

A functional skeleton transfer

PIETRO MUSONI, University of Verona, Italy

RICCARDO MARIN, Sapienza University of Rome, Italy

SIMONE MELZI, Sapienza University of Rome, Italy

UMBERTO CASTELLANI, University of Verona, Italy

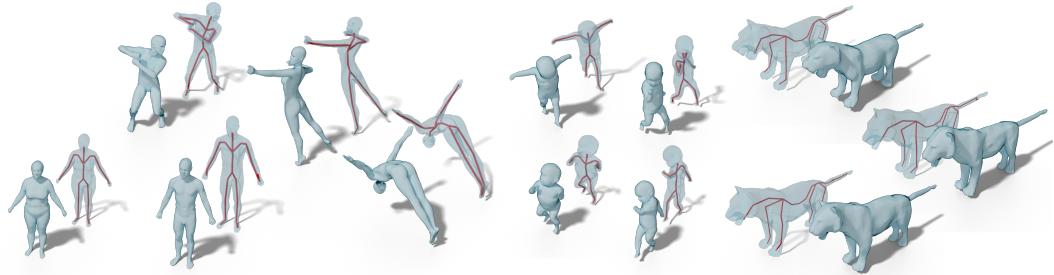


Fig. 1. Some results of our method. On the left, five adult humans that inherited the SMPL [Loper et al. 2015] skeleton; these surfaces come from different datasets and have different discretizations. In the middle, four children shapes on which we transfer the skeleton from SMIL [Hesse et al. 2018]. On the right, three lions that received the skeleton from SMAL [Zuffi et al. 2017].

The animation community has spent significant effort trying to ease rigging procedures. This is necessitated because the increasing availability of 3D data makes manual rigging infeasible. However, object animations involve understanding elaborate geometry and dynamics, and such knowledge is hard to infuse even with modern data-driven techniques. Automatic rigging methods do not provide adequate control and cannot generalize in the presence of unseen artifacts. As an alternative, one can design a system for one shape and then transfer it to other objects. In previous work, this has been implemented by solving the dense point-to-point correspondence problem. Such an approach requires a significant amount of supervision, often placing hundreds of landmarks by hand. This paper proposes a functional approach for *skeleton transfer* that uses limited information and does not require a complete match between the geometries. To do so, we suggest a novel representation for the skeleton properties, namely the *functional regressor*, which is compact and invariant to different discretizations and poses. We consider our functional regressor a new operator to adopt in intrinsic geometry pipelines for encoding the pose information, paving the way for several new applications. We numerically stress our method on a large set of different shapes and object classes, providing qualitative and numerical evaluations of precision and computational efficiency. Finally, we show a preliminary transfer of the complete rigging scheme, introducing a promising direction for future explorations.

CCS Concepts: • Computing methodologies → Animation; Shape analysis.

Authors' addresses: Pietro Musoni, University of Verona, Verona, Italy, pietro.musoni@univr.it; Riccardo Marin, Sapienza University of Rome, Rome, Italy, marin@di.uniroma1.it; Simone Melzi, Sapienza University of Rome, Italy, melzi@di.uniroma1.it; Umberto Castellani, University of Verona, Verona, Italy, umberto.castellani@univr.it.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2021 Copyright held by the owner/author(s). Publication rights licensed to ACM.

2577-6193/2021/9-ART \$15.00

Additional Key Words and Phrases: animation, skeleton transfer, LBS, functional methods

ACM Reference Format:

Pietro Musoni, Riccardo Marin, Simone Melzi, and Umberto Castellani. 2021. A functional skeleton transfer. *Proc. ACM Comput. Graph. Interact. Tech.* 4, 3 (September 2021), 15 pages.

1 INTRODUCTION

Animating a 3D object is a time-consuming and challenging problem; it requires matching the expectations suggested by our experience and balancing the trade-off between usability and expressiveness. Also, a coherent rigging system across similar objects is desirable to allow reuse of animations with little effort.

Several methods have been proposed to solve the automatic rigging of meshes. They rely on algorithms that process the geometric structure [Jacobson et al. 2012] or, more recently, by learning using data-driven approaches [Rajendran and Lee 2020]. However, they hardly produce coherent results among arbitrary objects and do not provide the user with additional information about the semantics and the structural role of each component of the shape. To keep a degree of control, another approach is defining a rigging system and transferring it to other models, but this is commonly achieved by establishing a dense point-to-point correspondence, which requires user annotations. Also, determining point-to-point correspondence is one of the fundamental open problems of geometry processing, particularly when facing non-isometric objects.

In this work, we claim that the skeleton can be accurately transferred by a functional approach without relying on a dense point-to-point correspondence. Our pipeline asks the user to define a rigged system for one shape. Then, we optimize for a regressor that is parsed to a functional representation, and finally, it is transferred to other shapes. This representation offers a new perspective in skeleton encoding; it is compact and entirely intrinsic (i.e., it does not change with varying poses of the subject). Given a new shape to animate, we ask for a few landmarks on the target object (seven on humans and which, in practice, can be detected almost automatically) and we rely on the Functional Maps framework [Nogneng and Ovsjanikov 2017] to transfer our rigging. This framework is state-of-the-art for the functional transferring task, both in terms of performance and efficiency. Our approach produces results coherent across different objects of the same class, as shown in Figure 1. Finally, we cannot find implementations for previous methods; we consider this one of the main reasons for proposing novel approaches. The code and data used in this paper will be made entirely available for research purposes.

To summarize, our contribution is threefold:

- (1) We propose the first functional pipeline for skeleton transfer. It produces accurate results efficiently and with limited user input. It is general across different settings and object classes.
- (2) We introduce a new spectral representation for the shape skeleton information; that is compact, entirely intrinsic, and invariant to different discretizations and poses of the subject. This representation provides a new perspective to analyze object mechanics.
- (3) We will release all the code and data used in our experiments.

2 STATE OF THE ART

2.1 Classical methods

Several researchers have tried to define the skeleton for a given shape by using automatic systems. One of the fundamental works in this field is Pinocchio [Baran and Popović 2007] by Baran and Popović; the proposed pipeline produces an animation-ready character starting from a generic object. This process is divided into two steps: skeleton embedding and skin attachment.

Some works [Gleicher 1998], [Lee and Shin 1999], [Choi and Ko 2000] solve kinematic constraints on joint positions using displacement maps. In [Allen et al. 2003], the authors provide joint retargeting through the use of three markers. The distance between the joints and the markers should be fixed. However, this method works only on near-similar morphological bodies. Methods like [Capell et al. 2005] and [Orvalho et al. 2008] introduced the transferring of the rig. The first is not compatible with common skinning methods, and the second can only be applied to a limited set of characters. In [Seo et al. 2010], the authors provide a method based on data mapping applicable to surfaces and volumes, but a careful manual selection of points is necessary to initialize the correspondence.

