STAG: Smart Tools and Applications in Graphics (2021)
P. Frosini, D. Giorgi, S. Melzi, and E. Rodolà (Editors)

Reposing and retargeting unrigged characters with intrinsic-extrinsic transfer

Pietro Musoni[1], Riccardo Marin[2], Simone Melzi[2], Umberto Castellani[1],

[1]University of Verona [2] Sapienza University of Rome

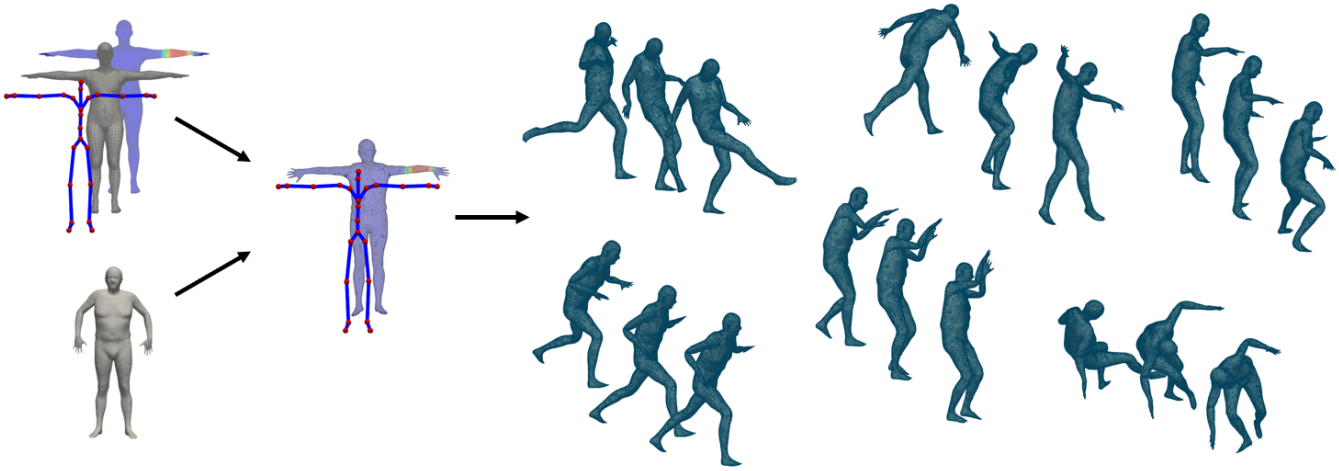

Figure 1: *We can transfer skinning information from a rigged template and recover the rest pose for an arbitrary shape. Then, the model can be animated using any animation already defined for the initial model. Our method let us transfer skinning information regardless of different discretizations or non-isometric deformations.*

Abstract

In the 3D digital world, deformations and animations of shapes are fundamental topics for several applications. The entertainment industry, virtual and augmented reality, human-robot interactions are just some examples that pay attention to animation processes and related tools. In these contexts, researchers from several communities desire to govern deformations and animations of 3D geometries. This task is generally very complicated because it requires several skills covering different kinds of knowledge. For this reason, we propose a ready-to-use procedure to transfer a given animation from a source shape to a target shape that shares the same global structure. Our method proposes highly geometrical transferring, reposing, and retargeting, providing high-quality and efficient transfer, as shown in the qualitative evaluation that we report in the experimental section. The animation transfer we provide will potentially impact different scenarios, such as data augmentation for learning-based procedures or virtual avatar generation for orthopedic rehabilitation and social applications.

CCS Concepts

• ***Computing methodologies*** → ***Animation;*** *Shape analysis;* • ***Theory of computation*** → *Computational geometry;*

1. Introduction

Studying the movement of 3D objects is a central topic for several fields [MZR*20]. First of all, the entertainment industry has produced an impressive interest in moving characters for movies, videogames and virtual/augmented reality applications [DVLP20]. Secondly, the robotic industry has deployed several efforts in study animation and introduced it in a simulated system that aims to be an accurate and pre-training environment for artificial intelligence agents which will interact with the real world [VGK17]. Last but not least, researchers from several communities study 3D geome-

DOI: 10.2312/stag.20211471

tries and their animation to simulate processes, possibly with little effort and without looking at complex systems of a reduced interest for the final application [LOP*18]. Unfortunately, the animation of 3D geometries is far from trivial. This procedure lives at the intersection of two pieces of knowledge: the one about the object to deform and the computational skills to animate it. Each specialized community generates these expertises independently, and only a few experts can control the two masteries, limiting usability and advancements in the fields. For example, in the Virtual Humans field, many researchers focus on studying body appearance, anthropometry, segmentation, or even synthesization of new humans. Still, only a few of them can animate a given human and produce data in new poses (for example, by attaching motion capture). While data-driven approaches have become predominant in recent years, we observed that not all fields had experienced a complete integration of such techniques. 3D Data are not always available, and even when it is the case, they are usually quite limited. Given these difficulties, a practitioner may inquire if such animation work is necessary for each new instance of the aimed class. In other words, is it possible to transfer the dynamic system across different 3D models? For instance, in the animation industry is consolidated the possibility to move the animation sequences among characters that share the same rigging and skinning structure. Here we address the scenarios where also the rigging and skinning properties are changed or not available for the target shape. This task requires to relate geometries which may differ for their embedding in $\mathbb{R}^3$, but also for some of their intrinsic characteristics like details, discretization or even representation (e.g., point clouds versus triangle meshes). Furthermore, several animations system assumes a given initial condition since they encode the motion as a deformation from it (e.g., a human animation produced by applying rotation to its limbs, starting from a neutral T-pose). In this work, we propose to solve the animation transferring using few simple but effective steps. Given a 3D object equipped with rigging and skinning properties and a library of defined animations, we show how to transfer them to a target shape with a similar global structure but not ready to be animated exploiting the available tools. While we frame ourselves on virtual humans animated through the LBS system, our approach is general, and we believe it is applicable in several scenarios. First, we propose to transfer the skinning properties as functions by adopting the functional map framework [OBCS*12]; but instead, to rely on standard function transfer paradigm or point-to-point map, we exploit a new intrinsic-extrinsic technique properly designed to work on skinning weights. Secondly, we implement an optimization process that regresses a given geometry to the desired pose. Finally, we show that the so obtained geometry is ready to inherit the animation from the starting model. Our contribution is threefold:

1. We deploy a system to transfer skinning properties, applied to Linear blend skinning (LBS) weights that is flexible and agnostic to any specific domain characteristics.
2. We propose an optimization to regress to a neutral pose a given shape. The proposed energies are useful for solving unconstrained cases (e.g., rotations that would produce the same extrinsic position of skeleton joints).
3. We release a complete implementation of our method, which will be made publicly available for research purposes and which we believe could find application in different communities.

