Fast Blended Transformations for Partial Shape Registration

Alon Shtern[*] · **Matan Sela**[*] · **Ron Kimmel**
Technion - Israel Institute of Technology

ashtern@campus.technion.ac.il
matansel@cs.technion.ac.il
ron@cs.technion.ac.il

Abstract Automatic estimation of skinning transformations is a popular way to deform a single reference shape into a new pose by providing a small number of control parameters. We generalize this approach by efficiently enabling the use of multiple exemplar shapes. Using a small set of representative natural poses, we propose to express an unseen appearance by a low-dimensional linear subspace, specified by a redundant dictionary of weighted vertex positions. Minimizing a nonlinear functional that regulates the example manifold, the suggested approach supports local-rigid deformations of articulated objects, as well as nearly isometric embeddings of smooth shapes. A real-time non-rigid deformation system is demonstrated, and a shape completion and partial registration framework is introduced. These applications can recover a target pose and implicit inverse kinematics from a small number of examples and just a few vertex positions. The result reconstruction is more accurate compared to state-of-the-art reduced deformable models.
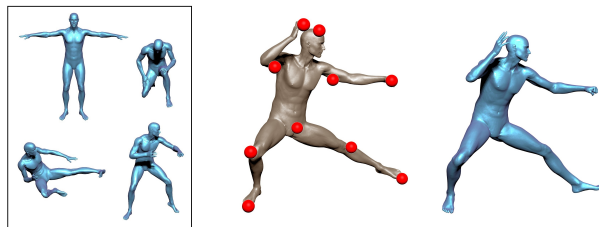
Fig. 1: A shape obtained by blended transformation using ten positional constraints and four reference shapes.

1 Introduction

The construction of an efficient automatic procedure that deforms one shape into another in a natural manner is a fundamental and well-studied challenge in computer graphics. Professional animators design deformable models for manually editing facial expressions, controlling postures and muscles of shapes, and creating sequences of gestures and motions of animated objects. Such models also play a key role in the field of shape analysis. For example, elastic surface registration techniques try to iteratively warp given shapes so as to establish an optimal alignment between them.

A major challenge in automatic shape deformation is preserving the expressiveness of the model while reducing its complexity. This can be accomplished by exploiting the potential redundancy in natural motions. For instance, in non-rigid articulated objects as hands, the bending of a single finger mainly influences the movement of nearby skin. The stiffness of the limbs restricts them to move freely and therefore the deformation of a shape as a whole can often be well approximated as a blend of a small number of affine transformations. One such skeletal deformation technique, the *Linear Blend Skinning* (LBS) [31], has been widely adopted by the gaming and the film industries due to its simplicity and efficiency.

More recently, Jacobson et al. [17] suggested to deform a single shape by looking for transformations that minimize the nonlinear *As-Rigid-As-Possible* (ARAP) energy [2,37]. This energy penalizes deviations from rigidity of the underlying structural skeleton. The optimization process alternates between finding the minimal affine transformations and projecting them onto the group of rigid ones. The algorithm converges after a few iterations and provides realistic deformations with a low computational effort. The method was designed for modifying a single reference shape. As such, it does not effectively incorporate the nature of plausible non-rigid deformations that can be well captured by a few examples. Therefore, this method requires a manually tailored pre-computation of biharmonically smooth blending functions, and relies on an initial pose of the shape that is usually selected as the previous frame in the motion sequence.

In many situations, while analyzing or synthesizing shapes, neither manual input nor the temporal state of the shape at the previous frame is available. In these circumstances, obtaining a natural initial pose for the nonlinear optimization procedure becomes a challenge. Nevertheless, in many of these events, static poses of the *same* shape might be available, such as in [6] where several human bodies in various postures were captured and reconstructed using range scanners. In this paper, we present an efficient generalization of the LBS model for the case where multiple exemplar shapes are available. To that end, the proposed framework uses the reference shapes to infer an expressive yet low dimensional model, which is computationally efficient and produces natural looking poses. The proposed method constructs a dictionary that contains prototype signal-atoms of weighted vertex coordinates, that effectively span the space of deformations represented by the exemplar shapes. We refer to [13], for applications of overcomplete dictionaries for sparse and redundant data representations in other domains.

The proposed algorithm is mainly motivated by the nonrigid 3D partial registration problem. This problem is considered a key challenge in the field of shape analysis. One of the most efficient approaches to solve this challenge is using deformation-driven correspondences [45]. A good deformation method for this purpose should efficiently produce plausible deformations that fits some known constraints. In our setting we use several example shapes and a few known vertex positions. Although some example-based methods produce excellent deformations, in this context of partial registration, they usually carry three major drawbacks. First, most of these methods have high complexity. Second, they depend strongly on a good initial shape alignment. Third, they require many examples for constructing a model which plausibly captures various poses. The proposed method tries to overcome these difficulties by using a redundant dictionary that spans a linear deformation subspace. The advantages of using a linear subspace are evident. Acceleration in this case is well established using the ARAP energy functional. Additionally, well known regularization techniques, such as L_1 and L_2 penalty terms, can easily be deployed in conjunction with the linear model to find a robust sparse representation for the initial shape alignment. Moreover, the simplicity and flexibility of using the linear representation enables the proposed algorithm to refine this initial shape deformation by gradually expanding the deformation space while simultaneously introducing more accurate model constraints.

The key contributions of the proposed approach include the following features.

– Given a few reference shapes, we construct a redundant, yet compact, dictionary of weighted positional-atoms that spans a rich space of deformations. A new deformation is represented as a linear combination of these atomsignals.
– Stable transformations are established by using sparse modeling over a limited subspace of deformations. The suggested framework ensures the use of only a few dictionary atoms relating a few given poses to a target one.
– The As-Rigid-As-Possible energy is reformulated to support multiple reference shapes and automatic global scale detection.
– Smooth deformations are realized by an additional biharmonic energy term that is computationally efficient to minimize when the skinning weights are set to be the eigenfunctions of the Laplace-Beltrami operator.

To demonstrate the fast blended transformations approach, animation sequences were generated given just a few reference shapes and a handful of point constraints that define each target frame. Quantitative evaluation indicates that the advantages of the proposed approach are fully realized when plugged into a shape completion and registration application that achieves low correspondence errors and deformation distortions.

2 Related efforts

Example-based deformation techniques attempt to establish a compact representation of shape deformations while trying to satisfy desirable properties. Forming these representations generally requires the processing of sets of poses, expressions, or identities of the same class of shapes. To fulfill this task, various methods have been proposed. Roughly speaking, they all share the following taxonomy.

Displacement field interpolation. This technique computes the pointwise difference between each example shape and a reference one at a resting pose, see for example [27, 36,24]. More recent methods include statistical [14] and rotational regressions [41].

Deformation gradient. These methods interpolate the example poses using the gradient fields of the coordinate functions, and construct the deformed surface by solving a Poisson equation. In [44,39] the deformation is estimated for each triangle of the given mesh. Example based deformation gradients and its variants, like the Green strain tensor, are also used for static or dynamic simulation of elastic materials [32,23,7,35,46]. For lowering the computational cost, Der et al. [11] proposed to cluster triangles that are subject to a similar rigid rotation with respect to a single reference shape. It allowed reformulating the problem in terms of transformations of a representative proxy point for each group of vertices.

Edge lengths and dihedral angles interpolation. Inspired by discrete shells [16], local properties were used for mesh interpolation [43], that naturally fits with the discrete shell energy for combined physics-based and example-driven mesh deformations [15].

Transformation blending. This approach describes the deformation by a set of affine transformations that are blended together to represent the deformed shape. In this case, the example shapes are used to find the skinning weights as well as the transformations by using non-linear optimization algorithms [19,21,25,26].

Linear subspace. Similar in its spirit to the proposed approach is Tycowicz et al. [40]. Their method computes an example-based reduced linear model for representing the high dimensional shape space using deformation energy derivatives and Krylov sequences. However, their framework and reduced linear subspace are specifically designed and restricted to the nonlinear shape interpolation problem.