2.2 Automatic Rigging and rigging transfer

The main disadvantage of automatic rigging methods is that the animation sequence should be designed on a specific skeleton. Instead, the rigging transfer methods tackled this problem by transferring a native skeleton with an attached animation, combining it to the new shape. Several works focus on the transfer of the medial-axis [Yang et al. 2018] [Seylan and Sahillioğlu 2019], which is the set of points having more than one closest point to the boundary of a shape. The medial axis of a 3D object can be expressed both as a curve and as a 2D surface. However, medial axis methods are not compatible with standard render engines, and they are computationally inefficient.

Another approach explored in [Le and Deng 2012] and [Le and Deng 2014] employs example-based methods to automatically generate linear blend skinning models. The method takes as input a sequence of poses of a model and it outputs the skeleton, the skinning weights and the bone transformations. The output is compatible with common game engines and 3D modelling software, however, such a sequence of shapes is not always easy to have since requires data acquisition from a real-world scenario and the method largely depends on the quality of input example meshes.

2.3 Motion transfer

Other approaches [Liu et al. 2018], [Basset et al. 2019] propose to transfer each pose of a 3D shape to another using a non-rigid registration. In these methods, a further adjustment is required to have a smooth animation from the sequence of poses obtained. A more promising procedure involves deforming a shape to adapt to a rigged one, applying the given skeleton to the deformed shape. In [Marin et al. 2020], [Melzi et al. 2020] and [Ni et al. 2020], the authors made a non-rigid registration from a source shape to a target one, and then transfer the skinning information. A similar work [Jin et al. 2018] introduces the idea of aura mesh, a volume that surrounds the character and permits avoidance of self-collisions. The rigless methods [Basset et al. 2019] do not need the post-processing angle correction, but they require additional steps to obtain an animation-ready shape as output. In [Avril et al. 2016] the authors compute a pointwise matching between the two shapes to transfer all the properties of an animation setup. This method requires the manual input of more than 300 key points to initialize the dense correspondence. Then, a regression matrix is estimated to compute the joint positions of the new shape. In addition to the high amount of user input, the method requests the two shapes to be in the same pose. A joint-regressor was also introduced for the SMPL morphable model [Loper et al. 2015], but it requires learning from a large dataset of 3D shapes. Furthermore, the joint-regressor of SMPL is defined pointwise, hence its transfer would require a dense point-to-point correspondence.

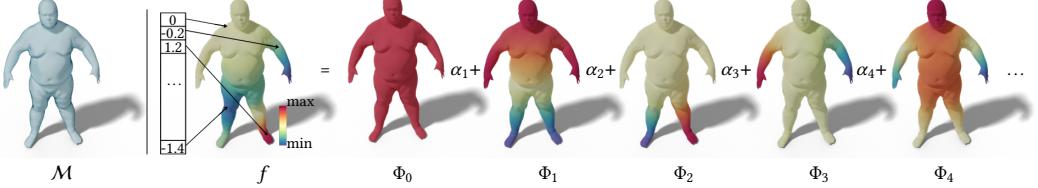


Fig. 2. Here we visualize some of the functional quantities we referred into the text. Given a geometry \mathcal{M} , a function f is a vector that associates a scalar value to each point of the surface. Such function can be decomposed for some basis Φ (e.g. the LBO eigenfunctions) recovering a list of coefficients α .

3 BACKGROUND

3.1 LBO basis on shape

We describe 2-dimensional surfaces embedded in \mathbb{R}^3 as compact and connected 2-dimensional Riemannian manifolds. Given a surface \mathcal{M} , we consider $\mathcal{F}(\mathcal{M}) := \{f : \mathcal{M} \rightarrow \mathbb{R}\}$ as the space of real-valued functions defined over it. In other words, a function $f \in \mathcal{F}(\mathcal{M})$ associates at every point $x \in \mathcal{M}$ a real value $f(x) \in \mathbb{R}$. In the discrete setting, we encode the shape \mathcal{M} in a triangular mesh, defined by the 3D coordinates of its $n_{\mathcal{M}}$ vertices $X_{\mathcal{M}} \in \mathbb{R}^{n_{\mathcal{M}} \times 3}$ and the list of the edges $E_{\mathcal{M}}$ that connect these vertices generating triangular faces. We associate to each mesh the cotangent Laplace-Beltrami operator (LBO), that is discretized by a $n_{\mathcal{M}} \times n_{\mathcal{M}}$ matrix [Meyer et al. 2003]. We collect the first k eigenfunctions of the LBO of \mathcal{M} (the ones associated to the k smallest eigenvalues) as columns in a matrix $\Phi_{k_{\mathcal{M}}} = [\phi_1, \phi_2, \dots, \phi_{k_{\mathcal{M}}}]$. In geometry processing, these eigenfunctions are widely used to approximate functions or signals defined on 3D shapes extending Fourier analysis on non-Euclidean domains [Lévy 2006; Taubin 1995; Vallet and Lévy 2008]. In particular, functions can be represented by a linear combination of the basis $\Phi_{k_{\mathcal{M}}}$ as $\tilde{f} = \Phi_{k_{\mathcal{M}}} \widehat{f}$, where \widehat{f} are the basis coefficients. Since just a subset of the eigenfunctions is used, this is just an approximation of f . The coefficients \widehat{f} can be obtained by the projection $\widehat{f} = \Phi_{k_{\mathcal{M}}}^{\dagger} f$, where $\Phi_{k_{\mathcal{M}}}^{\dagger}$ is the Moore-Penrose pseudo-inverse (i.e. $\Phi_{k_{\mathcal{M}}}^{\dagger} \Phi_{k_{\mathcal{M}}} = Id_{k_{\mathcal{M}} \times k_{\mathcal{M}}}$). The matrix $\Phi_{k_{\mathcal{M}}}^{\dagger}$ could be seen as the operator that projects the pointwise values of the function into its Fourier representation. We will exploit this projection through the paper, and we refer to this space of the Fourier coefficients as the functional representation of quantities defined pointwise on the mesh. A visualization of these quantities is shown in Figure 2.