2. Context

Skinning systems. In literature, several methods have been proposed to model the dynamics of objects through animation frameworks. For an overview of classical methods, we link to the seminal course of [JDKL14]. These methods require defining a relationship between the surface vertices and a limited set of handles. In this way, acting on a limited set of handles impacts large parts of the 3D object without caring for single vertices modification. For example, the cage paradigm [XLG09, XLG12, XLX15, CB17] wraps the mesh into several polyhedrons. Then, moving the corners of the cages, the surface is consequently stretched, translated, and rotated. Another popular approach (and the one that we will use for our experiments) requires defining a skeletal representation of a mesh and then connect surface deformations to joints displacement [MTLT88]. Many variations have been proposed, for example, to avoid artifacts and improve the representation capability of the animation system [KCŽO08]. Once a model is equipped with such properties, the animation is no more than a sequence of modifications of the specified handles. The design choice in this process are many, and in general, any domain adapts the paradigms to its needs. Such effort is not trivial, and a library of animations defined for an object can rarely be directly transferred to another one.

Animation: recycling and retargeting. As a direct consequence, the community has developed a significant interest in this problem. Some works focus on the definition of the skinning properties on a 3D character from scratch, as the important work [BP07]. This approach requires a minimum assumption on the character and a few input data. Although the interest in the transferring of entire animations has been growing, since the animated sequences can be obtained principally by expensive techniques such as motion capture or by the hands of experts 3*D* artists and the idea of reusing existing data on several different characters is extremely appealing. This objective can be achieved by transferring the deformation of the surface, especially between models that do not share a common animation framework. Such process is in general made by 3D registration pipelines [MMM*20, MMRC20] or by transferring, through optimization, the shape of a 3D model to a deformed character [BWBM19]. Another approach is using a point-to-point correspondence as the work [MMM*20] for the transfer of all the skinning properties. These methods, however, require strong assumption (i.e., [MMM*20] requires that the shapes share the same pose) and solving complex optimization or combinatorial problems, which easily fall in local minima. Alternatively, one can transfer the animation framework such that the deformation sequence can be directly applied. An approach is exploiting medial-axis computation [YYG18, SS19] or extracting the skeleton using human anatomy properties [DMMT*07]. However, such representations are computationally expensive, and the majority of rendering engines do not natively support them. Similarly, [LD14] generates automatically linear blend skinning models, ready to be used in common software. On the other hand, it requires a set of deformed meshes that are not always available. The majority of the previous works do not focus on the entire skinning process,

which requires several steps. In [] the authors a method that uses bi-harmonic distance for matching and be able to transfer the skinning weights from a character to another, relying on [BP07] for the skeleton embedding. [AGR*16] and [MF19] propose pipelines of animation transfer based on matching but requires massive user intervention for the initial marker selection.

In this context, our method follows these trends but using significantly less domain-prior knowledge and semantic assumption, both in terms of involved models and skinning framework.

3. Background

This section introduces the fundamental elements of our experimental setting: Functional maps as the frame on which we build our transferring approach, and LBS as the concrete example on which we deploy our method.

3.1. Functional Maps

Functional Maps [OBCS*12] is a state-of-the-art method to solve the point-to-point dense correspondence between two surfaces. The main idea is to interpret a shape $\mathcal{M}$ as non-euclidean support for functions (i.e. a proper functional space $\mathcal{F}_{\mathcal{M}}$). A function f defined on $\mathcal{M}$ is a map $f : x \in \mathcal{M} \to \mathbb{R}$ that associate for each point x of the surface $\mathcal{M}$ a real value. Given two surfaces, solving for a functional map requires to find the map $T_{\mathcal{F}} : \mathcal{F}_{\mathcal{M}} \to \mathcal{F}_{\mathcal{N}}$ where $\mathcal{F}_{\mathcal{M}}$ and $\mathcal{F}_{\mathcal{N}}$ are the functional spaces defined respectively on the two shapes.

This highly complex problem can be efficiently solved if such spaces are equipped with some basis $\Phi_{\mathcal{M}}$ and $\Phi_{\mathcal{N}}$ composed by $k_{\mathcal{M}}$ and $k_{\mathcal{N}} \in \mathbb{N}$ functions respectively. Given the basis $\Phi_{\mathcal{M}}$ every function f defined on $\mathcal{M}$ can be represented as a linear combination of the basis functions that compose $\Phi_{\mathcal{M}}$ and thus compactly encoded by the vector $\boldsymbol{a}_f \in \mathbb{R}^{k_{\mathcal{M}}}$ of the coefficients that define this linear combination. The same holds for every function $g \in \mathcal{F}_{\mathcal{N}}$ and its coefficients $\boldsymbol{b}_g \in \mathbb{R}^{k_{\mathcal{N}}}$. In that case, the functional map $T_{\mathcal{F}}$ is encoded by a matrix $\boldsymbol{C} \in \mathbb{R}^{k_{\mathcal{N}} \times k_{\mathcal{M}}}$ that is the matrix of change of basis and solves the linear system:

$$\boldsymbol{C}\boldsymbol{a}_f = \boldsymbol{b}_g, \tag{1}$$

where g is the function on $\mathcal{N}$ that corresponds to $f \in \mathcal{F}_{\mathcal{M}}$. Given enough corresponding functions $\{(f_1, g_1), (f_2, g_2), \dots\}$ (namely probe functions) on the two surfaces, such linear system is determined. If it is not the case (as commonly happens), it can be reformulated as an optimization problem introducing further regularization of desired properties like commutativity, bijectivity or orthonormality as widely described in the literature [OBCS*12, NO17, RPWO18, RMWO21].

More concretely, in the discrete setting, where a shape $\mathcal{M}$ is represented as a triangle mesh, a common choice for the basis of the functional space $\mathcal{F}_{\mathcal{M}}$ are the set of eigenfunctions of the Laplace-Beltrami Operator (LBO) associated to the eigenvalues with smallest absolute value. This choice is motivated by their theoretical properties; such eigenfunctions provide the smoothest orthonormal basis possible, and they are analogous to the Fourier Basis in the Euclidean domain. Also, they are invariant to isometries.

This functional perspective is strongly related to the correspondence between the two surfaces $\mathcal{M}$ and $\mathcal{N}$. Given two discrete surfaces $\mathcal{M}$ and $\mathcal{N}$ with with $n_{\mathcal{M}}$ and $n_{\mathcal{N}}$ vertices respectively, we can encode the correspondence between them as a matrix $\boldsymbol{\Pi} \in \mathbb{R}^{n_{\mathcal{N}} \times n_{\mathcal{M}}}$ such that $\Pi(i, j) = 1$ if the j-th vertex of $\mathcal{M}$ corresponds to the i-th vertex of $\mathcal{N}$ and $\Pi(i, j) = 0$ otherwise. Once we have this matrix we can explicitly compute the functional map $\boldsymbol{C}$:

$$\boldsymbol{C} = \Phi_{\mathcal{N}}^{\dagger} \Pi \Phi_{\mathcal{M}}, \tag{2}$$

where $\Phi_{\mathcal{N}}^{\dagger}$ is the Moore–Penrose pseudo-inverse of Φ_N. Thanks to this equation, it is possible to convert a functional map estimated through equation 1 plus additional regularizations into a dense point-to-point correspondence simply by solving the nearest neighbor search as described in [OBCS*12]. For this reason, the framework of functional maps constitutes an efficient and effective pipeline for the estimation of dense point-to-point correspondences between deformable shapes. This approach has been extensively investigated in the last decade, giving rise to a large set of alternatives such as [RMC15, NO17, NMR*18, EBC17, MRR*19, RMOW20, PRM*21] among many others.