The fast blended transformations method is affiliated with the class of transformation blending inspired by [17,42]. The deformation is performed by minimizing a nonlinear energy functional over the linear subspace of skinning transformations. Unlike previous efforts, we suggest to simultaneously blend affine transformations of several given poses of the same subject. The proposed framework allows us to learn the example manifold without estimating the explicit connections between the reference shapes. With these reference shapes, we construct an overcomplete dictionary that spans the space of allowed deformations up to a small tolerance. The nonlinear energy functional guides the transformations to achieve a physically-plausible deformation. Projecting a small set of constraints to the examples manifold, which is assumed to be of low dimensions, we obtain an efficient and accurate blending procedure for real time animation and for the partial shape registration task.

3 Notations and problem formulation

3.1 Linear blend skinning

Here, we follow the blend skinning model as described by Jacobson et al. in [17]. Let $\mathbf{v}_1, \ldots, \mathbf{v}_n \in \mathbb{R}^d$ ($d = 3$) be the vertex positions of the input reference mesh $\mathcal{M}$ with f triangles and n vertices. Denote the deformed vertex positions of a new target mesh $\tilde{\mathcal{M}}$ by $\tilde{\mathbf{v}}_1, \ldots, \tilde{\mathbf{v}}_n \in \mathbb{R}^d$. The target vertex positions relate to the given reference vertices through m affine transformation matrices $\mathbf{M}_j \in \mathbb{R}^{d \times (d+1)}$, $j = \{1, \ldots, m\}$ and real-valued skinning weight functions w_j, that measure the influence of each affine transformation on each point of the shape. For a discrete mesh, we denote $w_j(\mathbf{v}_i)$ by $w_{j,i}$, and readily have

$$\tilde{\mathbf{v}}_i = \sum_{j=1}^m w_{j,i} \mathbf{M}_j \begin{pmatrix} \mathbf{v}_i \\ 1 \end{pmatrix}. \tag{1}$$

Equation (1) can be rewritten in a matrix form as

$$\tilde{\mathbf{V}} = \mathbf{D}_{\text{LBS}} \mathbf{T}_{\text{LBS}},$$

where $\tilde{\mathbf{V}} \in \mathbb{R}^{n \times d}$ is the matrix whose rows are the positions of the target vertices, and the matrices $\mathbf{T}_{\text{LBS}} \in \mathbb{R}^{(d+1)m \times d}$ and $\mathbf{D}_{\text{LBS}} \in \mathbb{R}^{n \times (d+1)m}$ are created by stacking the skinning parameters in the following fashion

$$\mathbf{D}_{\text{LBS}} = \begin{pmatrix} w_{1,1} \left(\mathbf{v}_1^{\text{T}}, 1 \right) & \ldots & w_{m,1} \left(\mathbf{v}_1^{\text{T}}, 1 \right) \\ \vdots & \ddots & \vdots \\ w_{1,n} \left(\mathbf{v}_n^{\text{T}}, 1 \right) & \ldots & w_{m,n} \left(\mathbf{v}_n^{\text{T}}, 1 \right) \end{pmatrix},$$

$$\mathbf{T}_{\text{LBS}} = \begin{pmatrix} \mathbf{M}_1 \ldots \mathbf{M}_m \end{pmatrix}^{\text{T}}.$$

3.2 Fast automatic skinning transformations

The most general form of representing the position of a new target vertex by a linear transformation of some dictionary (such as the linear blend skinning formulation) can be expressed by

$$\tilde{\mathbf{V}} = \mathbf{D}\mathbf{T},$$

where $\mathbf{D} \in \mathbb{R}^{n \times b}$ is a dictionary of size b (in case of standard linear blend skinning $b = (d+1)m$), and $\mathbf{T} \in \mathbb{R}^{b \times d}$ is a matrix of unknown coefficients that represents the vertex positions in terms of the dictionary.

Jacobson et al. [17] introduced a method for automatically finding the skinning transformations $\mathbf{T}$ by minimizing the ARAP energy [37,30,10] between the reference shape $\mathcal{M}$ and the target one $\tilde{\mathcal{M}}$. Let $\mathbf{R}_1, \mathbf{R}_2, \ldots, \mathbf{R}_r \in \text{SO}(d)$ and $\mathcal{E}_1, \mathcal{E}_2, \ldots, \mathcal{E}_r$ be r local rotations and their corresponding

edge sets, respectively. The ARAP energy, which measures local deviation from rigidity, can be expressed as

$$E(\mathbf{V}, \tilde{\mathbf{V}}) = \frac{1}{2} \sum_{k=1}^{r} \sum_{(i,j) \in \mathcal{E}_k} c_{ijk} \|(\tilde{\mathbf{v}}_i - \tilde{\mathbf{v}}_j) - \mathbf{R}_k(\mathbf{v}_i - \mathbf{v}_j)\|^2,$$

where $c_{ijk} \in \mathbb{R}$ are the cotangent weighting coefficients [34]. As indicated in [17], it is unnecessary to estimate the local rotation for each edge separately since vertices undergoing similar deformations can be clustered together into a small number of rotation clusters.

The ARAP energy can be expressed in a simple matrix form. Denote $\mathbf{A}_k \in \mathbb{R}^{n \times |\mathcal{E}_k|}$ as the directed incidence matrix corresponding to edges $\mathcal{E}_k$, and let $\mathbf{C}_k \in \mathcal{R}^{|\mathcal{E}_k| \times |\mathcal{E}_k|}$ be a diagonal matrix with weights c_{ijk}. Then, the ARAP energy can be written in matrix form as

$$2E(\mathbf{V}, \tilde{\mathbf{V}}) = \operatorname{tr}(\tilde{\mathbf{V}}^{\mathrm{T}} \mathbf{L} \tilde{\mathbf{V}}) - 2 \operatorname{tr}(\mathbf{R} \mathbf{K} \tilde{\mathbf{V}}) + \operatorname{tr}(\mathbf{V}^{\mathrm{T}} \mathbf{L} \mathbf{V}),$$

where $\mathbf{R} = (\mathbf{R}_1, \dots, \mathbf{R}_r)$, $\mathbf{K} \in \mathbb{R}^{dr \times n}$ stacks the matrices $\mathbf{V}^{\mathrm{T}} \mathbf{A}_k \mathbf{C}_k \mathbf{A}_k^{\mathrm{T}}$, and $\mathbf{L} \in \mathbb{R}^{n \times n}$ is the cotangent-weights Laplacian up to a constant scale factor. Plugging in the linear blend skinning formula $\tilde{\mathbf{V}} = \mathbf{D} \mathbf{T}$ we obtain

$$2E(\mathbf{V}, \tilde{\mathbf{V}}) = \operatorname{tr}(\mathbf{T}^{\mathrm{T}} \tilde{\mathbf{L}} \mathbf{T}) - 2 \operatorname{tr}(\mathbf{R} \tilde{\mathbf{K}} \mathbf{T}) + \operatorname{tr}(\mathbf{V}^{\mathrm{T}} \mathbf{L} \mathbf{V}), \quad (2)$$

where $\tilde{\mathbf{L}} = \mathbf{D}^{\mathrm{T}} \mathbf{L} \mathbf{D}$ and $\tilde{\mathbf{K}} = \mathbf{K} \mathbf{D}$. For more details about the above derivation, we refer the reader to [17].

4 Example-based blended transformations

Overview. We now extend the framework described in the previous section for the case where multiple poses of the same shape are available. We begin by expressing the deformed shape as a combination of atoms from a dictionary that is constructed from the linear blend skinning matrices of the given examples. Then, we provide the details of various energy terms to be minimized with respect to the unknown transformations $\mathbf{T}$ using the proposed model. Next, we describe the nonlinear optimization process and its initialization, and conclude by discussing optional extensions that can be incorporated into the algorithm.