3.2 Functional Maps

A correspondence T between two surfaces \mathcal{N} and \mathcal{M} is a mapping that associates to each point $p \in \mathcal{N}$ the corresponding point $T(p) \in \mathcal{M}$. In the discrete case, we can encode this mapping in a matrix $\Pi_{\mathcal{NM}} \in \mathbb{R}^{n_{\mathcal{M}} \times n_{\mathcal{N}}}$ such that $\Pi(i, j) = 1$ if the j -th vertex of the shape \mathcal{N} corresponds to the i -th vertex of the shape \mathcal{M} and $\Pi(i, j) = 0$ otherwise. From a functional perspective, any function $f \in \mathcal{F}(\mathcal{M})$ could be transferred to a function $g = f \circ T \in \mathcal{F}(\mathcal{N})$. Hence, given the truncated bases $\Phi_{k_{\mathcal{M}}}$ and $\Psi_{k_{\mathcal{N}}}$ obtained from the eigendecomposition of the LBO respectively of \mathcal{M} and \mathcal{N} , the Functional Maps framework [Ovsjanikov et al. 2012] exploits these truncated bases to encode this mapping in a compact matrix $C \in \mathbb{R}^{k_{\mathcal{N}} \times k_{\mathcal{M}}}$ referred as a *functional map*. The matrix C can be directly derived from Π with the following equation $C = \Psi_{k_{\mathcal{N}}}^{\dagger} \Pi_{\mathcal{NM}} \Phi_{k_{\mathcal{M}}}$. Once we have a functional map C , we can recover the correspondence T by transferring an indicator function for each vertex, but [Ovsjanikov et al. 2012] show that this is equivalent to the nearest-neighbour assignment problem between $\Phi_{k_{\mathcal{M}}} C^{\top}$ and $\Psi_{k_{\mathcal{N}}}$. Different strategies and constraints have been proposed to solve for the matrix C [Nogneng et al. 2018; Nogneng and Ovsjanikov 2017; Ovsjanikov et al.

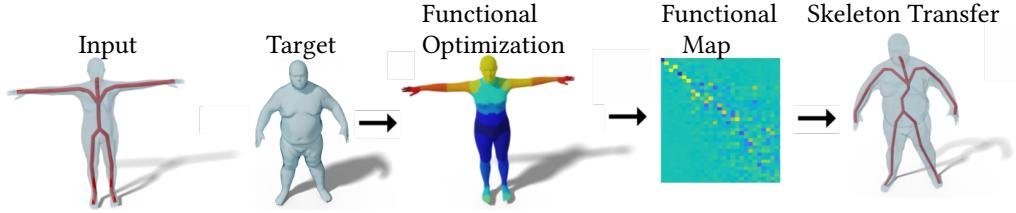


Fig. 3. This figure depicts our pipeline overview. Our method inputs are an initial rigged model and a target mesh. Then, we perform optimization to obtain the functional representation of the mesh skinning, and by computing a functional map between the two, we can transfer the skeleton to the target. The transfer lets us move the target following the semantic of the source.

2012; Ren et al. 2019, 2018]. Several attempts have also been implemented to extract and refine the pointwise correspondence from the functional representation [Ezuz and Ben-Chen 2017; Melzi et al. 2019b; Pai et al. 2021; Ren et al. 2020; Rodolà et al. 2015]. They produce meaningful results, but the conversion to pointwise maps induces distortions even for accurate functional maps. Here, we adopt the method proposed in [Nogneng and Ovsjanikov 2017] following the procedure described in [Melzi et al. 2019b].

4 METHOD

Here we describe our functional-based representation and its relation to the spatial one. Then, we present our algorithm for skeleton transfer, summarized by Figure 3.

4.1 Functional Regressor

A common choice is defining the skeleton $J_M \in \mathbb{R}^{Q \times 3}$ as a function of the given surface M , namely the *joint regressor*. Following [Loper et al. 2015] it can be expressed as a linear operator $R \in \mathbb{R}^{Q \times n_M}$, that acts on the coordinates of the vertices X_M in the discrete setting; hence:

$$J_M = RX_M. \quad (1)$$

We refer to R as the standard or spatial regressor.

Starting from this, we propose a novel functional formulation. We can consider the coordinates of the vertices of the meshes X_M as a set of three functions (one for each column) and project them in the space of the Fourier coefficients:

$$\widehat{X}_M = \Phi_{k_M}^\dagger X_M \quad (2)$$

where the coefficients $\widehat{X}_M \in \mathbb{R}^{k_M \times 3}$.

Motivation. Since our functional representation is defined on the low-pass filtering of the 3D coordinates, it captures the global structure of the shape, with resilience to high-frequency details. Also, it is intrinsic; it is agnostic to the subject pose and placement in the 3D space. Moreover, it is compact: a spatial joint regressor is encoded by a matrix $n_M \times Q$; our functional regressor has dimension $k_M \times Q$. We generally select $k_M \in [30, 200]$; smaller than n_M by at least two orders of magnitude.

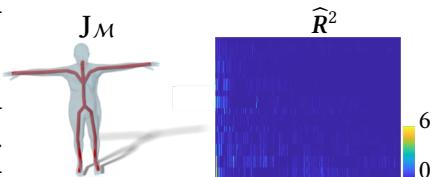


Fig. 4. On the left, the defined skeleton. On the right, the functional regressor.

In this functional representation, we want to find a linear operator \widehat{R} such that:

$$J_M = \widehat{R}\widehat{X}_M \quad (3)$$

where we refer to $\widehat{R} \in \mathbb{R}^{Q \times k_M}$, as the *functional* or *spectral* regressor. A visualization of the functional regressor is reported in Figure 4; each column represents a dimension of the LBO basis (one of the eigenfunctions), and each row a different joint. Further comments and analysis are shown in Section 6.

Relation between spectral and standard regressor. Let us consider a single shape M with its 3D coordinates X_M , its LBO eigenfunctions Φ_{k_M} , and the Fourier coefficients of its coordinates \widehat{X}_M . From Equation 3 and from $J_M = RX_M$, we can write:

$$J_M = RX_M \approx \widehat{R}\widehat{X}_M = \widehat{R}\Phi_{k_M}^\dagger X_M, \quad (4)$$

from which we obtain:

$$R \approx \widehat{R}\Phi_{k_M}^\dagger. \quad (5)$$

From the definition of the Moore–Penrose pseudo-inverse we have:

$$R\Phi_{k_M} \approx \underbrace{\widehat{R}\Phi_{k_M}^\dagger\Phi_{k_M}}_{Id_{k_M \times k_M}} = \widehat{R}, \quad (6)$$

and finally we can derive these two relations:

$$R \approx \widehat{R}\Phi_{k_M}^\dagger \quad (7)$$

$$\widehat{R} \approx R\Phi_{k_M}. \quad (8)$$

4.2 Skeleton transfer algorithm

Algorithm 1 Skeleton Transfer

- 1: Input: A rigged shape M , a target shape N , and a set of seven landmarks
 - 2: Step 1: Estimate the spatial regressor R by optimizing the energy E
 - 3: Step 2: Compute the FMAP C using the given landmarks
 - 4: Step 3: Convert the spatial regressor R in the functional representation \widehat{R}
 - 5: Step 4: Compute J_N using the transferred spectral regressor
 - 6: Output: A skeleton J_N for the target shape N
-

We consider a model M that is animation-ready, composed by a triangular mesh represented by n_M vertices $X_M \in \mathbb{R}^{n_M \times 3}$ and equipped with a skeleton $J_M \in \mathbb{R}^{Q \times 3}$ composed by Q joints. The SMPL model proposed in [Loper et al. 2015] is one possible model that we consider in our experiments. Our objective is to transfer the rigging of M to a new shape N represented by a triangular mesh with n_N vertices $X_N \in \mathbb{R}^{n_N \times 3}$. As additional information, we require some landmarks that for the human case are seven. Five of them are located on protrusions and can be automatically computed exploiting the method proposed in [Marin et al. 2020]. The remaining two are on the front and the back of the torso. In our experiments, we select them manually since this requires just a few seconds.