3.2. Linear Blend Skinning

Deforming a discrete shape $\mathcal{M}$ requires defining a new position for each of its $n_{\mathcal{M}}$ vertices v. Such freedom of choice is, in general, significantly greater than required. A common approach to limiting such variability to improve usability is introducing an animation framework that regularises the space of choices, providing intuitive handles to modify entire portions of the shape at once. Formally, the new vertices positions v' is given by:

$$v' = v + G(\Theta), \tag{3}$$

where $G : \Theta \to \mathbb{R}^{n_{\mathcal{M}} \times 3}$ is a function that given a relatively small set of parameters Θ produces the desired offset. Among the possibilities, a popular schema for articulated characters is Linear Blend Skinning (LBS) paradigm. LBS approach requires to equip the shape $\mathcal{M}$ with two elements: a set of $Q \in \mathbb{N}$ joints $\mathcal{J} \in \mathbb{R}^{Q \times 3}$ which are also called *skeleton* sorted as a tree structure, and the skinning weights matrix $\mathcal{W} \in \mathbb{R}^{Q \times n_{\mathcal{M}}}$; its rows quantify the impact of joints rotations to the surface vertices. The vertex displacement proceeds as follows. First, the parameters associated with each joint is converted using the Rodrigues formula to a matrix rotation:

$$R_j(\theta, \boldsymbol{J}) = \prod_{l \in A(J_q)} \left[\begin{array}{c|c} \mathrm{rodr}(\theta_l) & j_l \\ \hline 0 & 1 \end{array}\right] \tag{4}$$

where j_l is the joint position and $A(J_q)$ are the ancestors of the joint J_q in the kinematic tree $\forall l, q \in \{1, \dots, q\}$. Then:

$$R'_q(\theta, \boldsymbol{J}) = R_q(\theta, \boldsymbol{J}) R_q(\theta^*, \boldsymbol{J})^{-1} \tag{5}$$

$$\bar{v_i} = \sum_{q=1}^{Q} w_{i,q} R'(\theta, \boldsymbol{J}) v_i, \tag{6}$$

produce the new configuration for each vertex, where $R_q(\theta^*, \boldsymbol{J})$ is the transformation of joint q in the world frame, and $R'_q(\theta, \boldsymbol{J})$ is the

same after removing the transformation induced by the rest pose θ^*. We refer to [LMR*15] for further details.

3.3. Skinning weights transfer

As introduced in Section 2, there are different possibilities to transfer the skinning weights from the source shape to a target shape. Two of them come directly from the framework of functional maps. The third one, the one that we propose, requires more task-specific adjustments that we present in this paper for the first time.

Each of the skinning weight w_q, $\forall q \in \{1,\dots,Q\}$ of the source shape $\mathcal{M}$ could be represented by a function defined over the surface and in the discrete setting as a vector with a value for each vertex, that is $w_q \in \mathbb{R}^{n_\mathcal{M}}$ and $w_{i,q}$ is the value of the function w_q at the i-th vertex in the list of vertices of $\mathcal{M}$. Thanks to the framework of functional maps, we can 1) project this function in the space of coefficients, transfer these coefficients using the functional map $\boldsymbol{C}$, and then reconstruct the function on the target shape by linearly combining the target basis functions with the transferred coefficients. We refer to this solution as the functional solution. Even if this method is very efficient and simple, it does not provide very accurate results due to the limited representation quality of the Fourier-like basis provided by the truncated subset.

As we saw in the previous section, the functional map $\boldsymbol{C}$ can be converted in a point-to-point map by solving a nearest neighbor search or one of its alternatives as described in [PRM*21]. The point-to-point map allows us to transfer the skinning weights just assigning at each vertex of the target shape the value assumed by the skinning function on the correspondent vertex of the source shape. In other words, if $\boldsymbol{\Pi}_{j,i} = 1$, and thus the j-th vertex of the target shape corresponds to the i-th vertex of the source shape, then we define the value of the q-th skinning weight on the target shape $\mathcal{N}$ at the vertex j-th equal to the value of the original q-th skinning weight on the source shape $\mathcal{M}$ at the vertex i-th, that is $w^{\mathcal{N}}_{j,q} = w^{\mathcal{M}}_{i,q}$. We refer to this approach as the point-to-point transfer or, more briefly, p2p.

Usually, each of these skinning weights has a value different from 0 only in a localized region of the surface, as shown in Figure 4. These kinds of localized functions are not easy to represent with the smallest subset of the Laplace-Beltrami eigenfunctions. Moreover, even if the functional map has a high quality, the point-to-point map extracted from it is not very smooth, and thus it generates a skinning weight with some noise and blur. As described in the literature, this problem can be alleviated by a refinement process [EBC17, MRR*19], but not completely solved.

For this reason, we implement an alternative solution, that as we will show in the following section, provides more accurate results at a minimum computational additional cost.

4. Proposed method

Here we report the main steps of our method. A complete overview is provided in Figure 2.

4.1. Intrinsic-extrinsic transfer of the skinning weights

Given the problem presented in the previous Section, we propose a novel approach for transferring the skinning weights between shapes, not in the same pose. We assume as input a template $\mathcal{T}$ equipped with skinning information, a non-rigged target shape $\mathcal{S}$, a (given) functional map $\boldsymbol{C}$ that maps functions defined on $\mathcal{T}$ to the functional space of $\mathcal{S}$. We show how to transfer the skinning weights $\mathcal{W}_\mathcal{T}$ from $\mathcal{T}$ to $\mathcal{S}$, while we do not see any constraint to do not apply our method on other properties (i.e., skeleton). Our approach relies on two main observations. Firstly, the skinning weights can be seen as functions highly localized. Hence, we require a method that guarantees to transfer only to local regions of the mesh, without the risk of diffusing the information elsewhere. Secondly, functional maps are agnostic to the pose but produce a low-pass transfer. In contrast, spatial transfers (i.e., nearest neighbor association) require rigid alignment but provide perfect transfer, preserving all the original information. Hence, our strategy is composed of four steps. First of all, we obtain the coefficient representation for the coordinates functions for the template shape:

$$\mathbf{x}_\mathcal{T} = \Phi^\dagger_\mathcal{T} X_\mathcal{T}, \tag{7}$$

where $X_\mathcal{T}$ is the matrix $n_\mathcal{T} \times 3$ containing the vertex coordinates of the shape. Hence, $x_\mathcal{T}$ has the dimensions $k_\mathcal{T} \times 3$, where $k_\mathcal{T}$ is the number of LBO eigenfunctions selected on $\mathcal{T}$. Then, we use the given functional map $\boldsymbol{C}$ to transfer the coordinates in the functional space of the source shape:

$$\widetilde{\mathbf{x}_\mathcal{T}} = \boldsymbol{C}\mathbf{x}_\mathcal{T}. \tag{8}$$

