly chosen** base shape.
   (c) Construct the shape space by embedding the shape difference operators in a low dimension space by **linear** dimensionality reduction using **SPD distance and multidimensional scaling (MDS)** [Mead, 1992].
3. Align the two shape spaces using **Procrustes analysis (by convex semidefinite programming (SDP) relaxation [Maron et al., 2016])** to obtain the matching pairs and **automatically determine an *optimal base shape* in each collection**.
4. **For each collection, recompute the shape difference operators with respect to the new base shape.**
5. Use the matches and the base shapes to compute a functional inter-map for the base shape pair by solving an optimization problem with an analogies term and **a regularization term**.
6. **Recover a point-to-point inter-map from the functional map using [Ezuz and Ben-Chen, 2017]**.
7. **Optional: Refine the map obtained in (6) using reversible harmonic maps (RHM) [Ezuz et al., 2019]**.

[SBC14]'s Algorithm:

1. Use the input intra-maps to construct the shape differences between the input shapes in the same collection.
2. Use the shape differences to construct a low-dimensional shape space embedding for both collections:
   (a) Choose a base shape using **shape irregularity index**.
   (b) Compute the shape difference operator for each shape with respect to the **computed** base shape.
   (c) Construct the shape space by embedding the shape difference operators in a low dimension space by **non-linear** dimensionality reduction using **diffusion maps**.
3. Align the two shape spaces using **affine registration (by coherent point drift (CPD)** to obtain the matching pairs.
4. Use the matches and the base shapes to compute a functional inter-map for the base shape pair by solving an optimization problem with an analogies term **only**.

Note that:

- Computing **shape irregularity index** for all shapes in both collections is computationally costly compared to random base shape choice and recomputing the shape difference operators using the optimal base shape.
- **Non-linear** dimensionality reduction is sensitive to the collection size and requires large collections in order to obtain fair results.
- Our alignment method is parameter free in contrast to CPD used in SBC14.
- Solving the optimization problem without a **regularization term** leads to poor and unstable results.
- Recovering a **pointwise map** (steps 6-7 in our method) was not applicable using [SBC14] (see next section).

2.2 Pointwise Map Comparison

In the paper, we compared the *functional map* obtained using our algorithm to the map obtained by SBC14. The results appear in Figure 11 in the main paper, and show that we achieve much better results, very similar to the ground truth.

Now we would also like to compare the *pointwise* map can be extracted from the functional map. Figure 3 compares the obtained point-to-point map using our method and using SBC14, with and without RHM refinement on some shapes from FAUST dataset. It can be seen that SBC14 cannot achieve a fair map, not even as an input for RHM post-processing. Hence, we conclude that the method of SBC14 lacks the fine tuning needed to obtain comparable pointwise maps to state-of-the-art methods. Therefore, it is not included in the main paper.