4.1 Dictionary Construction

Suppose we are given q reference meshes $\mathcal{M}_1, \mathcal{M}_2, \dots, \mathcal{M}_q$. Let $\mathbf{v}_1^\ell, \dots, \mathbf{v}_n^\ell \in \mathbb{R}^d$ be the positions of vertices belonging to the reference mesh $\mathcal{M}_\ell$, $\ell = 1, \dots, q$, and let $\mathbf{V}_1, \mathbf{V}_2, \dots, \mathbf{V}_q \in \mathbb{R}^{n \times d}$ be the matrices whose rows denote the positions of the corresponding vertices. We are also given some h linear constraints represented by the matrix $\mathbf{H} \in \mathbb{R}^{h \times n}$, such that $\mathbf{H} \tilde{\mathbf{V}} \approx \mathbf{Y}$, where $\mathbf{Y} \in \mathbb{R}^{h \times d}$ is the value of these constraints for the target shape. We can define the linear constraints to be simply the coordinates of points on the mesh

or use more refined measures such as the Laplacian coordinates [1,29], or a weighted average of some vertex positions, to constrain our non-rigid blended shape deformation. Using this setup, we are interested in finding the positions of the target vertices as a result of a smooth transformation of the input meshes such that it approximately preserves local rigidity and satisfies the linear constraints up to a small error.

Example-based dictionary. Given m real-valued weight functions w_j, $j = 1, \dots, m$, we propose the example-based representation of the positions of the target vertices to be a combination of the linear blend skinning deformations of each given reference mesh

$$\tilde{\mathbf{v}}_i = \sum_{\ell=1}^{q} \tilde{\mathbf{v}}_i^\ell,$$

where

$$\tilde{\mathbf{v}}_i^\ell = \sum_{j=1}^{m} w_{j,i} \mathbf{M}_j^\ell \begin{pmatrix} \mathbf{v}_i^\ell \\ 1 \end{pmatrix}. \quad (3)$$

We can explicitly write the new vertex positions as

$$\begin{aligned} \tilde{\mathbf{v}}_i &= \sum_{\ell=1}^{q} \sum_{j=1}^{m} w_{j,i} \mathbf{M}_j^\ell \begin{pmatrix} \mathbf{v}_i^\ell \\ 1 \end{pmatrix} \\ &= \sum_{\ell=1}^{q} \sum_{j=1}^{m} w_{j,i} \hat{\mathbf{M}}_j^\ell \mathbf{v}_i^\ell + \sum_{j=1}^{m} \sum_{\ell=1}^{q} w_{j,i} \bar{\mathbf{M}}_j^\ell, \end{aligned} \quad (4)$$

where $\hat{\mathbf{M}}_j^\ell \in \mathbb{R}^{d \times d}$ and $\bar{\mathbf{M}}_j^\ell \in \mathbb{R}^{d \times 1}$ are sub-matrices of $\mathbf{M}_j^\ell$, such that $\mathbf{M}_j^\ell = \left(\hat{\mathbf{M}}_j^\ell, \bar{\mathbf{M}}_j^\ell \right)$. This formula can be equivalently expressed in the standard matrix form by

$$\tilde{\mathbf{V}} = \mathbf{D} \mathbf{T},$$

where $\mathbf{D} \in \mathbb{R}^{n \times (1+qd)m}$ is the proposed dictionary of size $b = (1 + qd)m$, that multiplies the examples' vertex positions $\mathbf{v}_i^\ell$ with the vertex weights $w_j(v_i)$, and $\mathbf{T} \in \mathbb{R}^{(1+qd)m \times d}$ stacks the matrices $\hat{\mathbf{M}}_j^m$ and $\bar{\mathbf{M}}_j^m$ in the following way

$$\mathbf{D} = \left(\bar{\mathbf{D}}, \hat{\mathbf{D}}_1 \dots \hat{\mathbf{D}}_q \right),$$
$$\mathbf{T} = \left(\bar{\mathbf{T}}^{\mathrm{T}}, \hat{\mathbf{T}}_1^{\mathrm{T}} \dots \hat{\mathbf{T}}_q^{\mathrm{T}} \right)^{\mathrm{T}},$$

where

$$\hat{\mathbf{D}}_\ell = \begin{pmatrix} w_{1,1} \mathbf{v}_1^{\ell\,\mathrm{T}} & \dots & w_{m,1} \mathbf{v}_1^{\ell\,\mathrm{T}} \\ \vdots & \ddots & \vdots \\ w_{1,n} \mathbf{v}_n^{\ell\,\mathrm{T}} & \dots & w_{m,n} \mathbf{v}_n^{\ell\,\mathrm{T}} \end{pmatrix},$$
$$\hat{\mathbf{T}}_\ell = \left(\hat{\mathbf{M}}_1^\ell \dots \hat{\mathbf{M}}_m^\ell \right)^{\mathrm{T}},$$
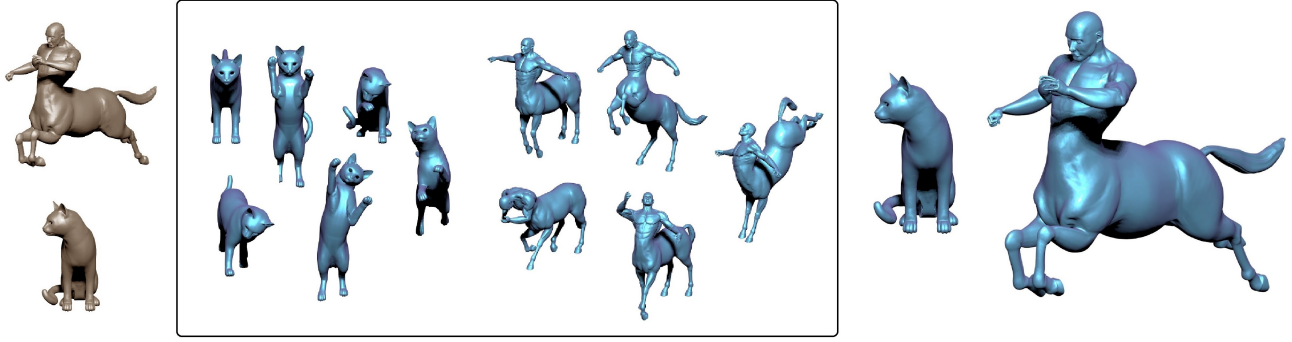
Fig. 2: Deformation using the example-based LBO dictionary. The left portion of the figure shows the cat and centaur *ground-truth* target shapes (colored in gray). On the right we show the near perfect representation of these target shapes by a linear combination of the dictionary's atoms. In this case, the weighting functions are the eigenfunctions of the Laplace-Beltrami operator that correspond to the lowest 15 eigenvalues. The exemplar shapes that are used to extract the example-based LBO dictionaries for representing the shapes are shown inside the box.

and

$$\bar{\mathbf{D}} = \begin{pmatrix} w_{1,1} & \dots & w_{m,1} \\ \vdots & \ddots & \vdots \\ w_{1,n} & \dots & w_{m,n} \end{pmatrix},$$

$$\bar{\mathbf{T}} = \left(\sum_{\ell=1}^{q} \bar{\mathbf{M}}_1^\ell \ \dots \ \sum_{\ell=1}^{q} \bar{\mathbf{M}}_m^\ell \right)^{\mathrm{T}}.$$

Weighting functions. There are many ways to choose weighting functions. One is to consider the weights of bones like in the standard linear blend skinning model. In that case, we name the constructed dictionary as the *example-based skeleton dictionary*. When there is no significant underlying skeletal structure, we suggest to use the first m eigenfunctions of the Laplace-Beltrami operator (LBO) [12,8]. This choice of a dictionary is useful, for example, when handling facial expressions, for analyzing internal organs in volumetric medical imaging applications, or for deforming non-rigid objects such as an octopus. The eigendecomposition of the LBO consists of non-negative eigenvalues $0 = \lambda_0 < \lambda_1 < \dots < \lambda_i < \dots$, with corresponding eigenfunctions $\Phi \equiv \{\phi_0, \phi_1, \dots, \phi_i, \dots\}$, that can be considered as an orthonormal basis. We refer to this dictionary as the *example-based LBO dictionary*.