Namely, $\widetilde{\mathbf{x}_\mathcal{T}}$ are the coefficients of the coordinates of $\mathcal{T}$, in the functional space of $\mathcal{S}$. It is worth noting that the only assumption required is that $\boldsymbol{C}$ matrix transfer the function properly. However, such an assumption is particularly weak considering that $\boldsymbol{C}$ can be efficiently and automatically estimated, with few or no user input depending on the context. Also, in some particular domains, the availability of data lets us obtain high-quality functional maps with deep learning techniques such as coefficients let us recover the geometry of the target but with the discretization of the template. If this holds, then the coefficients $\widetilde{\mathbf{x}_\mathcal{T}}$ let us to recover the geometry of $\mathcal{T}$ with the discretization of $\mathcal{S}$:

$$\widetilde{X_\mathcal{T}} = \Phi_\mathcal{S} \widetilde{\mathbf{x}_\mathcal{T}}. \tag{9}$$

As stated in the previous Section, such a result is just a low pass representation of the coordinates, with potentially further noise introduced by the transfer performed with the $\boldsymbol{C}$. However, we expect that with a large enough $\boldsymbol{C}$ (that in general are around one hundred dimensions), there are enough frequencies to reliably recover the geometry structure, at least with a local coherence. Finally, we can transfer the skinning weights by nearest-neighbor pairing:

$$\mathcal{W}_{\mathcal{S}_i} = \mathcal{W}_{\mathcal{T}_j}, \text{ where } j = \arg\min_k \|\widetilde{X_{\mathcal{T}i}} - X_{\mathcal{T}k}\|_2. \tag{10}$$

$\mathcal{W}_{\mathcal{S}_i}$ denotes the set of values of the skinning weights associated to the vertex i-th of the shape $\mathcal{S}$ and similarly for $\mathcal{W}_{\mathcal{T}_j}$. In this way, we select j as the vertex of $\mathcal{T}$ the coordinates of which are closest (in the Euclidean sense) to the i-th point of the set $\widetilde{X_\mathcal{T}}$ that corresponds to the i-th vertex of the shape $\mathcal{S}$ mapped on the geometry of $\mathcal{T}$.

Then we exploit his nearest vertex association to assign the skinning weights to all the vertices of $\widetilde{X_{\mathcal{T}}}$ and therefore to the vertices of $\mathcal{S}$. This transfer preserves the original information, maintaining the absolute values of the skinning weights over the whole surface. Since the skinning weights, in general, express a proportion (i.e., how much a joint rotation affect a vertex position, where 0 applies no effect, and 1 applies the same rotation), it is essential to keep them fix (i.e., avoid variations due to scale or proportions). Notice also that potential mismatches are highly localized. In the results, we will show that if misassignments occur, they do not impact the appearance of the final retargeting.

4.2. Recovering the rest pose by optimization.

After the skinning weights transfer, we need to transfer also the skeleton. In this work, we rely on the method proposed in [MMMC21]. Now that the target shape has obtained a skinning system, it is not yet ready to inherit all the animation defined on the source shape. The LBS rotations are expressed w.r.t. a certain initial pose. To move a shape from one pose to another presents difficulties, even with the correspondence between the two joints. The component of the twist that rotates a bone long its axis cannot be directly inferred by the joints' positions. As an example, consider a leaf joint, in our case, the wrist one. The rotation of such joint affects the surface vertices, while it has no direct impact on the skeleton joint. To solve this problem, we rely on two major considerations. First, we assume that the input model, since it is not ready for animation, has a triangulation that is already convenient to represent its geometry in its present pose. Hence, any twist along the bone axis would produce undesirable artifacts which cannot be solved without retriangulation. Therefore, we would avoid such deformations. Secondly, it could be that it is not possible to align joints perfectly just by applying rotations. For example, consider a joint with two sons (e.g., the root joint and the two hips). To align the sons, we can align the two fathers and also rescaling the bones to share the same length. However, if the angle produced by the vectors that connect the father to the two sons is not equal in the two configurations, it is preserved by the rotations. This situation often occurs since different bodies may have similar skeleton structures but differences in their proportions, especially if the skeleton comes from an automatic transfer. Given these points, we propose to solve the problem as an optimization process in two different iterations. First, we solve for the best translation t that align the skeleton spatially:

$$E_{rigid}(t) = \|J_s(\Theta) + t - J_t\|. \tag{11}$$

This lets us have a global initialization before moving the joints, preventing local minima. Then, we include also the LBS parameters to the optimization variables, and we include a regularization term to minimize the performed rotation:

$$E_{pose}(t,\Theta) = \|J_s(\Theta) + t - J_t\| + \lambda \sum_i \|\theta_i\|_2. \tag{12}$$

This formulation strongly regularises the rotation looking for the easiest possible solution and avoids any rotation that does not impact the joints' position. In this way, no twists along the axis of the bones are performed. During the optimization, the weight λ is increased. In this way, the optimization starts with enough freedom to fit the rest pose, while at the end of the optimization it is constrained to remove any extra rotations.

5. Experiments and applications

Here we list our experiments in terms of qualitative and quantitative results to validate our approach. We will show qualitative examples of skinning weights transfer between different subjects, with different poses, discretizations, and identities, and we measure the transfer error w.r.t. a given ground truth. Then, we provide a qualitative example for recovering the rest pose of a given subject, and finally the result of retargeting animation on the target shape.

5.1. Skinning weights transfer

In Figure 3 we report a comparison of transfer for different methods. The source shape is the SMPL template [LMR*15], while the target is a shape from FAUST [BRLB14]. The two shapes share the connectivity; hence there exists a ground truth for the transfer. As can be seen, the point-to-point correspondence provided by functional maps is not accurate enough locally, and the result is practically unusable. Similarly, Functional maps better preserve the local pattern, but the Gibbs effect diffuses noisy values on the whole surface. Therefore, moving the joint of the right knee would impact all other vertices, which is highly undesirable. Finally, our method can preserve the local pattern avoiding frequencies leaks. Notice that our result is significantly similar to the ground truth. Further examples are shown in Figure 4. Despite the different poses and identities, the semanticity is coherent across the shapes and no serious noise is introduced.