Step 1: Spatial Regressor Optimization We want to compute the mesh skeleton through a linear operator \widehat{R} defined over the Fourier coefficients of the 3D coordinates of the vertices of the input

mesh. To do so, we first obtain a standard regressor R as the solution of an optimization defined as the linear combination of 4 energies.

Reconstruction Term. The first energy asks that the regressor R returns the correct position of the joints:

$$E_{rec} = \|RX_M - J_M\|^2. \quad (9)$$

Locality Term. To have a regressor invariant to pose, each joint should depend only on the vertices linked by a rigid relation to it, i.e. by a near set of them. To do so, we define a mask matrix $F \in \mathbb{R}^{Q \times n_M}$, that has $F(q, i) = 1$ if the position of the vertex v_i influences the position of the joint J_q and 0 otherwise. To select which vertices should affect the joint position, we took the ones in the intersection between the non-zero skinning weights belonging to each joint and its hierarchical parent. The resulting set is near to the joint, and so we selected them to populate F . We decided to choose the vertices between the joint and its parent since they are near the joint and they have low weights so the deformation of these vertices is low changing between a pose and another. Then, the energy is:

$$E_{loc} = \|R \odot F - R\|^2 \quad (10)$$

where \odot is the element-wise product.

Sparsity Term. While E_{loc} guarantees that joints are affected by a limited set of vertices, nothing has been imposed on the entries of R that correspond to zero values in the mask F . For this reason, we promote zero-values in all these entries that should not contribute to the computation of J_M :

$$E_{spa} = \||F - \mathbf{1}| \odot R\|^2, \quad (11)$$

where $\mathbf{1}$ is a matrix with the same size of F , the entries of which are all equal to 1.

Convexity Term. We observed in our experiment that requiring the weights of each joint to sum to 1 helps to obtain joints inside the mesh boundaries, even if we do not explicitly require that all the weights are non negative. We implement this through the following energy:

$$E_{con} = \|Re_{n_M} - e_Q\|^2 \quad (12)$$

where e_{n_M} and e_Q are vectors of ones with length n_M and the number of joints respectively.

Optimization formulation. The optimization to recover the desired regressor is obtained by a linear combination of these four energies:

$$E = \omega_{rec} * E_{rec} + \omega_{loc} * E_{loc} + \omega_{spa} * E_{spa} + \omega_{con} * E_{con} \quad (13)$$

where $\omega_{rec}, \omega_{loc}, \omega_{spa}, \omega_{con}$ are positive real numbers that act as weights. We empirically set them to $\omega_{rec} = 1$, $\omega_{loc} = 10e3$, $\omega_{con} = 1$ and $\omega_{spa} = 100$. Putting high weights on E_{loc} and E_{spa} forces the regressor to better compute the joints location as a function of the vertices selected by F . These dependencies between each joint and the vertices are independent across the different joints. Apart from setting a rough balance between energies, no exhaustive research has been done.

Implementation. We use the Tensorflow auto-differentiation library to solve this problem, relying on the Adam optimizer with a learning rate of 0.1 and using 1000 iterations.

Step 2: Functional Maps computation Given a new shape N for which the rigging is not available, our goal is to transfer the regressor from M to N . Hence, we estimate a functional map C between $\mathcal{F}(M)$ and $\mathcal{F}(N)$ following the procedure proposed in [Nogneng and Ovsjanikov 2017]. We set both the sizes of the Fourier bases k_M and k_N computing an initial functional map of dimension 20×20 . We refine this map with ZoomOut the refinement technique proposed in [Melzi

et al. 2019b], bringing to a 120×120 map. Thanks to our functional definition of the regressor, we are able to perform this transfer without requiring a pointwise correspondence between \mathcal{M} and \mathcal{N} .

Step 3: Regressor conversion Once we optimize the regressor R we can easily compute its functional representation \widehat{R} exploiting Equation 8: $\widehat{R} \approx R\Phi_{k_M}$.

Step 4: Skeleton transfer Finally, the skeleton of the shape \mathcal{N} is obtained by:

$$J_N = \widehat{R}C\widehat{X}_N. \quad (14)$$

Our complete implementation is available online for full reproducibility.¹

5 RESULTS

In this section, we test the robustness of our method on several shapes collected from different datasets, providing both qualitative and quantitative evaluations and comparisons to the most related approaches. Moreover, we analyze the computational efficiency of our approach.

Data. We evaluate our method on a wide selection of shapes. From the *SHREC19* dataset [Melzi et al. 2019a], which consists of 44 shapes of different subjects in many poses and with varying discretizations, and the *Dyna* set, composed of 155 shapes of 8 different subjects from Dynamic Faust [Bogo et al. 2017]. They present the typical artifacts of real scans and are significantly different in their body characteristics. We also stress our transferring by assuming that *Dyna* is composed just of point clouds (i.e., without a triangulation) and by estimating the appropriate LBO operator from [Sharp and Crane 2020]. Notice that this is a significantly challenging setting (marked with a final *P.C.* in the table), showing our flexibility. Finally, we also consider two different populations of characters to show the generalization of our method: animals (i.e., *SMAL*[Zuffi et al. 2017]; examples are reported in Figures 1 and 5) and children (i.e., *SMIL*[Hesse et al. 2018]; examples are reported in Figures 1). We generate ten different shapes from each of the respective morphable models introducing an independent remeshing procedure to break the regularity of the connectivity.

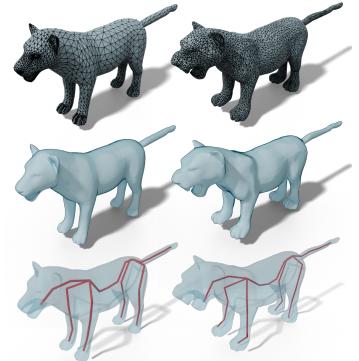


Fig. 5. A transfer between two animals with different geometry and discretization. Notice that finding a point-to-point correspondence, in this case, could be difficult even if the two shapes are in the same pose.

Transferring. We quantitatively evaluated our method on the skeleton transfer task, and we report the results in Table 1. For each dataset, the transfer is performed i) by using the functional approach described in Section 4.2 (*functional* in the table), and ii) by converting the functional mapping to a point-to-point correspondence and using regressor in its spatial form (*pointwise* in the table). In the first two datasets of adult humans (SHREC19 and *Dyna*) the shapes shared the same skeleton of the SMPL model [Loper et al. 2015], while for children and animals the skeleton is provided by the related morphable models (*SMIL* and *SMAL*). For each setting, we report the Mean Squared Error with respect to the skeletons obtained using ground truth correspondences with the related morphable model.