4.2 Nonlinear Energy Terms

Linear constraints. The energy of the h linear constraints can be calculated by

$$2E_{\mathrm{lc}}(\tilde{\mathbf{V}}) = \left\| \mathbf{H}\tilde{\mathbf{V}} - \mathbf{Y} \right\|_2^2 = \|\mathbf{H}\mathbf{D}\mathbf{T} - \mathbf{Y}\|_2^2 = \|\mathbf{X}\mathbf{T} - \mathbf{Y}\|_2^2$$
$$= \mathrm{tr}\left(\mathbf{T}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{T} \right) - 2\,\mathrm{tr}\left(\mathbf{Y}^{\mathrm{T}}\mathbf{X}\mathbf{T} \right) + \mathrm{tr}\left(\mathbf{Y}^{\mathrm{T}}\mathbf{Y} \right),$$
$$(5)$$

where $\mathbf{X} = \mathbf{H}\mathbf{D}$.

Smoothness energy. Let $\mathbf{v}_{i,k}$, $k \in \{1, \dots, d\}$ be the kth coordinate of the vertex position $\mathbf{v}_i$. Notice from Equation (4), that the amount of influence of $\mathbf{v}_{i,k}^\ell$ on $\tilde{\mathbf{v}}_{i,\tilde{k}}$ is some linear combination of $w_j(v_i)$, $j = 1, \dots, m$. Following the same reasoning as in [18], we search for a smooth variation of this influence, for example, one that minimizes the Laplacian energy $\frac{1}{2} \int_{\mathcal{M}} \Delta(\cdot)^2 da$ of this linear combination, where da is an area element on the surface $\mathcal{M}$ of our shape. For the special case where the weights are the LBO eigenfunctions, the sum of all smoothness energy terms can be expressed as

$$E_{\mathrm{sm}} = \frac{1}{2}\,\mathrm{tr}(\mathbf{T}^{\mathrm{T}}\boldsymbol{\Lambda}\mathbf{T}), \tag{6}$$

where $\boldsymbol{\Lambda}$ is a diagonal matrix. The values of the diagonal are the squares of the eigenvalues of the respective eignefunctions. Thus, in this case, the smoothness energy amounts to a simple quadratic regularization term. Note, that when the weighting functions are chosen in a different way, the smoothness energy expression is a bit more involved.

Scaling. In some applications, there is a scale difference between the example shapes and the linear constraints. To compensate for such a discrepancy, we introduce a scaling factor α into the ARAP energy. It reflects the ratio between the reference shape and the deformed one, in the following manner,

$$E_{\mathrm{sc}}(\mathbf{V}, \tilde{\mathbf{V}}) = \frac{1}{2}\sum_{k=1}^{r} \sum_{(i,j)\in\mathcal{E}_k} c_{ijk} \left\| (\tilde{\mathbf{v}}_i - \tilde{\mathbf{v}}_j) - \alpha \mathbf{R}_k(\mathbf{v}_i - \mathbf{v}_j) \right\|^2.$$
$$(7)$$

Hence, the ARAP energy with the global scale factor reads

$$2E_{\mathrm{sc}}(\mathbf{V}, \tilde{\mathbf{V}}) = \mathrm{tr}(\mathbf{T}^{\mathrm{T}}\tilde{\mathbf{L}}\mathbf{T}) - 2\alpha\,\mathrm{tr}(\mathbf{R}\tilde{\mathbf{K}}\mathbf{T}) + \alpha^2\,\mathrm{tr}(\mathbf{V}^{\mathrm{T}}\mathbf{L}\mathbf{V}). \tag{8}$$

Average ARAP energy. One way to define an example-

based energy functional is by taking the average between all as-rigid-as-possible energies, namely,

$$E_{\text{av}} = \frac{1}{q} \sum_{\ell=1}^{q} E_{\text{sc}}(\mathbf{V}_\ell, \tilde{\mathbf{V}}), \tag{9}$$

with the additional linear constraints and the smoothness energies,

$$E_{\text{total}}(\tilde{\mathbf{V}}) = E_{\text{av}}(\tilde{\mathbf{V}}) + \beta_{\text{lc}} E_{\text{lc}}(\tilde{\mathbf{V}}) + \beta_{\text{sm}} E_{\text{sm}}(\tilde{\mathbf{V}}), \tag{10}$$

where β_{lc}, β_{sm}, are some tuning parameters that control the importance of the linear constraints and the smoothness term. We can simplify this expression, plugging in Equations (6), (5) and (8)

$$\begin{aligned}
2E_{\text{total}}(\tilde{\mathbf{V}}) = {}& \text{tr}(\mathbf{T}^{\text{T}} \tilde{\mathbf{L}} \mathbf{T}) \\
& - \frac{1}{q} \sum_{\ell=1}^{q} (2\alpha \, \text{tr}(\mathbf{R}_\ell \tilde{\mathbf{K}}_\ell \mathbf{T}) + \alpha^2 \, \text{tr}(\mathbf{V}_\ell^{\text{T}} \mathbf{L} \mathbf{V}_\ell)) \\
& + \beta_{\text{lc}} \, \text{tr}(\mathbf{T}^{\text{T}} \mathbf{X}^{\text{T}} \mathbf{X} \mathbf{T}) - 2\beta_{\text{lc}} \, \text{tr}(\mathbf{Y}^{\text{T}} \mathbf{X} \mathbf{T}) \\
& + \beta_{\text{lc}} \, \text{tr}(\mathbf{Y}^{\text{T}} \mathbf{Y}) + \beta_{\text{sm}} \, \text{tr}(\mathbf{T}^{\text{T}} \mathbf{\Lambda} \mathbf{T}). \tag{11}
\end{aligned}$$

Minimal ARAP energy. Another way to define an example-based energy functional is to find the minimal ARAP energy between the deformed mesh and each of the input meshes separately,

$$E_{\text{mn}}(\tilde{\mathbf{V}}) = \min_{\ell} E_\ell(\tilde{\mathbf{V}}, \mathbf{T}_\ell), \tag{12}$$

where

$$E_\ell(\tilde{\mathbf{V}}, \mathbf{T}_\ell) = E_{\text{sc}}(\mathbf{V}_\ell, \tilde{\mathbf{V}}) + \beta_{\text{lc}} E_{\text{lc}}(\tilde{\mathbf{V}}) + \beta_{\text{sm}} E_{\text{sm}}(\tilde{\mathbf{V}}). \tag{13}$$

This can be expressed as

$$\begin{aligned}
2E_\ell(\tilde{\mathbf{V}}, \mathbf{T}_\ell) = {}& \text{tr}(\mathbf{T}_\ell^{\text{T}} \tilde{\mathbf{L}} \mathbf{T}_\ell) - 2\alpha_\ell \, \text{tr}(\mathbf{R}_\ell \tilde{\mathbf{K}}_\ell \mathbf{T}_\ell) \\
& + \alpha_\ell^2 \, \text{tr}(\mathbf{V}_\ell^{\text{T}} \mathbf{L} \mathbf{V}_\ell) + \beta_{\text{lc}} \, \text{tr}(\mathbf{T}_\ell^{\text{T}} \mathbf{X}^{\text{T}} \mathbf{X} \mathbf{T}_\ell) \\
& - 2\beta_{\text{lc}} \, \text{tr}(\mathbf{Y}^{\text{T}} \mathbf{X} \mathbf{T}_\ell) + \beta_{\text{lc}} \, \text{tr}(\mathbf{Y}^{\text{T}} \mathbf{Y}) \\
& + \beta_{\text{sm}} \, \text{tr}(\mathbf{T}_\ell^{\text{T}} \mathbf{\Lambda} \mathbf{T}_\ell). \tag{14}
\end{aligned}$$