Quantitative analysis. To test our method, we perform a quantitative analysis by transferring SMPL skinning weights to 10 remeshed FAUST. We report in Table 1 the error for each region, and we compare ourselves against other methods. It is worth noting that skinning weights values go between 0 and 1. Our method produces all errors below 0.017, showing that the transfer is reliable. We observe that the lowest error is on the part that belongs to the central part of the body, while higher errors, in general, are distributed over the protrusions. This is coherent with the functional maps theory: the low pass representation of the basis penalizes the high-frequency part of the meshes. Despite this, the nearest-neighbor assignment can still recover correct values since recovering structural information is enough. On contrary, other methods produce dramatically ruined results. The transfer performed by the p2p experience drastic errors, in some cases two orders of magnitude larger than our method. Functional maps transfer provides more stable results across the shape, but the Gibbs effect diffuses error over all the surfaces, in particular on some crucial central joints (hips and Spine3).

5.2. T-Posing

In Figure 5 we report results of our optimization process. It is worth noting that the skinning weights are transferred with our method; hence it is also a certification of the quality of our transfer. As can be seen, the optimization lifts the hand but also re-align the legs. Such minor details are crucial to obtain successful animation retargeting.

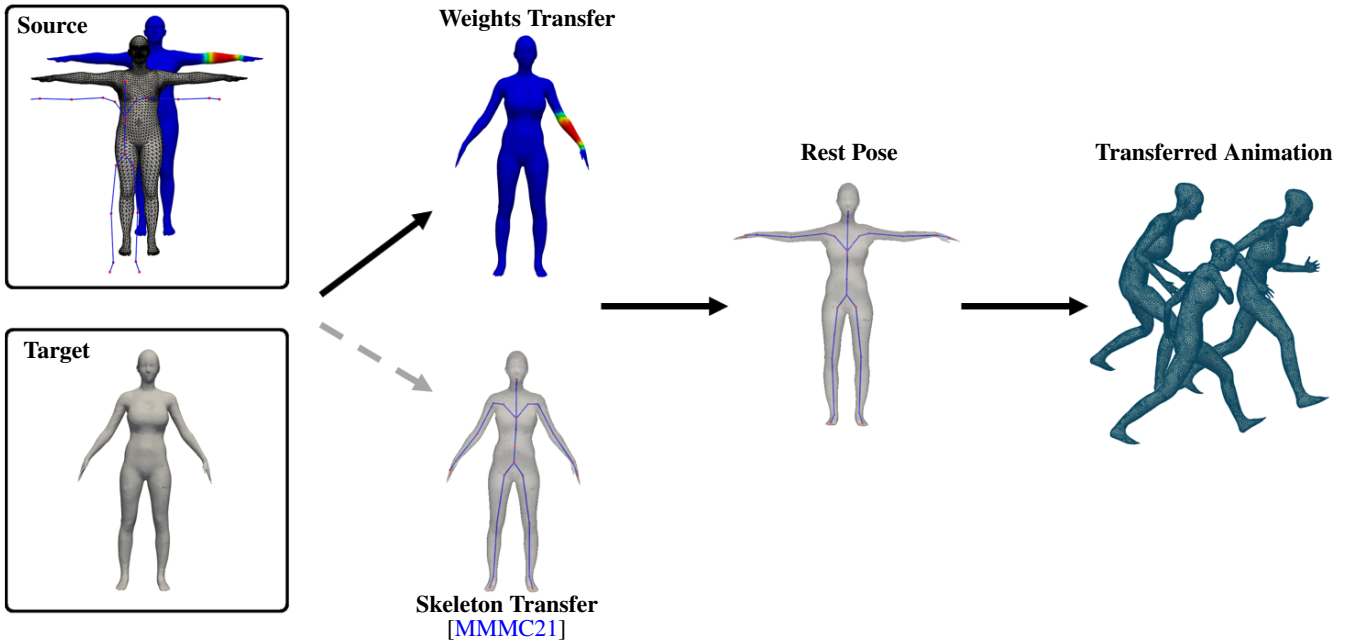

Figure 2: *Here we show the detailed steps of our pipeline. On the left, the two input shapes: a source, equipped with skinning information, and an unrigged target one. Then, we transfer the skinning weights relying on functional maps and coordinate mapping. For the skeleton, we relies on [MMMC21]. Then, through an optimization, we obtain the rest pose coherent with the source one. Now the target shape is ready to be animated using animation defined on the source. Our method does not require that the shapes share the same triangulation or are in correspondence.*

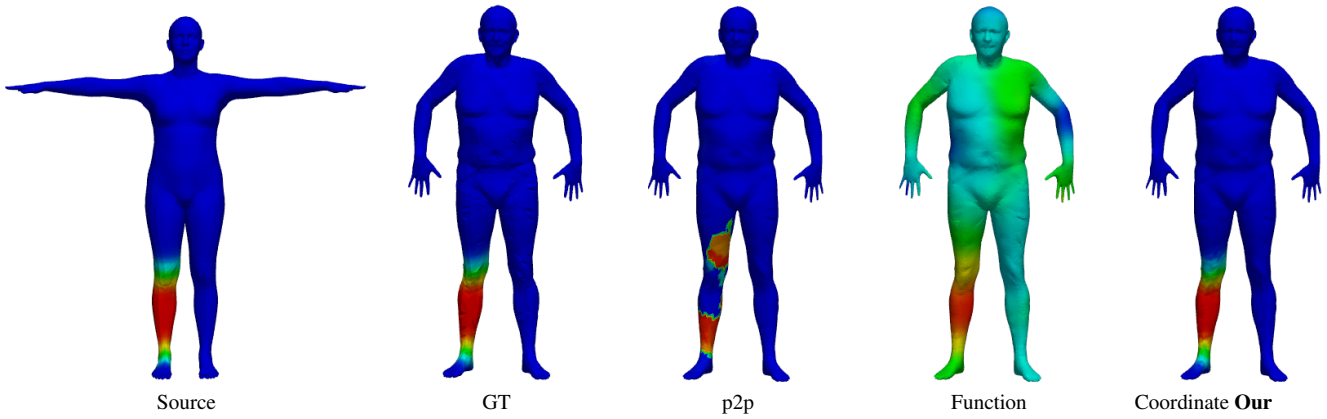

Figure 3: *A comparison between the skinning weights transfer methods. From the left: the source geometry with the skinning weights, the coordinate transfer method, the point-to-point map method, the function transfer method.*

Ablation study. As further validation of our optimization strategy, in Figure 6 we report results by changing the contribution of the different energies. Starting from the given source, we first optimize without the angle regularization (No Regularization). The provided example shows that while the skeleton is recovered in the correct position, several joints have a serious twist, producing an unreasonable result. Then, we optimize keeping the regularization energy, but without changing its value across the optimization (Fixed Weights). While the energy provides an improvement, the method still presents several artifacts in a similar zone to the previous case. Finally, we report our result (Our). It is worth noting that all results produce almost identical joints positions, and just from skeleton representation, it is impossible to infer the resulting quality of the mesh.