¹Demo Code: <https://github.com/PietroMsn/Functional-skeleton-transfer>

Dataset	mean	min	max
SHREC19 functional	0.0210	0.0072	0.0512
SHREC19 pointwise	0.0215	0.0072	0.0525
Dyna functional P.C.	0.0237	0.0078	0.1934
Dyna pointwise P.C.	0.0255	0.0125	0.2013
SMIL functional	0.0526	0.0376	0.0658
SMIL pointwise	0.0527	0.0376	0.0660
SMAL functional	0.0635	0.0323	0.1190
SMAL pointwise	0.0692	0.0350	0.1323

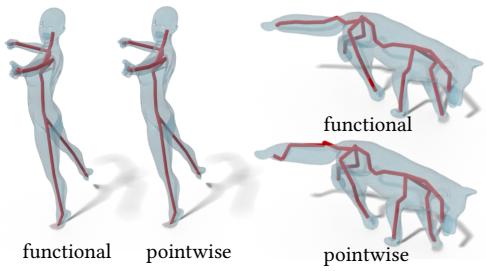


Table 1. Mean, minimum and maximum error averaged over the datasets, comparing functional and spatial representations. All the errors are expressed in meters. On the right, two examples of functional and pointwise transfer. In some cases, the difference is minimal, while the functional approach requires a step less; on the animal the functional transfer is noticeably better than the one induced by the pointwise correspondence.

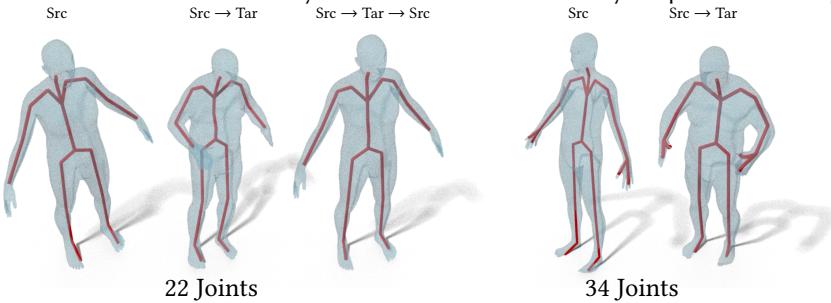


Fig. 6. Transfer examples using two different skeletons from Mixamo. On the left, a version with 22 joints; on the right, one composed of 34 joints. Please note the additional joints in the hands region on the second pair. We remark the coherence and independence from the involved skeleton of the two results.

We would remark that the dense matching is a step built upon the functional approach (see Section 3.2); it cannot be obtained without computing the functional maps before. Our method obtains stable results, despite the variabilities. The results achieved by our functional approach are similar to the ones obtained by applying the SMPL regressor to the point-to-point map obtained by functional map conversion, showing that the latter is not required. Solving for a point-to-point correspondence is challenging for these shapes, and there is not always an exact solution. We report a result on SMAL in Figure 5, highlighting the difference in their connectivity. Also, in Figure 6, we tested our method on two different skeletons from Mixamo², one with 22 joints and the extended version with 34 joints that also control the hands. We show that our method can be applied to different skeletons and that we can represent the joints on terminal protrusions (i.e., hands) without impacting the global results on the rest of the shape. Furthermore, in Figure 7 we tested our resilience to the presence of surface noise, by transferring the skeleton to a shape with clothes and hair. The qualitative result is convincing.

Comparison. We compare our skeleton transfer approach against the method proposed in [Marin et al. 2020], which we consider our baseline. This method consists of transferring the coordinates of the target into the source space and then applying the spatial regressor to the low-pass representation. While the two approaches follow similar principles, [Marin et al. 2020] applies the spatial regressor to a degraded representation of the target mesh. This combination

²www.mixamo.com

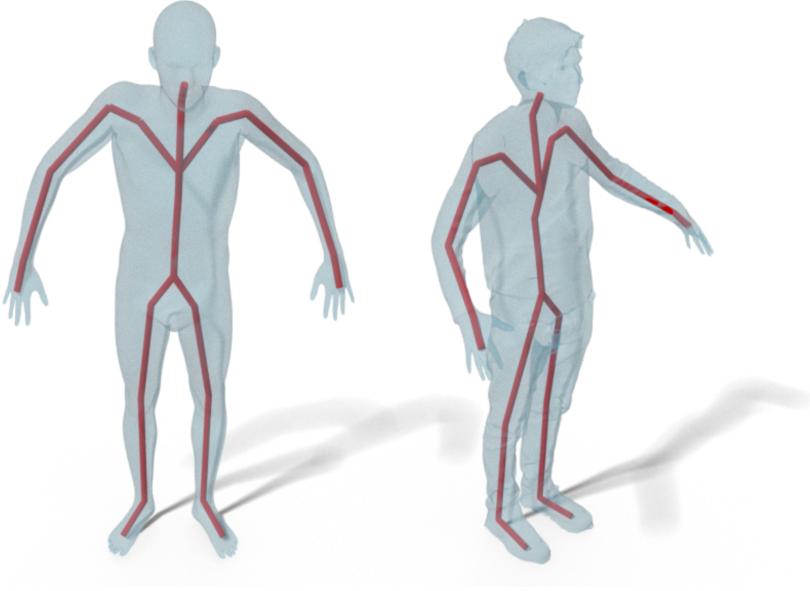


Fig. 7. A transfer of a skeleton between a FAUST shape and a clothed model. Although the differences in the surface of the two shapes, our regressor can transfer the skeleton on the clothed model.

of a high-frequency operator with low-frequency structure generates undesirable artifacts, and it cannot be considered useful for animation. In Table 2 we report a comparison on the adult human datasets, showing an improvement of more than 20%. Finally, in Figure 8, we compare ourselves against [Avril et al. 2016]. To the best of our knowledge, the code of [Avril et al. 2016] is not publicly available. We thus implement their regressor and apply it to the test shape using the point-to-point mapping obtained by the functional map involved in our method. Our method outputs better results, in particular in the chest region of the shape. The reason is that [Avril et al. 2016] assume the two shapes are in the same pose, while our do not.

5.1 Efficiency and timing

Since efficiency is a critical aspect of skeleton transfer, we measure our method timing on a subset of 54 shapes from Dynamic Faust subsampled with seven different vertex resolutions: 1K, 2K, 3K, 5K, 10K, and 30K. As can be noticed in Figure 9, our method execution time depends on the mapping and is linear on the number of the vertices of the shapes, while other operations have almost a constant cost. In all cases, the overall execution time is under 1 minute, even for 30K resolution meshes. The machine used for this experiment is composed of a Ryzen 7 with a frequency of 3.6 GHz and 64 Gb of RAM. We consider our setting comparable to one of previous methods [Avril et al. 2016] [Moutafidou and Fudos 2019] which relies on a point-to-point correspondence computation; even if we cannot perform a direct comparison due to lack of available code, our method runs significantly faster on shapes of similar resolution (from 2 \times to 10 \times). We remark that we also require less user intervention.