4.3 Optimization

To minimize the energy $E_{\text{total}}(\tilde{\mathbf{V}})$ and find the local rotations $\mathbf{R}_\ell$, $\ell = 1, \ldots, q$, the global scale factor α and the transformations $\mathbf{T}$, we follow the local-global approach of [37] with an additional step to find the global scale α. First we fix $\mathbf{T}$ and α and solve for $\mathbf{R}_\ell$ (local step). Then, we find α by fixing $\mathbf{T}$, $\mathbf{R}_\ell$ (scale step). Finally, we fix $\mathbf{R}_\ell$ and α, and solve for $\mathbf{T}$ (global step). *Local step.* For fixed α and $\mathbf{T}$, maximizing $\text{tr}(\mathbf{R}_\ell \mathbf{S}_\ell)$, $\ell = 1, \ldots, q$, where $\mathbf{S}_\ell = \tilde{\mathbf{K}}_\ell \mathbf{T}_\ell$ is constant,

amounts to maximizing $\text{tr}(\mathbf{R}_{\ell,k} \mathbf{S}_{\ell,k})$, $k = 1, \ldots, r$, which is obtained by taking $\mathbf{R}_{\ell,k} = \mathbf{\Psi}_{\ell,k}^{\text{T}} \mathbf{\Phi}_{\ell,k}^{\text{T}}$, where

$$\mathbf{S}_{\ell,k} = \mathbf{\Phi}_{\ell,k} \mathbf{\Sigma}_{\ell,k} \mathbf{\Psi}_{\ell,k}$$

is given by the singular value decomposition of $\mathbf{S}_{\ell,k}$.

Scale step. For fixed $\mathbf{T}$ and $\mathbf{R}_\ell$, $\ell = 1, \ldots, q$, we can differentiate by α

$$\frac{\partial E_{\text{total}}}{\partial \alpha} = -\frac{1}{q} \sum_{\ell=1}^{q} (\text{tr}(\mathbf{R}_\ell \tilde{\mathbf{K}}_\ell \mathbf{T})) + \alpha \, \text{tr}(\mathbf{V}_\ell^{\text{T}} \mathbf{L} \mathbf{V}_\ell). \tag{15}$$

Setting the derivative to zero, we get

$$\alpha = \frac{1}{q} \sum_{\ell=1}^{q} (\text{tr}(\mathbf{R}_\ell \tilde{\mathbf{K}}_\ell \mathbf{T})) / \text{tr}(\mathbf{V}_\ell^{\text{T}} \mathbf{L} \mathbf{V}_\ell). \tag{16}$$

Global step. For fixed α and $\mathbf{R}_\ell$, $\ell = 1, \ldots, q$, we differentiate E_{total}

$$\begin{aligned}
\frac{\partial E_{\text{total}}}{\partial \mathbf{T}} = {}& \frac{1}{q} \sum_{\ell=1}^{q} (\tilde{\mathbf{L}} \mathbf{T} - \alpha \tilde{\mathbf{K}}_\ell^{\text{T}} \mathbf{R}_\ell^{\text{T}}) \\
& + \beta_{\text{lc}} (\mathbf{X}^{\text{T}} \mathbf{X} \mathbf{T} - \mathbf{X}^{\text{T}} \mathbf{Y}) + \beta_{\text{sm}} \mathbf{\Lambda} \mathbf{T} \\
= {}& (\tilde{\mathbf{L}} + \beta_{\text{lc}} \mathbf{X}^{\text{T}} \mathbf{X} + \beta_{\text{sm}} \mathbf{\Lambda}) \mathbf{T} - \beta_{\text{lc}} \mathbf{X}^{\text{T}} \mathbf{Y} \\
& - \frac{\alpha}{q} \sum_{\ell=1}^{q} \tilde{\mathbf{K}}_\ell^{\text{T}} \mathbf{R}_\ell^{\text{T}}. \tag{17}
\end{aligned}$$

Setting these derivatives to zero, we obtain

$$(\tilde{\mathbf{L}} + \beta_{\text{lc}} \mathbf{X}^{\text{T}} \mathbf{X} + \beta_{\text{sm}} \mathbf{\Lambda}) \mathbf{T} = \beta_{\text{lc}} \mathbf{X}^{\text{T}} \mathbf{Y} + \frac{\alpha}{q} \sum_{\ell=1}^{q} \tilde{\mathbf{K}}_\ell^{\text{T}} \mathbf{R}_\ell^{\text{T}}. \tag{18}$$

Let us define $\mathbf{\Gamma} = (\tilde{\mathbf{L}} + \beta_{\text{lc}} \mathbf{X}^{\text{T}} \mathbf{X} + \beta_{\text{sm}} \mathbf{\Lambda})$. Then, we can solve for $\mathbf{T}$ by precomputing the Cholesky factorization of $\mathbf{\Gamma}$

$$\mathbf{T} = \mathbf{\Gamma}^{-1} \Big(\beta_{\text{lc}} \mathbf{X}^{\text{T}} \mathbf{Y} + \frac{\alpha}{q} \sum_{\ell=1}^{q} \tilde{\mathbf{K}}_\ell^{\text{T}} \mathbf{R}_\ell^{\text{T}} \Big). \tag{19}$$

As for optimizing the minimal ARAP energy $E_{\text{mn}}(\tilde{\mathbf{V}})$, in the local step we find each set of rotations $\mathbf{R}_\ell$ by maximizing $\text{tr}(\mathbf{R}_\ell \mathbf{S}_\ell)$, where $\mathbf{S}_\ell = \tilde{\mathbf{K}}_\ell \mathbf{T}_\ell$. We then find the global scale factor relative to each reference shape

$$\alpha_\ell = \text{tr}(\mathbf{R}_\ell \tilde{\mathbf{K}}_\ell \mathbf{T}_\ell) / \text{tr}(\mathbf{V}_\ell^{\text{T}} \mathbf{L} \mathbf{V}_\ell).$$

In the global step we calculate the respective blended transformations $\mathbf{T}_\ell$, by

$$\mathbf{T}_\ell = \mathbf{\Gamma}^{-1} \Big(\beta_{\text{lc}} \mathbf{X}^{\text{T}} \mathbf{Y} + \alpha_\ell \tilde{\mathbf{K}}_\ell^{\text{T}} \mathbf{R}_\ell^{\text{T}} \Big).$$

Then, we calculate the minimal energy $E_\ell(\tilde{\mathbf{V}}, \mathbf{T}_\ell)$, $\ell = 1, \ldots, q$ of Equation (13).