5.3. Animation Retargeting

Finally, to check the overall quality of our methodology, we qualitatively check the obtained transferring across different shapes. Also, in this case, the experiments are performed on the remeshed FAUST dataset. We perform the same animation with the original template and the model that inherited the skinning. We report in Figure 7 a comparison between the two. As can be seen, the two

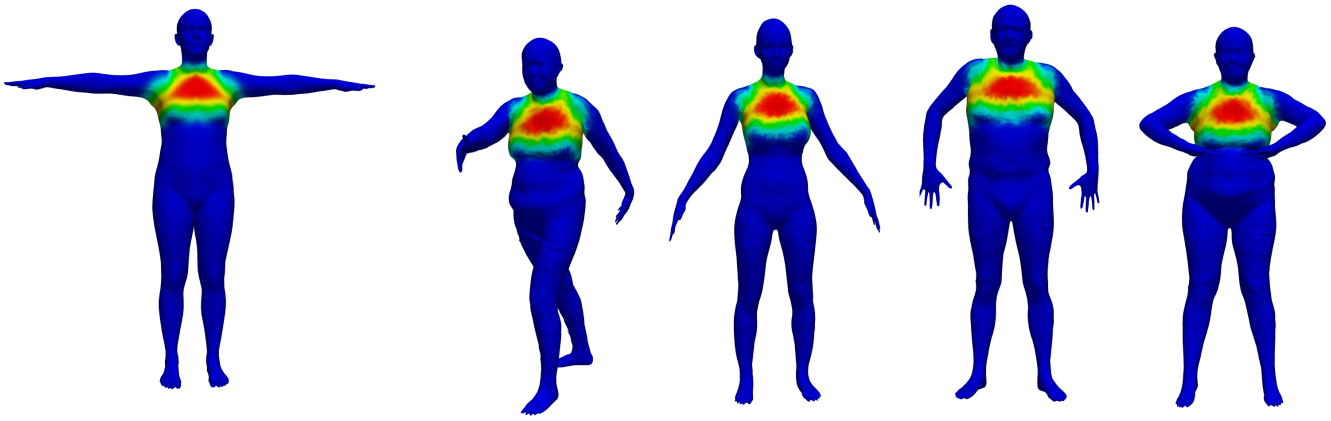

Figure 4: *Some results of the skinning weights transfer using our transfer method on different identities and initial poses. The target shape does not share the same triangulation of the starting template.*

Joint	p2p	Function	**Our**
Hips	4.06	5.60	0.16
UpLeftLeg	2.90	3.38	0.17
UpRightLeg	2.37	3.14	0.20
Spine1	1.48	2.39	0.13
LeftKnee	2.84	2.70	0.34
RightKnee	2.26	2.35	0.30
Spine2	1.69	2.75	0.22
LeftAnkle	1.00	1.19	0.27
RightAnkle	0.99	1.10	0.21
Spine3	5.35	6.46	1.44
LeftFoot	1.21	0.85	0.06
RightFoot	1.47	0.87	0.08
Neck	1.71	1.29	0.67
LeftShoulder	1.25	1.59	0.28
RightShoulder	1.12	1.44	0.25
Head	10.86	3.40	1.20
LeftArm	3.12	3.76	0.72
RightArm	3.10	4.21	0.63
LeftElbow	2.36	2.94	0.63
RightElbow	2.57	3.05	0.59
LeftForeArm	9.39	2.60	1.64
RightForeArm	12.64	2.83	1.47
LeftHand	5.23	2.94	1.64
RightHand	6.10	3.12	1.47

Table 1: *The average Mean Squared Error between the transferred skinning weights and the correspondent ground truth (values are multiplied by 10e2). We make this test over ten different shapes of FAUST dataset remeshed, and we compare the results with the other methods exposed in 5.1. The values of the skinning weights are in the range between 0 and 1*

shapes present significant morphological differences (e.g., the different head shapes, the chest and stomach shapes, the length of the legs). We can retarget the animation without producing evident artifacts, even in positions in which all the limbs are involved and in different locations w.r.t. the initial position.

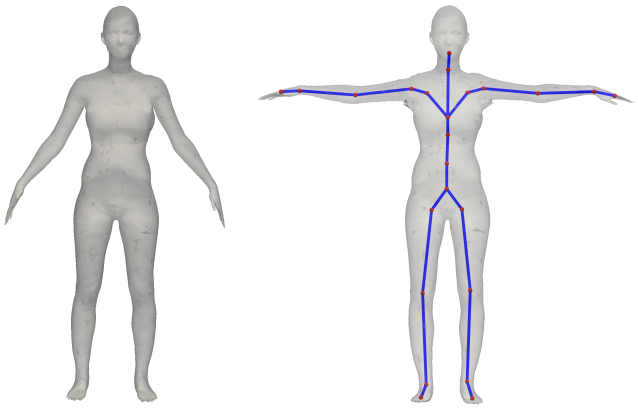

Figure 5: *A qualitative example of the recovered T-Pose. The arms are lift without introducing evident artifacts. The legs are also slightly re-aligned.*

5.4. Limitations

We believe that our study presents several limitations, which may be good starting points for future works. First of all, we have seen increasing errors in protrusions. This is a direct consequence of using the LBO eigenfunctions and could introduce errors in domains where shapes present thin and ramified protrusions. In the future, other kinds of bases may be involved to overcome this problem [MMM*20] even exploiting deep-learning techniques [MRMO20]. Also, the optimization cannot disambiguate when the shape presents self-twisting since this is an unconstrained problem in the skeleton representation. In our opinion, such a problem can be solved exploit surface features. Finally, we do not involve any further refinement of the obtained transfer. We believe that some geometrical constraints (e.g., outlier detection based on geodesic distances, deblurring techniques for transfer [EBC17]) could pro-

