A. MCMC - pseudo code

In Section 5 we already gave a rough description of our MCMC initialization algorithm. Here, we provide a more detailed pseudo code:

Algorithm 3. *(MCMC)*

```
1. Initialize $\tau_{\text{best}} := 0, \mathbf{X}^*_{\text{best}} := \mathbf{X}$.
2. For $i = 1, ..., N_{\text{prop}}$:
    2.1 Sample new proposal $\tau_{\text{prop}} \sim \mathcal{N}(0, \mathbf{I})$.
    2.2 Compute the current alignment $\mathbf{X}^*_{\text{prop}}$ by making
        a surrogate run with the initial guess $\tau_{\text{prop}}$.
    2.3 Compute the acceptance probability $\alpha := \exp\Big(-\frac{1}{2\sigma^2_{\text{match}}}\big(E(\mathbf{X}^*_{\text{prop}}) - E(\mathbf{X}^*_{\text{best}})\big)\Big)$ using
        the energy $E$ from Eq. (12).
    2.4 Sample $u \sim \text{Unif}(0,1)$ and either accept or reject
        the new sample $\tau_{\text{prop}} \in \mathbb{R}^{K_{\text{init}} \times 3}$:
        $(\tau_{\text{best}}, \mathbf{X}^*_{\text{best}}) := \begin{cases} (\tau_{\text{prop}}, \mathbf{X}^*_{\text{prop}}), & u \leq \alpha \textit{ (accept)} \\ (\tau_{\text{best}}, \mathbf{X}^*_{\text{best}}), & u > \alpha \textit{ (reject)} \end{cases}$
```

We usually set the number of surrogates to $N_{\text{prop}} := 100$. In the majority of cases in our experiments this is more than sufficient. Furthermore, we usually choose a small objective variance $\sigma^2_{\text{match}} := 0.001$ to get a sharp distribution and therefore more accurate samples τ.

Remarks One aspect of our method that we did not talk about yet is how to compute a good initial rigid pose. For most datasets in our experiments this is a requirement, e.g. SHREC'19 [34] connectivity has random rigid poses for all inputs. In theory, our MCMC algorithm can account for rigidly displaced inputs $\mathcal{X}$ and $\mathcal{Y}$ but in practice our $N_{\text{prop}} = 100$ surrogates are not enough for extreme cases. Therefore, we initially apply another surrogate based method that initializes with different rigid poses and determines the best one according to the objective E from Eq. (12). A thorough description of this is beyond the scope of this paper, but all the details can be found in our implementation.

B. Proof of Theorem 1

Theorem 1 gives an upper bound on how much the geometry of our smooth shells can change between two states K and $K+1$. For spectral reconstruction, a projection on a new eigenfunction is added in each iteration. Depending on the magnitude of the new projection $\big(\phi_{K+1} \otimes \phi_{K+1}\big)$, this can lead to arbitrarily high changes :

$$\|\mathcal{T}_{K+1}(X) - \mathcal{T}_K(X)\|_{L^2} = \|\big(\phi_{K+1} \otimes \phi_{K+1}\big)X\|_{L^2}. \quad (14)$$

In comparison, Theorem 1 states that the change from $\mathcal{S}_K(X)$ to $\mathcal{S}_{K+1}(X)$ can be bounded by choosing a small upsampling variance σ.

Proof. We will proof the statement for scalar functions $X \in L^2(\mathcal{X})$. The extension to vector valued functions $L^2(\mathcal{X}, \mathbb{R}^3)$ is trivial – we just need to apply the identity to each component at a time. Now let $K > 0$ and $\sigma > 0$. For brevity we will denote the sigmoid weights with $s_k^K := \frac{1}{1+\exp\big(\sigma(k-K)\big)}$. Using the spectral decomposition of operators, we can deduce:

$$\begin{aligned}
&\big\|\mathcal{S}_{K+1}(X) - \mathcal{S}_K(X)\big\|^2_{L^2} = \\
&\sum_{k=1}^{\infty}\big|\big(s_k^{K+1} - s_k^K\big)\big\langle\phi_k, X\big\rangle_{L^2}\big|^2 = \\
&\sum_{k=1}^{\infty}\Big|\bigg(\big(s_k^K\big)^{-1} - \big(s_k^{K+1}\big)^{-1}\bigg)s_k^{K+1}s_k^K\big\langle\phi_k, X\big\rangle_{L^2}\Big|^2 = \\
&\sum_{k=1}^{\infty}\big|(1-e^{-\sigma})\exp\big(\sigma(k-K)\big)s_k^{K+1}s_k^K\big\langle\phi_k, X\big\rangle_{L^2}\big|^2 \leq \\
&\sum_{k=1}^{\infty}\Big|(1-e^{-\sigma})\bigg(1+\exp\big(\sigma(k-K)\big)\bigg)s_k^{K+1}s_k^K\big\langle\phi_k, X\big\rangle_{L^2}\Big|^2 = \\
&\sum_{k=1}^{\infty}\big|(1-e^{-\sigma})s_k^{K+1}\big\langle\phi_k, X\big\rangle_{L^2}\big|^2 = \\
&\big|(1-e^{-\sigma})\big|^2\big\|\mathcal{S}_{K+1}(X)\big\|^2_{L^2}.
\end{aligned}$$

Taking the square root on both sides then yields the desired identity. □

Remarkably, this bound is independent of the index K. Small eigenfunctions ϕ_k typically represent coarse structures like limbs. Therefore, in particular the first iterations using spectral reconstruction lead to big changes in the geometry, see Eq. (14).

C. Runtime Analysis

We analyze the time complexity of our method in comparison to other popular matching methods in Figure 9. In particular, we compare the runtime of the whole pipelines for instances of the same pair of Michael shapes from the TOSCA dataset that was remeshed to different resolutions between 500 and $50k$ vertices.

D. Additional Qualitative Evaluations

We provide some additional qualitative evaluations and comparisons of our pipeline in order to give the reader a better understanding about the merits of our method, see Figure 10. Additionally, we provide a failure case in Figure 11. Our method is deformation based with an as-rigid-as-possible assumption. This means that in places of

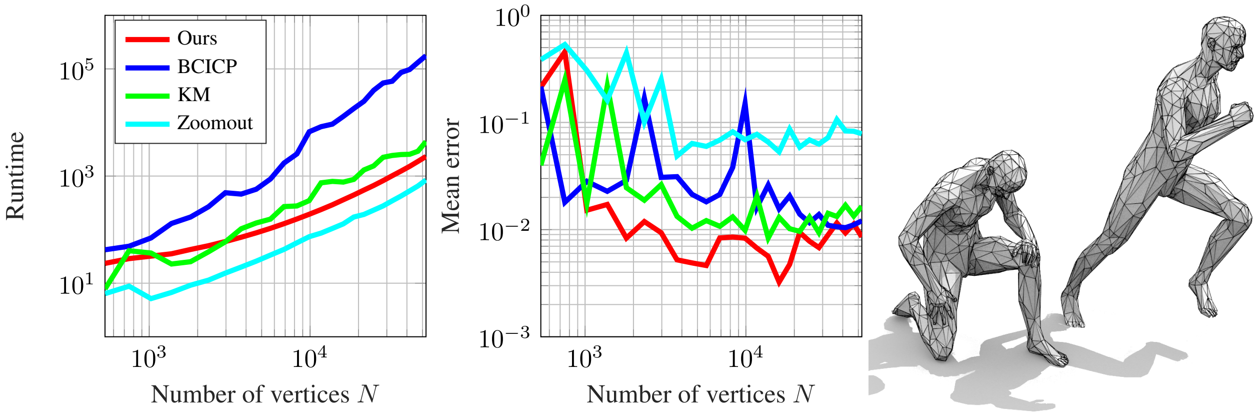

Figure 9: We compare the runtimes of our method with BCICP [42], Kernel matching [56] and Zoomout [35]. To this end, we remesh the Michael shape from TOSCA to different resolutions, on the right side we display the pair for $N = 1000$. Besides the runtime we also compare the matching accuracies of all methods. Here, our method is the most accurate one and stable across resolutions, whereas our runtime is the second best after Zoomout.