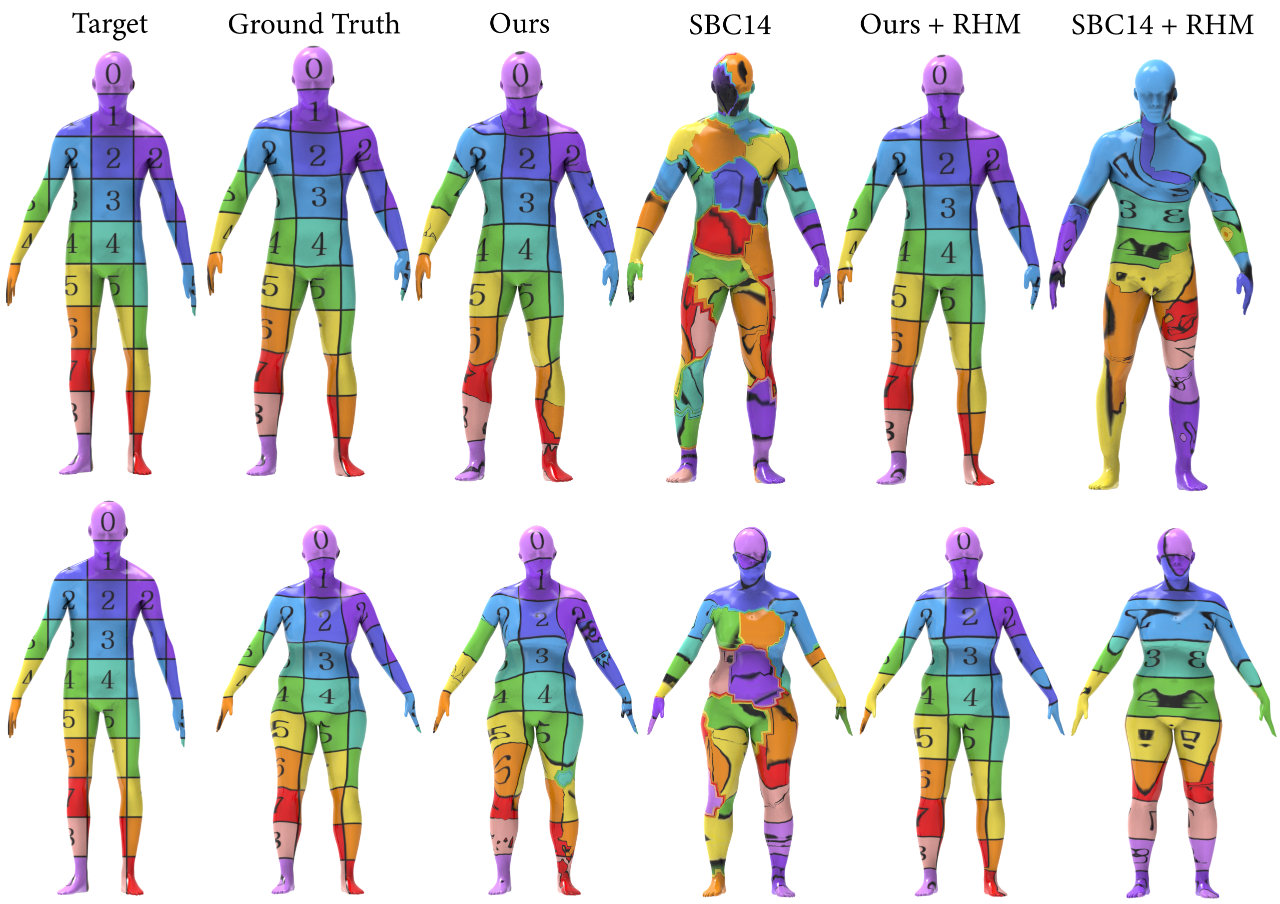


Figure 3: FAUST - qualitative comparison with SBC14, with and without final refinement using RHM.

References

[Bhatia, 2009] Bhatia, R. (2009). *Positive definite matrices*, volume 16. Princeton university press.

[Ezuz and Ben-Chen, 2017] Ezuz, D. and Ben-Chen, M. (2017). Deblurring and denoising of maps between shapes. In *Computer Graphics Forum*, volume 36, pages 165–174. Wiley Online Library.

[Ezuz et al., 2019] Ezuz, D., Solomon, J., and Ben-Chen, M. (2019). Reversible harmonic maps between discrete surfaces. *ACM Transactions on Graphics*.

[Maron et al., 2016] Maron, H., Dym, N., Kezurer, I., Kovalsky, S., and Lipman, Y. (2016). Point registration via efficient convex relaxation. *ACM Transactions on Graphics (TOG)*, 35(4).

[Mead, 1992] Mead, A. (1992). Review of the development of multidimensional scaling methods. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 41(1):27–39.

[Shapira and Ben-Chen, 2014] Shapira, N. and Ben-Chen, M. (2014). Cross-collection map inference by intrinsic alignment of shape spaces. In *Computer Graphics Forum*, volume 33, pages 281–290. Wiley Online Library.