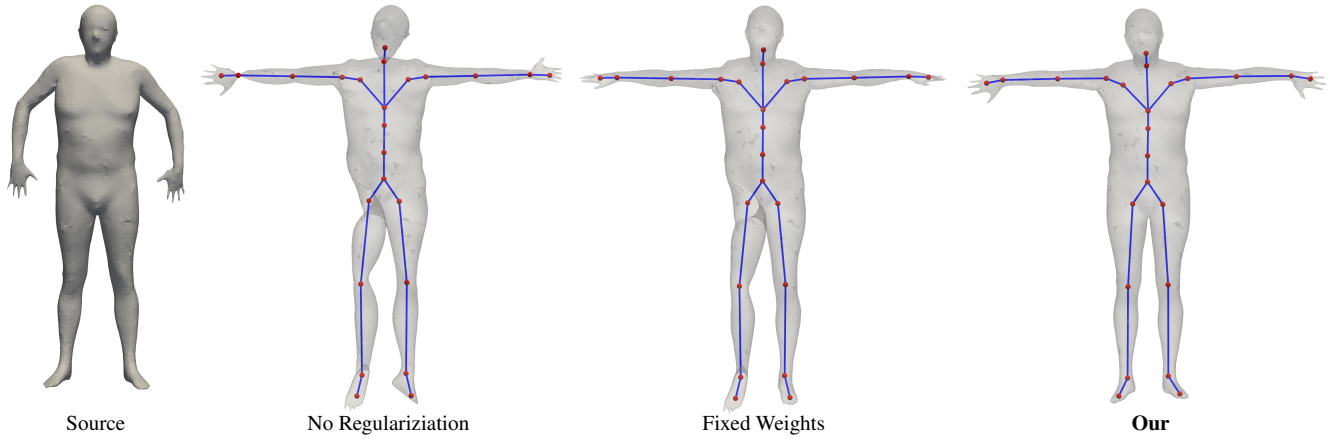

Figure 6: *An example of the importance of each step of the optimization process to obtain the rest pose. From left to right: the source geometry, the resulting shape without the regularization step, the resulting shape without adapting the weights in the regularization step, the result of the full optimization.*

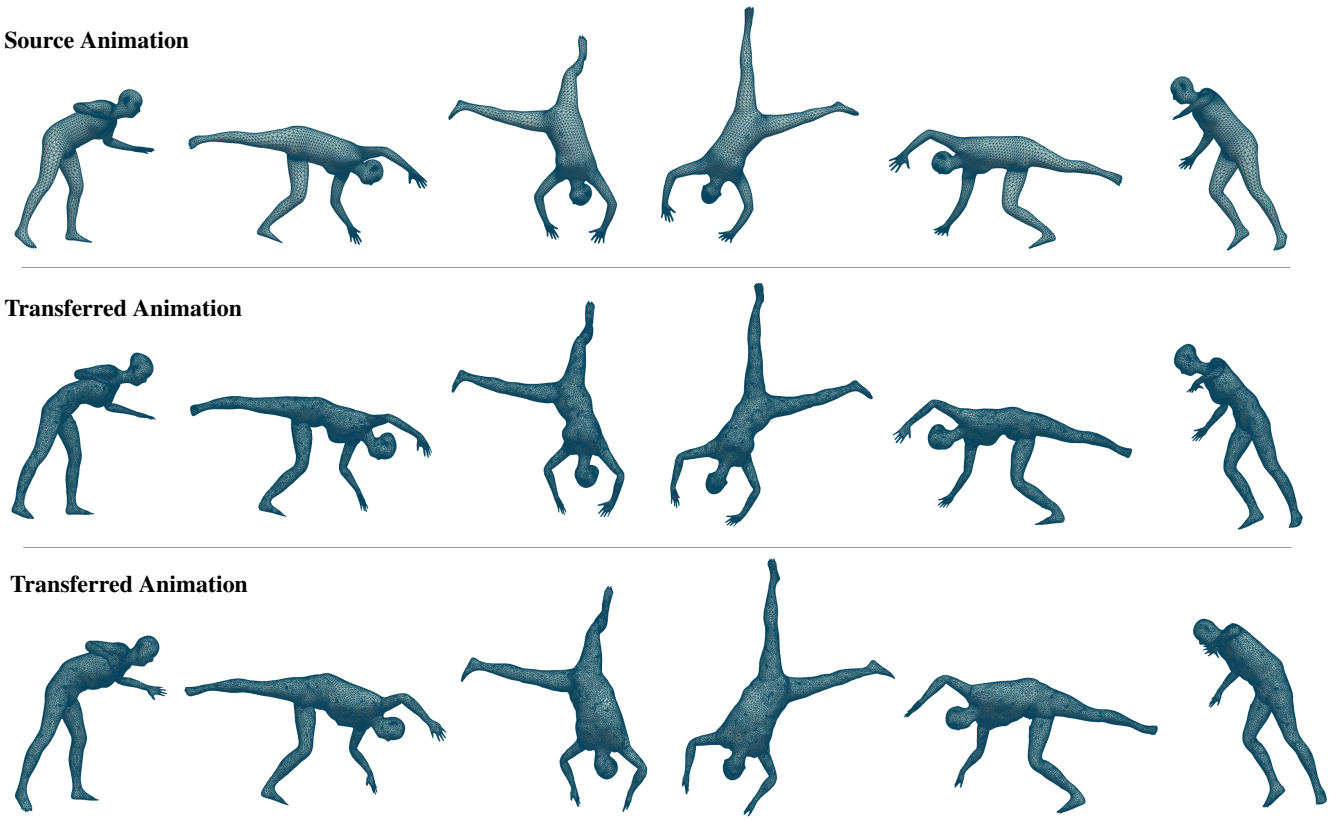

Figure 7: *An example of retargeting, where an animation defined for the SMPL template is transferred to two different remeshed FAUST shapes. In the first row, the ground-truth animation. The same motion sequence is applied to two models with different morphologies and discretizations in the second and third rows. The transferred animation sequence comes from SURREAL dataset [VRM*17].*

duce sharper results, removing artifacts in cases where exact precision is required.

6. Conclusion

This paper proposes a simple technique to transfer the animation properties between two shapes, regardless of their 3D embedding, discretization, and without requiring an explicit dense correspondence. However, there are unaddressed points. We believe our ap-

proach may find significant application in different scenarios. Still, we expect that artifacts may arise in some settings due to the low-pass representation given by the LBO basis. In the future, we believe that applying our pipeline to different animation settings could potentially shed new light on the problem and the interaction between intrinsic geometry and animation frameworks.

References

[AGR*16] AVRIL Q., GHAFOURZADEH D., RAMACHANDRAN S., FALLAHDOUST S., RIBET S., DIONNE O., DE LASA M., PAQUETTE E.: Animation setup transfer for 3d characters. In *Computer Graphics Forum* (2016), vol. 35, Wiley Online Library, pp. 115–126. 3

[BP07] BARAN I., POPOVIĆ J.: Automatic rigging and animation of 3d characters. *ACM Transactions on graphics (TOG) 26* (2007), 72–es. 2, 3

[BRLB14] BOGO F., ROMERO J., LOPER M., BLACK M. J.: FAUST: Dataset and evaluation for 3D mesh registration. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* (Piscataway, NJ, USA, June 2014), IEEE. 5

[BWBM19] BASSET J., WUHRER S., BOYER E., MULTON F.: Contact preserving shape transfer for rigging-free motion retargeting. In *Motion, Interaction and Games*. 2019, pp. 1–10. 2

[CB17] CALDERON S., BOUBEKEUR T.: Bounding proxies for shape approximation. *ACM Transactions on Graphics (TOG) 36*, 4 (2017), 1–13. 2