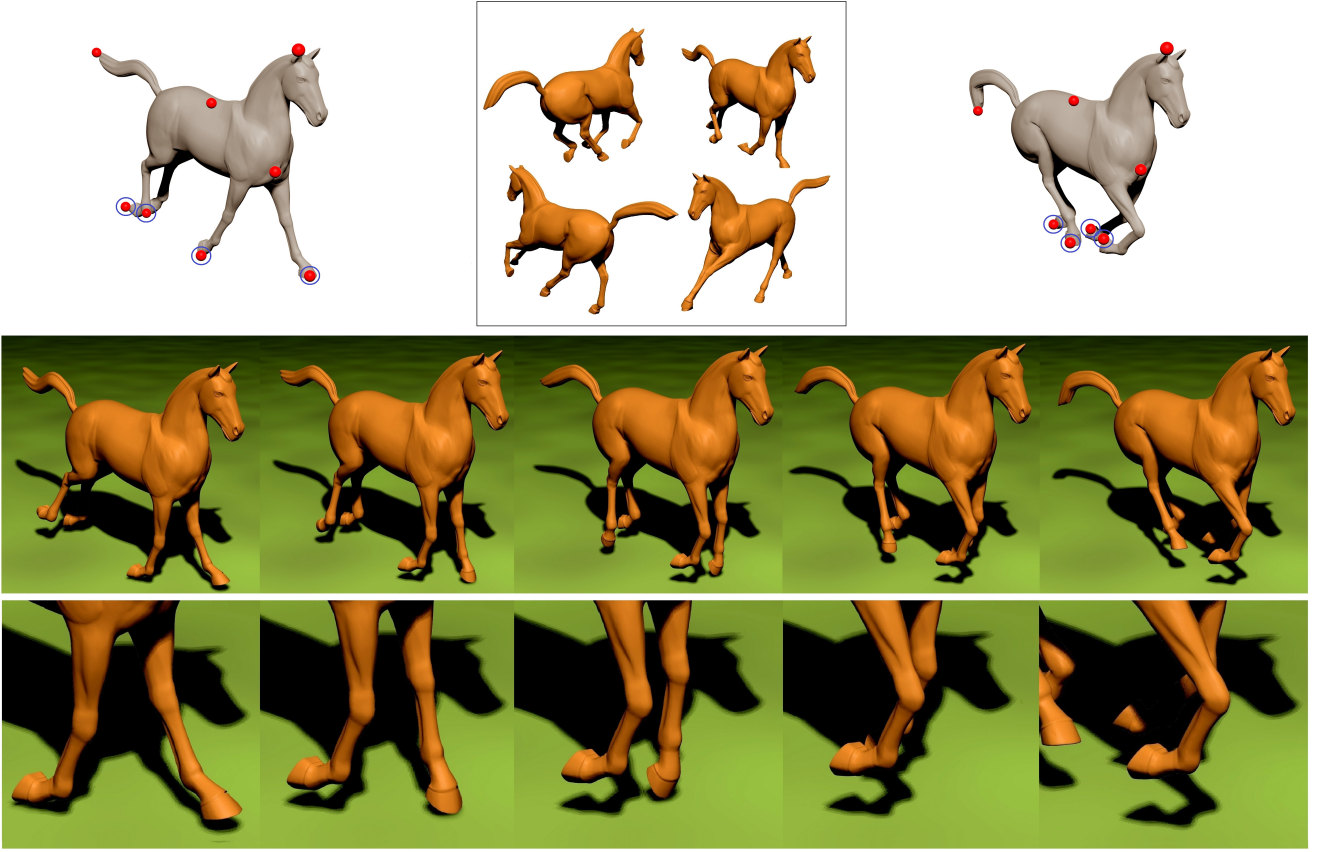**Initial transformations.** In the first global step, there are no

Fig. 3: Automatic feature point correspondence and shape interpolation. The four examples of a horse (top middle) and the two sets of vertex positions (top left, top right) were used to generate a sequence of frames. Correspondence of the points on the four legs (circled in blue) was detected by minimizing the example-based deformation energy for all permissible correspondences. The example-based deformations (bottom left and right) were then interpolated at four times the original frame rate to produce the movie sequence (bottom).

rotation matrices that can be used. Hence, the energy that we need to minimize is

$$
\begin{aligned}
2E_{\mathrm{init}}(\tilde{\mathbf{V}}) =&\beta_{\mathrm{lc}} \|\mathbf{XT} - \mathbf{Y}\|^2 + \beta_{\mathrm{sm}} \mathbf{T}^{\mathrm{T}} \mathbf{\Lambda} \mathbf{T} \\
=&\beta_{\mathrm{lc}} \operatorname{tr}(\mathbf{T}^{\mathrm{T}} \mathbf{X}^{\mathrm{T}} \mathbf{X} \mathbf{T}) - 2\beta_{\mathrm{lc}} \operatorname{tr}(\mathbf{Y}^{\mathrm{T}} \mathbf{X} \mathbf{T}) \\
&+ \beta_{\mathrm{lc}} \operatorname{tr}(\mathbf{Y}^{\mathrm{T}} \mathbf{Y}) + \beta_{\mathrm{sm}} \operatorname{tr}(\mathbf{T}^{\mathrm{T}} \mathbf{\Lambda} \mathbf{T}).
\end{aligned} \quad (20)
$$

We readily have,

$$
\begin{aligned}
\frac{\partial E_{\mathrm{init}}}{\partial \mathbf{T}} =&\beta_{\mathrm{lc}} (\mathbf{X}^{\mathrm{T}} \mathbf{X} \mathbf{T} - \mathbf{X}^{\mathrm{T}} \mathbf{Y}) + \beta_{\mathrm{sm}} \mathbf{\Lambda} \mathbf{T} \\
=&(\beta_{\mathrm{lc}} \mathbf{X}^{\mathrm{T}} \mathbf{X} + \beta_{\mathrm{sm}} \mathbf{\Lambda}) \mathbf{T} - \beta_{\mathrm{lc}} \mathbf{X}^{\mathrm{T}} \mathbf{Y}.
\end{aligned} \quad (21)
$$

Setting these derivatives to zero, we obtain

$$
\mathbf{T} =(\beta_{\mathrm{lc}} \mathbf{X}^{\mathrm{T}} \mathbf{X} + \beta_{\mathrm{sm}} \mathbf{\Lambda})^{-1} (\beta_{\mathrm{lc}} \mathbf{X}^{\mathrm{T}} \mathbf{Y}). \quad (22)
$$

Sparse initial transformations. A more robust initial transformation can be achieved by adding an L_1 penalty to the energy given in Equation (20)

$$
E_{\mathrm{sp}}(\tilde{\mathbf{V}}) = E_{\mathrm{init}}(\tilde{\mathbf{V}}) + \beta_{\mathrm{sp}} \|\mathbf{T}\|_{L_1}. \quad (23)
$$

The effect of this additional penalty is that it makes the initial transformations sparse, which results in a deformation with less artifacts. The parameter β_{sp} controls the amount of sparsity in the initial solution of $\mathbf{T}$. Equation (23) can be solved efficiently using the elastic net regression method [47].

4.4 Extensions

Updating constraints. It may happen that some of the h linear constraints are unavailable due to noise or occlusions. This can be easily solved by deleting the appropriate rows of $\mathbf{X}$ and $\mathbf{Y}$ and efficiently updating the Cholesky factorization.

Dictionary reduction. When the input meshes are similar to each other, the proposed example-based dictionary becomes redundant. The dictionary can be reduced considerably by clustering similar dictionary atoms. For this purpose, we use the k-medoids clustering algorithm [20]. The advantage of

k-medoids over k-means clustering is that each cluster center of the k-medoids procedure is represented by one of the original dictionary atoms. This makes the appearance of the deformed shape more plausible compared to using k-means clustering for dimensionality reduction.

Change of dictionaries. It is sometimes useful to work with two different dictionaries. In that case, the representations of the mesh in these two subspaces can be converted from one to the other in a simple way. Suppose we are given the dictionaries $\mathbf{D}_1$, $\mathbf{D}_2$ and a good approximation of the transformation $\mathbf{T}_1$. Then, the transformation $\mathbf{T}_2$ can be set to

$$\mathbf{T}_2 = (\mathbf{D}_2^{\mathrm{T}} \mathbf{D}_2)^{-1} \mathbf{D}_1^{\mathrm{T}} \mathbf{T}_1. \tag{24}$$

This is particularly useful when one wants to initialize the transformations using a low dimensional dictionary by applying Equation (23), and then change to a richer dictionary for obtaining more refined transformations.
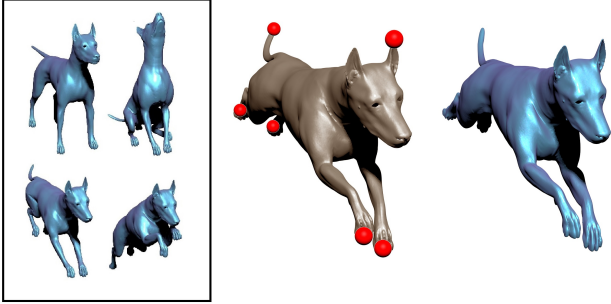

Fig. 4: The four dog shapes are used as examples for our method (left). The deformed shape (right) is found from the vertex positions (middle). In this case the deformed shape is 50% larger than the reference ones.
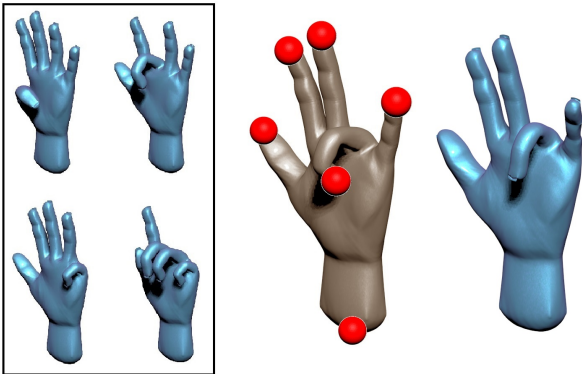

Fig. 5: Example-based deformed shapes from few vertex positions of a hand shape.