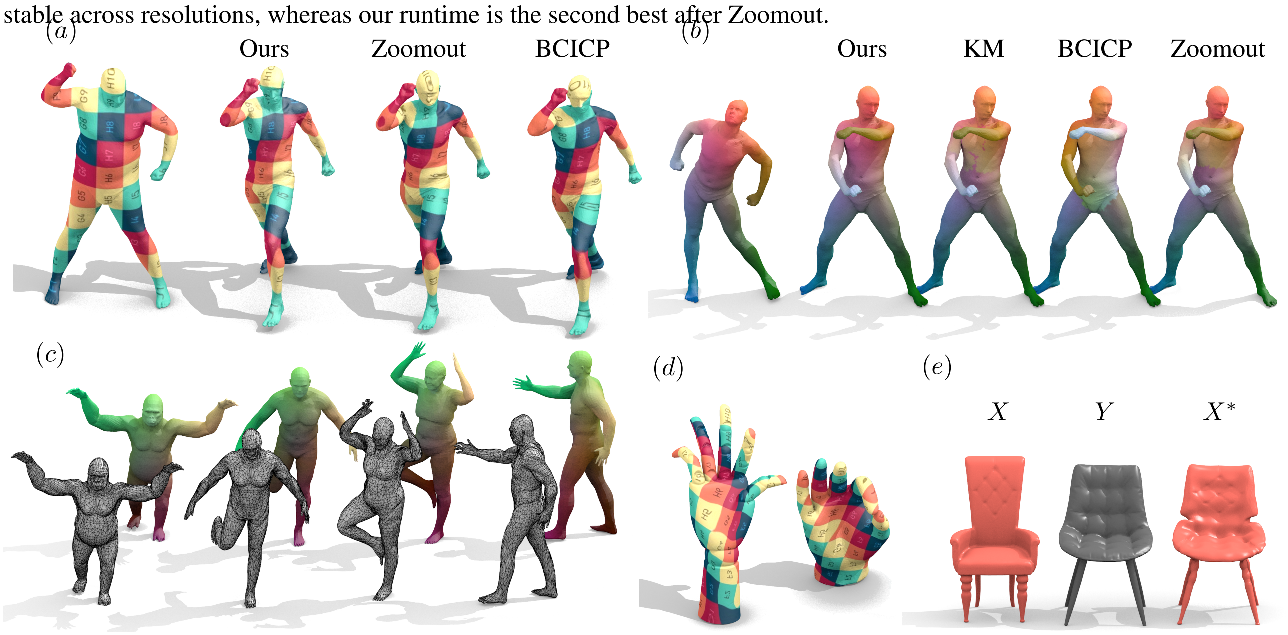

Figure 10: Here, we show additional qualitative evaluations of our method. (a) and (b) are comparisons for a pair from the SHREC'19 and the SCAPE dataset respectively. (a) is challenging due to an incompatible meshing, (b) has equivalent meshing but is still susceptible to mismatches due to self-similarities (left-right: BCICP, front-back: KM, Zoomout). (c) shows how our method can be used to smoothly transfer meshings for interclass pairs, here for a gorilla from TOSCA to humans from FAUST. The maps are smooth in the sense that local structures are preserved and deformations only occur in the form of uniform, global stretching of parts. I.e. the face still looks like a gorilla after deformation although the rest of the body adapts to the human form. (d) shows a texture transfer from a template hand (right) to a scanned hand of a puppet (left). The latter is a scan of a real world object from [14], obtained with the handheld Space Spider scanner from Artec. This is a challenging example due to different resultions of the inputs, different small scale features and a different size of the residual part at the bottom. (e) shows how our method can be applied to deform an object (red chair) and align it with a reference shape (black chair) to create a new object. The deformed red chair X^* has the global structure of Y and the fine scale details of X.

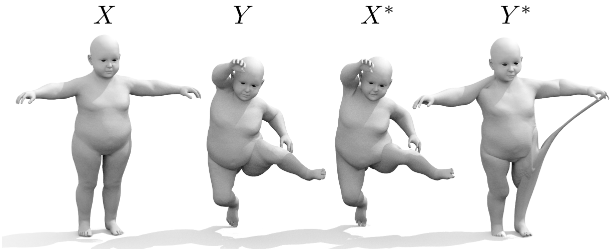

Figure 11: A failure case of our method for a pair of shapes from TOPKIDS. If we align the the canonical pose X with the reference Y we get a meaningful alignment X^* and high quality correspondences. However, if we try to apply our method the other way around, we get undesirable ”cheese pull” effects at the left hand of the deformed kid Y^*. The reason for that is that in the pose Y the fingers touch the left knee and the meshing connects. In order to avoid this effect, we either need a mesh separation policy or use an intermediate template where the original topology is known. We prefer the latter approach in our quantitative evaluations on FAUST and TOPKIDS because finding a meaningful topological cut is a complicated problem on its own.

topological changes the meshing cannot be separated. Our method still tries to align the shape as good as possible with the reference which invariably leads to a ”cheese pull” effect. This is also the main reason why we use an intermediate template to match the FAUST and TOPKIDS shapes in our quantitative evaluations.