Method	mean	min	max
Dyna Our	0.0278	0.0076	0.0817
Dyna FARM	0.0332	0.0096	0.0815
SHREC19 Our	0.0210	0.0072	0.0512
SHREC19 FARM	0.0295	0.0005	0.1277

Table 2. Comparison with [Marin et al. 2020] for the skeleton transfer from SMPL shape to Dyna and SHREC19 shapes. The average, minimum and maximum errors are expressed in meters and computed as the distance between the estimated and the SMPL regressor joints.

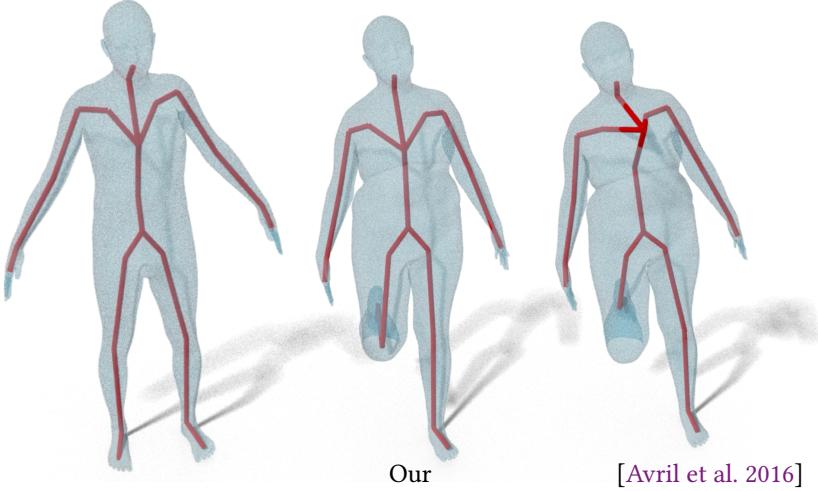


Fig. 8. Comparison between our method and the regressor presented in [Avril et al. 2016].

num. vert.	mapping	regressor	total
1K	10.33 s	11.43 s	24.06 s
2K	12.63 s	12.65 s	27.53 s
3K	13.05 s	12.43 s	26.68 s
5K	12.91 s	11.44 s	26.67 s
10K	17.23 s	12.09 s	31.64 s
30K	40.82 s	14.86 s	57.90 s

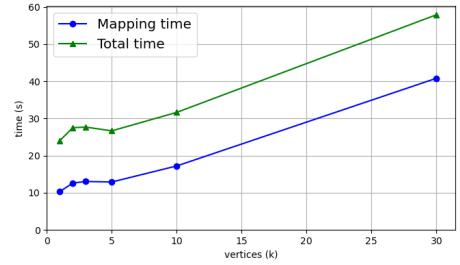


Fig. 9. On the left, we report the execution time of the two main steps of our method with respect to an increasing number of vertices in the target shape. On the right, we highlight the linearity in this relation.

5.2 Skinning weights transfer

While our work focuses on skeleton transfer, we also wondered about our capability to transfer skinning-weights information.

LBS principles Given a shape \mathcal{M} with $n_{\mathcal{M}}$ vertices, the Linear Blend Skinning (LBS) on \mathcal{M} is defined by a set of components that give rise to a semantic and kinematic meaningful deformation of the 3D object represented by the shape \mathcal{M} (e.g. the possible and realistic deformations of a human body). LBS is one of the most popular frameworks for skinning characters due to its efficiency and simplicity. The key idea is defining a relation between each vertex and all the joints of a skeleton as a scalar weight. Hence, the position of a vertex is a weighted sum of the contribution of all the rotations of all the joints:

$$\bar{v}_i = \sum_{q=1}^Q w_{i,q} R'(\theta, J) v_i, \quad (15)$$

Where the transformation $R'(\theta, J)$ takes into account the transformation induced by the hierarchy in the kinematic tree:

$$R'_q(\theta, J) = R_q(\theta, J) R_q(\theta^*, J)^{-1} \quad (16)$$

$$R_j(\theta, J) = \prod_{l \in A(J_q)} \left[\frac{\text{rodr}(\theta_l)}{0} \mid j_l \right] \quad (17)$$

Where rodr represent the Rodrigues formula to convert axis-angle notation to rotation matrix; j_l is the single joint center; $A(J_q)$ is the list of ancestors of J_q in the kinematic tree; $R_q(\theta^*, J)$ is the transformation of joint q in the world frame, and $R'_q(\theta, J)$ is the same after removing the transformation induced by the rest pose θ^* . We refer to [Loper et al. 2015] for further details.

Skinning transfer Transferring the skinning weights consists of moving some high-frequencies details not well represented by the first LBO eigenfunctions. In particular, by using the functional map C , we observe the typical Gibbs phenomenon of Fourier analysis over all the surface which makes the skinning weights on the target too noisy for being usable. However, while a full investigation is beyond our scope, we would suggest a first possible solution. First, we transfer the coordinate functions of the target using our C , mapping them in the SMPL space. This generates a low-pass representation of the target geometry with the connectivity of SMPL. Then, we compute the Euclidean nearest-neighbour in the 3D to transfer the skinning weights defined on SMPL to the target low-pass representation. In this way, the Gibbs phenomenon affects the coordinate, while the nearest-neighbour assignment preserves the skinning weights original values. The results can be appreciated in the *attached video*, where we show an animation transferred from a SMPL shape to a new target shape (a remeshed shape from the FAUST dataset [Bogo et al. 2014]) using the transferred skeleton and skinning. We consider this an exciting direction, which elicits further analysis.

6 SPECTRAL REGRESSOR INSIGHTS

We would emphasize some of the properties of our spectral regressor, which could open it to impactful and innovative applications. To highlight the intrinsic properties of our representation, we show in Figure 10 some results of our regressor transfer. The three shapes A, B and C in the first row are significantly non-isometric to the source shape (depicted on the top left of Figure 3). In the second row, we report the pointwise square values of our functional regressor transferred to these three shapes from the source shape. More explicitly, given C_A , C_B and C_C , the functional maps between the source and the three shapes are: $\widehat{R}_A = \widehat{R}C_A$, $\widehat{R}_B = \widehat{R}C_B$ and $\widehat{R}_C = \widehat{R}C_C$ respectively. In these matrices, the rows correspond to the joints and the columns to the Fourier basis functions. At first sight, some patterns emerge. The joints from the central region of the body (corresponding to the top rows of the matrices) mainly influence the low frequencies (i.e. the first 20-30 columns of the matrix). Instead, the joints from the limbs (the bottom rows of the matrix) involve also high frequencies. The matrices look similar, highlighting a shared structure. However, there are differences between transfers on the same subject (left and middle shapes) and transfer between different identities (the right shape). We visualize these differences in the bottom part of the image. We see how the first two are almost identical, while non-isometric shapes present differences, especially in high frequencies.