Table 1: Model parameters and performance (in milliseconds).

| | Input mesh | | Model | | | Performance | |
class	n	f	r	b	q	1-iter	full
woman	45659	91208	30	178	5	1.3	16.7
centaur	15768	31532	31	166	5	1.2	16.3
wolf	4344	8684	14	82	2	0.7	10.3
dog	25290	50528	25	136	4	1.0	13.9
man	52565	105028	31	235	9	1.9	23.2
cat	27894	55712	22	148	6	1.2	15.3
hand	2224	4424	18	163	7	1.2	15.5
horse	16843	8431	26	139	4	1.1	14.7

5 Experimental results

Table 2: Example-based dictionary. Maximal relative distortion (in percent) of the deformed shapes.

q	dict. size	example-based LBO dict.	example-based skeleton dict.	one example
1	94	2.18	3.59	2.39
2	139	1.97	2.50	2.10
3	184	1.82	2.16	1.95
4	218	1.72	1.89	1.88
5	258	1.63	1.78	1.83
6	293	1.56	1.68	1.77
7	339	1.52	1.60	1.71
8	382	1.51	1.50	1.65

Implementation Considerations. In our implementation we use $m \leq 15$ eigenfunctions as the weighting functions for the example-based LBO dictionary. To support natural articulated shapes deformation, we construct the example-based skeleton dictionary. Its weights are generated using an automatic example-based skinning software package [25]. These skeleton weights are also used to define the rotation clusters. After constructing the dictionary from our mesh examples, we decrease the size of the dictionary using the k-medoids clustering algorithm. This step typically reduces the size of the dictionary in half.

The transformations are found in several steps. We begin by estimating the sparse initial transformations using Equation (23). Typically, we start with $m \leq 4$ eigenfunctions as the weighting functions. Then, we apply a two stage optimization procedure. In the first step we minimize the average ARAP energy of Equation (11). This energy, al-

Fig. 6: Nonrigid ICP. Left to right: The three exemplar shapes, the uncorrupted and complete target shape, the acquired partial shape with eight known feature points (marked as red dots), the initial deformation using the feature points and the final deformation after applying the nonrigid ICP.

though robust, tends to smooth out some of the details of the shape. Therefore, in the second step we optimize the minimal ARAP energy of Equation (12), that effectively selects one example pose which seems to be closest to the target pose. After a few iterations, we apply Equation (24), and change to a richer dictionary that can reflect finer details of the shape. We construct this richer dictionary according to the properties of the subject we want to deform. For articulated shapes, we use the example-based skeleton dictionary. For non-articulated objects, we increase the number of eigenfunctions used to construct the example-based LBO dictionary.

The algorithm was implemented in MATLAB with some optimizations in C++. We use the SVD routines provided by McAdams et al. [33]. All the experiments were executed on a 3.00 GHz Intel Core i7 machine with 32GB RAM. In Table 1 we give the settings for different mesh classes [9,38] and typical performance of the algorithm. For these settings the algorithm takes between 10 and 25 milliseconds.

Example-based dictionary. The example-based dictionary spans natural deformations of a given shape with a small error. In Figure 2 we show some examples of deformations created using the example-based LBO dictionary with 15 eigenfunctions. The mesh parameters and number of example shapes used is as in Table 1. Observe, that there are no noticeable artifacts in these deformations.

We note that the experiments indicate that the accuracy of the proposed model increases with the number of example shapes. For each shape in the database [5], we found the closest deformed shape in the L_2 sense. We calculated the maximal Euclidean distortion between the deformed and the original shape and normalized it by the square root of the original shape's area. Then, we average this maximal distortion for all shapes. We notice, that the maximal distortion decreases as the number of example shapes grows. We also compare the example-based LBO and the skeleton dictionar-

ies. Although quantitatively, the example-based LBO dictionary seems to perform better, our experience suggests that for shapes that have a well-defined skeleton, the example-based skeleton dictionary is more pleasing to the eye, as it captures the stiffness of the bones. Another conclusion is that using many examples improves the deformation accuracy. This can be seen by calculating the distortion of the deformed shape when the example-based LBO dictionary is constructed using one shape only, while keeping the number of dictionary atoms the same and without applying the dictionary reduction step. Table 2 summarizes the results.

Example-based deformation from few vertex positions. Perhaps, the most powerful application of our example-based framework is finding a naturally deformed shape from just a couple of vertex positions. In this scenario, we are given the positions of just a few points of a single depth image of a target shape. Given prior example shapes in different postures, we are able to faithfully and reliably reconstruct the target shape. In Figures 4 and 5, we show reconstructed dog and hand, shapes from a small number of feature points. In these examples, the feature points were sampled in a scale different than that of the example-shapes by a factor of 1.5 (dog), and 0.7 (hand).

Automatic feature point correspondence. The example-based deformation energy can be used to find correspondence between the example shapes and the given feature points [45]. Because our method does not rely on good initialization nor on many input points, it is ideal for such a purpose. For example, in Figure 3, we are given four reference shapes and eight feature points. In this demonstration, the correspondences of the four feature points that belong to each leg (circled in blue) are difficult to find. We can resolve this ambiguity by running our optimization algorithm for all 24 options of permissible correspondences, and calculate the example based energy of Equation (12) for each. Then, the
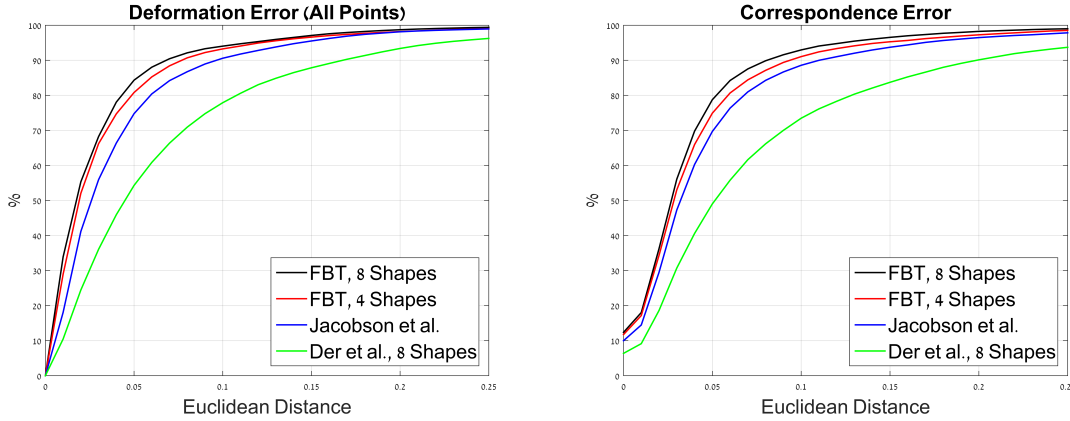
Fig. 7: Evaluation of the shape completion and registration procedure applied to shapes from the TOSCA database.

correspondence can be found by choosing the option that gave the minimal deformation energy.