[DMMT*07] DELLAS F., MOCCOZET L., MAGNENAT-THALMANN N., MORTARA M., PATAN'E G., SPAGNUOLO M., FALCIDIENO B.: Knowledge-based extraction of control skeletons for animation. In *IEEE International Conference on Shape Modeling and Applications 2007 (SMI'07)* (2007), IEEE, pp. 51–60. 2

[DVLP20] DEHESA J., VIDLER A., LUTTEROTH C., PADGET J.: Touché: Data-driven interactive sword fighting in virtual reality. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (2020), pp. 1–14. 1

[EBC17] EZUZ D., BEN-CHEN M.: Deblurring and denoising of maps between shapes. *Computer Graphics Forum 36*, 5 (2017), 165–174. 3, 4, 7

[JDKL14] JACOBSON A., DENG Z., KAVAN L., LEWIS J.: Skinning: Real-time shape deformation. In *ACM SIGGRAPH 2014 Courses* (2014). 2

[KCŽO08] KAVAN L., COLLINS S., ŽÁRA J., O'SULLIVAN C.: Geometric skinning with approximate dual quaternion blending. *ACM Transactions on Graphics (TOG) 27*, 4 (2008), 1–23. 2

[LD14] LE B. H., DENG Z.: Robust and accurate skeletal rigging from mesh sequences. *ACM Trans. Graph. 33*, 4 (July 2014). 2

[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* (2015), 1–16. 4, 5

[LOP*18] LIVIGNI A., O'HARA L., POLAK M. E., ANGUS T., WRIGHT D. W., SMITH L. B., FREEMAN T. C.: A graphical and computational modeling platform for biological pathways. *Nature protocols 13*, 4 (2018), 705–722. 2

[MF19] MOUTAFIDOU A., FUDOS I.: A mesh correspondence approach for efficient animation transfer. In *CGVC* (2019), pp. 129–133. 3

[MMM*20] MELZI S., MARIN R., MUSONI P., BARDON F., TARINI M., CASTELLANI U.: Intrinsic/extrinsic embedding for functional remeshing of 3d shapes. *Computers & Graphics 88* (2020), 1 – 12. 2, 7

[MMMC21] MUSONI P., MARIN R., MELZI S., CASTELLANI U.: A functional skeleton transfer. *Proceedings of the ACM on Computer Graphics and Interactive Techniques 4*, 3 (2021), 1–15. 5, 6

[MMRC20] MARIN R., MELZI S., RODOLÀ E., CASTELLANI U.: Farm: Functional automatic registration method for 3d human bodies. In *Computer Graphics Forum* (2020), vol. 39, Wiley Online Library, pp. 160–173. 2

[MRMO20] MARIN R., RAKOTOSAONA M.-J., MELZI S., OVSJANIKOV M.: Correspondence learning via linearly-invariant embedding. 7

[MRR*19] MELZI S., REN J., RODOLÀ E., SHARMA A., WONKA P., OVSJANIKOV M.: Zoomout: spectral upsampling for efficient shape correspondence. *ACM Transactions on Graphics 38*, 6 (2019), 1–14. 3, 4

[MTLT88] MAGNENAT-THALMANN N., LAPERRIRE R., THALMANN D.: Joint-dependent local deformations for hand animation and object grasping. In *In Proceedings on Graphics interface'88* (1988), Citeseer. 2

[MZR*20] MANSOR N. R., ZAKARIA R., RASHID R. A., ARIFIN R. M., ABD RAHIM B. H., ZAKARIA R., RAZAK M. T. A.: A review survey on the use computer animation in education. In *IOP Conference Series: Materials Science and Engineering* (2020), vol. 917, IOP Publishing, p. 012021. 1

[NMR*18] NOGNENG D., MELZI S., RODOLÀ E., CASTELLANI U., BRONSTEIN M., OVSJANIKOV M.: Improved functional mappings via product preservation. *Computer Graphics Forum 37*, 2 (2018), 179–190. 3

[NO17] NOGNENG D., OVSJANIKOV M.: Informative Descriptor Preservation via Commutativity for Shape Matching. *Computer Graphics Forum* (2017). 3

[OBCS*12] OVSJANIKOV M., BEN-CHEN M., SOLOMON J., BUTSCHER A., GUIBAS L.: Functional maps: a flexible representation of maps between shapes. *ACM Transactions on Graphics (TOG) 31*, 4 (2012), 30:1–30:11. 2, 3

[PRM*21] PAI G., REN J., MELZI S., WONKA P., OVSJANIKOV M.: Fast sinkhorn filters: Using matrix scaling for non-rigid shape correspondence with functional maps. In *CVPR* (2021). 3, 4

[RMC15] RODOLÀ E., MOELLER M., CREMERS D.: Point-wise map recovery and refinement from functional correspondence. In *Proc. Vision, Modeling and Visualization (VMV)* (2015). 3

[RMOW20] REN J., MELZI S., OVSJANIKOV M., WONKA P.: Maptree: Recovering multiple solutions in the space of maps. *ACM Trans. Graph. 39*, 6 (Nov. 2020). 3

[RMWO21] REN J., MELZI S., WONKA P., OVSJANIKOV M.: Discrete optimization for shape matching. *Computer Graphics Forum 40*, 5 (2021), 81–96. 3

[RPWO18] REN J., POULENARD A., WONKA P., OVSJANIKOV M.: Continuous and orientation-preserving correspondences via functional maps. *ACM Transactions on Graphics (TOG) 37*, 6 (2018). 3

[SS19] SEYLAN Ç., SAHILLIOĞLU Y.: 3d skeleton transfer for meshes and clouds. *Graphical Models 105* (2019), 101041. 2

[VGK17] VONACH E., GATTERER C., KAUFMANN H.: Vrrobot: Robot actuated props in an infinite virtual environment. In *2017 IEEE Virtual Reality (VR)* (2017), IEEE, pp. 74–83. 1

[VRM*17] VAROL G., ROMERO J., MARTIN X., MAHMOOD N., BLACK M. J., LAPTEV I., SCHMID C.: Learning from synthetic humans. In *CVPR* (2017). 8

[XLG09] XIAN C., LIN H., GAO S.: Automatic generation of coarse bounding cages from dense meshes. In *2009 IEEE International Conference on Shape Modeling and Applications* (2009), IEEE, pp. 21–27. 2

[XLG12] XIAN C., LIN H., GAO S.: Automatic cage generation by improved obbs for mesh deformation. *The Visual Computer 28*, 1 (2012), 21–33. 2

[XLX15] XIAN C., LI G., XIONG Y.: Efficient and effective cage generation by region decomposition. *Computer Animation and Virtual Worlds 26*, 2 (2015), 173–184. 2

[YYG18] YANG B., YAO J., GUO X.: Dmat: Deformable medial axis transform for animated mesh approximation. In *Computer Graphics Forum* (2018), vol. 37, Wiley Online Library, pp. 301–311. 2