We think our intrinsic operator can extract a compact representation of the extrinsic embedding of the shape (i.e. the pose). We believe that it could be beneficial for several pipelines, filling the gap generated by the missing extrinsic information in the intrinsic analysis, as occurs, for example, for the functional maps and the shape difference operators where the skeleton (extrinsic) information could help to solve the standard issues related to the (intrinsic) symmetries of shapes [Ren et al.

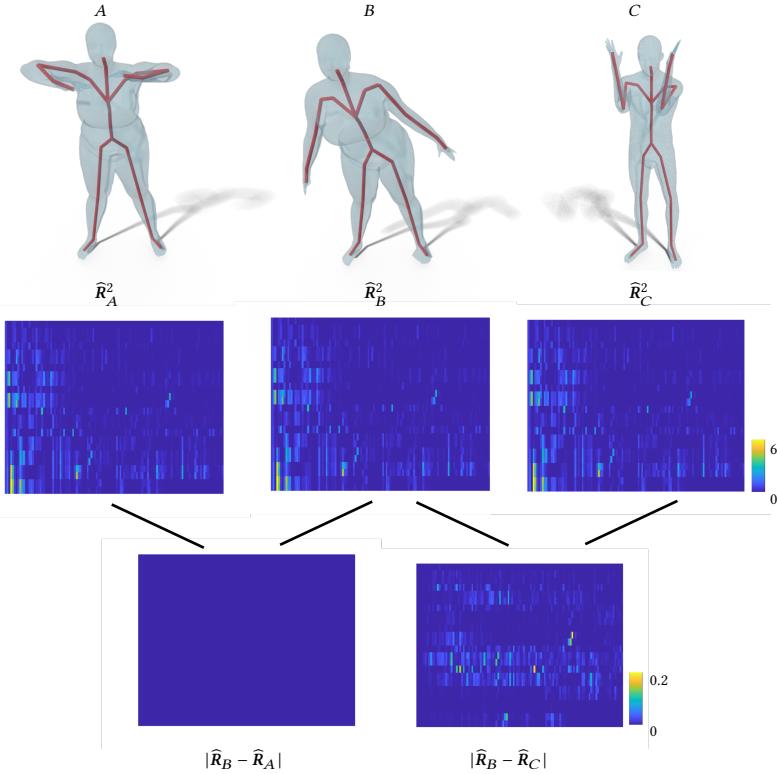


Fig. 10. A visualization of the skeleton transfer on three shapes: A, B, C. In the first row: the obtained results. In the second row: the spectral version of the regressor transferred on the meshes; we squared their values to ignore sign flips of the eigenfunctions. In the third row, we compute the differences between two shapes which share the same identity, and two shape which not. Notice that in the second case it can be noticed a marked difference.

2020]. We propose the exploration of shape collections and the learning of a latent representation in a data-driven pipeline as two interesting future directions.

7 CONCLUSIONS

In this paper, we have shown a straightforward method to transfer skeleton information across shapes efficiently. Our functional approach is agnostic to the mesh discretization and lets us transfer between the meshes without solving directly for a point-to-point correspondence. Our study has several limits: our experiments are limited only to LBS, while theoretically, there is no limit to other animation settings (e.g., deformation cages). Also, while we stressed our method on some challenging cases (noisy, broken, non-isometric shapes), we think many other cases would be interesting to analyze (e.g., topological issues, clutter). As future work, we plan to explore our new representation, for example, by studying its behavior on different skeletons.

8 ACKNOWLEDGEMENTS

RM and SM are supported by the ERC Starting Grant No. 802554 (SPECGEO). This work is partially supported by the project of the Italian Ministry of Education, Universities and Research (MIUR) "Dipartimenti di Eccellenza 2018-2022" of the Department of Computer Science of Sapienza University and the Department of Computer Science of Verona University

REFERENCES

- Brett Allen, Brian Curless, and Zoran Popović. 2003. The space of human body shapes: reconstruction and parameterization from range scans. *ACM transactions on graphics (TOG)* 22 (2003), 587–594.
- Quentin Avril, Donya Ghafourzadeh, Srinivasan Ramachandran, Sahel Fallahdoust, Sarah Ribet, Olivier Dionne, Martin de Lasa, and Eric Paquette. 2016. Animation setup transfer for 3D characters. In *Computer Graphics Forum*, Vol. 35. Wiley Online Library, 115–126.
- Ilya Baran and Jovan Popović. 2007. Automatic rigging and animation of 3d characters. *ACM Transactions on graphics (TOG)* 26 (2007), 72–es.
- Jean Basset, Stefanie Wuhrer, Edmond Boyer, and Franck Multon. 2019. Contact preserving shape transfer for rigging-free motion retargeting. In *Motion, Interaction and Games*. 1–10.
- Federica Bogo, Javier Romero, Matthew Loper, and Michael J. Black. 2014. FAUST: Dataset and evaluation for 3D mesh registration. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Piscataway, NJ, USA.
- Federica Bogo, Javier Romero, Gerard Pons-Moll, and Michael J. Black. 2017. Dynamic FAUST: Registering Human Bodies in Motion. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.
- Steve Capell, Matthew Burkhardt, Brian Curless, Tom Duchamp, and Zoran Popović. 2005. Physically based rigging for deformable characters. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*. 301–310.
- Kwang-Jin Choi and Hyeong-Seok Ko. 2000. Online motion retargetting. *The Journal of Visualization and Computer Animation* 11 (2000), 223–235.
- Danielle Ezuz and Mirela Ben-Chen. 2017. Deblurring and Denoising of Maps between Shapes. *Computer Graphics Forum* 36, 5 (2017), 165–174.
- Michael Gleicher. 1998. Retargetting motion to new characters. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*. 33–42.
- Nikolas Hesse, Sergi Pujades, Javier Romero, Michael J. Black, Christoph Bodensteiner, Michael Arens, Ulrich G. Hofmann, Uta Tacke, Mijna Hadders-Algra, Raphael Weinberger, Wolfgang Müller-Felber, and A. Sebastian Schroeder. 2018. Learning an Infant Body Model from RGB-D Data for Accurate Full Body Motion Analysis. In *Int. Conf. on Medical Image Computing and Computer Assisted Intervention (MICCAI)*.
- Alec Jacobson, Ilya Baran, Ladislav Kavan, Jovan Popović, and Olga Sorkine. 2012. Fast Automatic Skinning Transformations. *ACM Trans. Graph.* 31, 4, Article 77 (July 2012), 10 pages.
- Taeil Jin, Meekyoung Kim, and Sung-Hee Lee. 2018. Aura mesh: Motion retargeting to preserve the spatial relationships between skinned characters. In *Computer Graphics Forum*, Vol. 37. Wiley Online Library, 311–320.
- Binh Huy Le and Zhigang Deng. 2012. Smooth Skinning Decomposition with Rigid Bones. *ACM Trans. Graph.* 31, 6, Article 199 (Nov. 2012), 10 pages. <https://doi.org/10.1145/2366145.2366218>
- Binh Huy Le and Zhigang Deng. 2014. Robust and Accurate Skeletal Rigging from Mesh Sequences. *ACM Trans. Graph.* 33, 4, Article 84 (July 2014), 10 pages. <https://doi.org/10.1145/2601097.2601161>
- Jehee Lee and Sung Yong Shin. 1999. A hierarchical approach to interactive motion editing for human-like figures. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. 39–48.
- B. Lévy. 2006. Laplace-Beltrami Eigenfunctions Towards an Algorithm That Understands Geometry. In *Proc. SMI*. IEEE, Washington, DC, 13–25.
- Zhiguang Liu, Antonio Mucherino, Ludovic Hoyet, and Franck Multon. 2018. Surface based motion retargeting by preserving spatial relationship. In *Proceedings of the 11th Annual International Conference on Motion, Interaction, and Games*. 1–11.
- Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. 2015. SMPL: A skinned multi-person linear model. *ACM transactions on graphics (TOG)* 34 (2015), 1–16.
- Riccardo Marin, Simone Melzi, Emanuele Rodolà, and Umberto Castellani. 2020. Farm: Functional automatic registration method for 3d human bodies. In *Computer Graphics Forum*, Vol. 39. Wiley Online Library, 160–173.
- Simone Melzi, Riccardo Marin, Pietro Musoni, Filippo Bardon, Marco Tarini, and Umberto Castellani. 2020. Intrinsic/extrinsic embedding for functional remeshing of 3D shapes. *Computers & Graphics* 88 (2020), 1 – 12.
- Simone Melzi, Riccardo Marin, Emanuele Rodolà, Umberto Castellani, Jing Ren, Adrien Poulenard, Peter Wonka, and Maks Ovsjanikov. 2019a. SHREC 2019: Matching Humans with Different Connectivity. In *Eurographics Workshop on 3D Object*

- Retrieval.* The Eurographics Association.
- Simone Melzi, Jing Ren, Emanuele Rodolà, Peter Wonka, and Maks Ovsjanikov. 2019b. ZoomOut: Spectral Upsampling for Efficient Shape Correspondence. arXiv:1904.07865 [cs.GR]
- Mark Meyer, Mathieu Desbrun, Peter Schröder, and Alan H Barr. 2003. Discrete Differential-Geometry Operators for Triangulated 2-Manifolds. In *Visualization and mathematics III*. Springer, New York, NY, 35–57.
- Anastasia Moutafidou and Ioannis Fudos. 2019. A Mesh Correspondence Approach for Efficient Animation Transfer.. In *CGVC*. 129–133.
- Saifeng Ni, Ran luo, Yue Zhang, Madhukar Budagavi, Andrew Joseph Dickerson, Abhishek Nagar, and Xiaohu Guo. 2020. Scan2Avatar: Automatic Rigging for 3D Raw Human Scans. In *ACM SIGGRAPH 2020 Posters* (Virtual Event, USA) (*SIGGRAPH '20*). Association for Computing Machinery, New York, NY, USA, Article 15, 2 pages. <https://doi.org/10.1145/3388770.3407420>
- Dorian Nogneng, Simone Melzi, Emanuele Rodolà, Umberto Castellani, Michael Bronstein, and Maks Ovsjanikov. 2018. Improved Functional Mappings via Product Preservation. *Computer Graphics Forum* 37, 2 (2018), 179–190.
- Dorian Nogneng and Maks Ovsjanikov. 2017. Informative Descriptor Preservation via Commutativity for Shape Matching. *Computer Graphics Forum* (2017).
- Verónica Costa Orvalho, Ernesto Zacur, and Antonio Susin. 2008. Transferring the rig and animations from a character to different face models. In *Computer Graphics Forum*, Vol. 27. Wiley Online Library, 1997–2012.
- Maks Ovsjanikov, Mirela Ben-Chen, Justin Solomon, Adrian Butscher, and Leonidas Guibas. 2012. Functional maps: a flexible representation of maps between shapes. *ACM Transactions on Graphics (TOG)* 31, 4 (2012), 30:1–30:11.
- Gautam Pai, Jing Ren, Simone Melzi, Peter Wonka, and Maks Ovsjanikov. 2021. Fast Sinkhorn Filters: Using Matrix Scaling for Non-Rigid Shape Correspondence with Functional Maps. In *CVPR*.
- Gopika Rajendran and Ojus Thomas Lee. 2020. Virtual Character Animation based on Data-driven Motion Capture using Deep Learning Technique. In *2020 IEEE 15th International Conference on Industrial and Information Systems (ICIIS)*. IEEE, 333–338.
- Jing Ren, Simone Melzi, Maks Ovsjanikov, and Peter Wonka. 2020. MapTree: Recovering Multiple Solutions in the Space of Maps. *ACM Trans. Graph.* 39, 6 (Nov. 2020).
- Jing Ren, Mikhail Panine, Peter Wonka, and Maks Ovsjanikov. 2019. Structured regularization of functional map computations. *Computer Graphics Forum* 38, 5 (2019), 39–53.
- Jing Ren, Adrien Poulenard, Peter Wonka, and Maks Ovsjanikov. 2018. Continuous and Orientation-preserving Correspondences via Functional Maps. *ACM Transactions on Graphics (TOG)* 37, 6 (2018).
- Emanuele Rodolà, Michael Moeller, and Daniel Cremers. 2015. Point-wise Map Recovery and Refinement from Functional Correspondence. In *Proc. Vision, Modeling and Visualization (VMV)*.
- Jaewoo Seo, Yeongho Seol, Daehyeon Wi, Younghui Kim, and Junyong Noh. 2010. Rigging transfer. *Computer Animation and Virtual Worlds* 21, 3-4 (2010), 375–386.
- Çağlar Seylan and Yusuf Sahillioglu. 2019. 3D skeleton transfer for meshes and clouds. *Graphical Models* 105 (2019), 101041.
- Nicholas Sharp and Keenan Crane. 2020. A Laplacian for Nonmanifold Triangle Meshes. *Computer Graphics Forum (SGP)* 39, 5 (2020).
- Gabriel Taubin. 1995. A Signal Processing Approach to Fair Surface Design. In *Proc. CGIT*. ACM, New York, NY, 351–358.
- B. Vallet and B. Lévy. 2008. Spectral Geometry Processing with Manifold Harmonics. *Computer Graphics Forum* 27, 2 (2008), 251–260.
- Baorong Yang, Junfeng Yao, and Xiaohu Guo. 2018. DMAT: Deformable medial axis transform for animated mesh approximation. In *Computer Graphics Forum*, Vol. 37. Wiley Online Library, 301–311.
- Silvia Zuffi, Angjoo Kanazawa, David Jacobs, and Michael J. Black. 2017. 3D Menagerie: Modeling the 3D Shape and Pose of Animals. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.