Shape interpolation. A nice application that can easily be performed, is to interpolate between two deformed shapes. In our setting, we are given two instances of positional constraints. From these constraints we find two deformed shapes and their rotations. Then, we are able to interpolate between these rotations. To produce the new transformations, we apply one additional global-step. Figure 3 demonstrates an interpolation between two deformed shapes of a galloping horse. Four example meshes are used as an input. In the supplementary material we add a video of a galloping horse reconstructed from few feature points. The video frames are interpolated by a factor of eight. Based on the proposed ideas, we developed a computer program that automatically finds a natural deformed shape from a user's specified vertex locations and interpolates between the start pose and the final deformation of the shape, creating a smooth and intuitive motion of the shape. We provide a video that shows how this software is used to make an animation sequence of a moving person.

Nonrigid ICP. The blended transformations can be plugged into a simple nonrigid ICP framework [3, 4, 28]. Nonrigid ICP registration alternates between finding pointwise correspondences and deforming one shape to best fit the other. Hence, we propose the following strategy. To find correspondences compare the vertex positions of all points and their surface normal vectors. In each iteration, we set new linear constraints according to the vertex positions of the obtained point-to-point correspondences, and apply our blended transformations method to wrap the nonrigid shapes while keeping the deformed shape inside the example manifold. We note that because the representation space is defined by the blended transformations it is suffice to match only a subset of points on the two shapes.

Shape completion and registration. In many depth data acquisition scenarios, the acquired data consists of an incomplete, occluded and disconnected parts of a shape. Given some known feature points in those parts of the shape, we want to find the deformation that best fits the partial data and detect the pointwise mapping between acquired partial shape and the reference shapes. To this end, we propose a two step procedure. In the first step, the feature points are used to find an initial deformation. In the second step, the deformation is refined by applying a nonrigid ICP procedure. Since our deformation technique is able to find a good approximation from just a few vertex positions it is ideal to be plugged into this procedure. Figure 6 shows an example of partial data of a cat shape (middle) with some known feature points (marked in red). The initial deformation was found by applying the proposed Fast Blended Transformation algorithm using the known feature points (second from left). The final deformation was attained by applying the nonrigid ICP algorithm in conjunction with our blended transformations approach (left). Notice that the nonrigid ICP algorithm corrected the tilt of the cat's head.

We tested the proposed shape completion and registration procedure on shapes represented by triangulated meshes from the TOSCA database [9]. We performed 50 random experiments with different example and target shapes. For each experiment, we were given eight reference shapes and one target shape for which some of its vertices were removed. We assume that the remaining shape includes some predefined parts that amount to more than 50% of the shape's area. We farther assume that in these parts there are a number of identifiable feature points and that around each feature point, within a certain geodesic circle, no vertices were removed. In the test we performed the number of feature points was set to eight and the radius of the geodesic circle about each point was 15% of the square-root of the shape's area.

We studied the performance of our approach with different number of example shapes. In our implementation,

we set the initial linear constraints to be the weighted average of the vertex positions in 10 different geodesic circles around each feature point. The weights of each vertex were proportional to its voronoi area. Using these linear constraints, we found an initial guess of the deformation. Then, we employed the nonrigid ICP algorithm for the rest of the mesh. We also compared our results with the ones obtained by plugging in the deformation method proposed by Der et al. [11] into our shape completion procedure, using the same skeleton structure. This method applies an example-based deformation gradient model on the problem, and is computationally comparable to the proposed fast blended transformations algorithm. To achieve better results, we used a modified version of the deformation gradient model that supports soft constraints. For leveling the playing field, the automatic skeleton structure was found in the same way for all methods [25].

Figure 7 (left) compares the accuracy of the achieved deformations. The distortion curves describe the percentage of surface points falling within a relative distance from the target mesh. For each shape, the Euclidean distance is normalized by the square root of the shape's area. As for the partial registration, the distortion curves shown in Figure 7 (right) describe the percentage of correspondences that fall within a relative Euclidean distance from what is assumed to be their true locations, similar to the protocol of [22]. We see, that both the deformation quality and the correspondence accuracy increase with the number of reference shapes. This is expected, since as more example poses are introduced, the example-based dictionary better spans the space of natural deformations and we have more poses to compare against. We also notice that for these experiments, our deformation approach (even with one reference shape as in [17]) significantly outperforms the inverse kinematics method of Der et al. [11]. This can be explained by the fact that the reduced deformable model of Der et al. is based on explicit interpolation between the reference poses using deformation gradients. Apparently, this model needs a large number of reference poses to cover all the allowed isometric transformations. In contrast, our model implicitly finds the example manifold by a linear combination of the dictionary atoms and the ARAP energy. Hence, it needs far fewer examples.
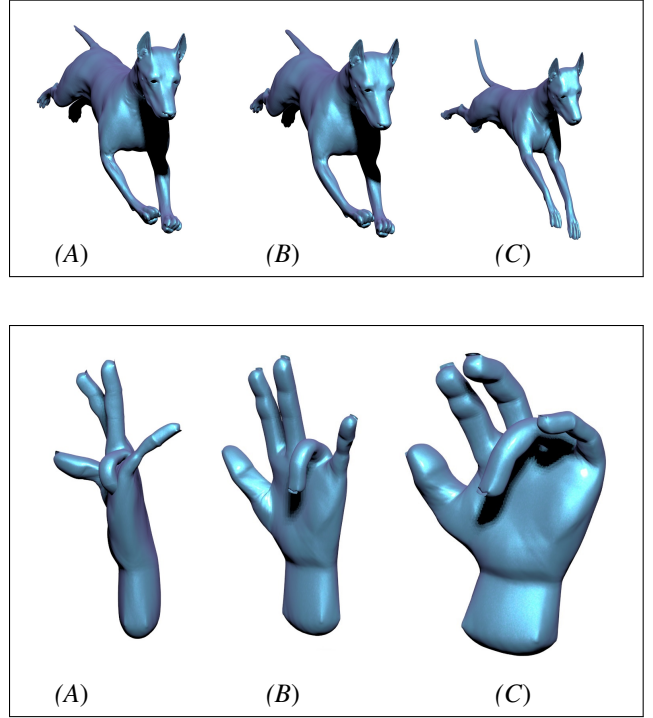

Fig. 8: Deformed shapes constructed by omitting some of the steps in the proposed example-based framework.

(A) Restricting the example-based dictionary to use one example.
(B) Comparing the deformed shape to only one reference shape.
(C) Skipping the scale-step, by setting $\alpha = 1$.

Table 3: List of Mathematical Symbols

Symbol	Description
$\mathcal{M}, \tilde{\mathcal{M}}$	reference and deformed meshes
d	dimension of the shape
n	number of vertices
f	number of faces
m	number of blending weight functions
q	number of example shapes
r	number of rotation clusters
$\mathbf{v}_i^\ell$	ith vertex of the ℓth shape
$w_{j,i}$	jth blending weight function at the ith vertex
ϕ_j	jth eigenfunction of the Laplace-Beltrami operator
λ_j	jth eigenvalue of the Laplace-Beltrami operator
$\mathbf{M}_j^\ell$	jth transformation of the ℓth shape
$\mathbf{V}, \tilde{\mathbf{V}}$	set of reference and deformed vertices
$\mathbf{D}$	example-based dictionary
$\mathbf{T}$	transformation matrix
$\mathbf{R}_k^\ell$	kth rotation of the ℓth shape
$\mathcal{E}_k$	set of vertices of the kth rotation cluster
c_{ijk}	cotangent weight of the edge (i, j) in the kth rotation cluster
$\mathbf{H}$	constraint sampling matrix
$\mathbf{Y}$	constraint matrix
$\boldsymbol{\Lambda}$	diagonal eigenvalue matrix
E_{sm}	smoothness term
E_{lc}	linear constraint term
E_{sc}	scaled as-rigid-as-possible term
E_{av}	averaged as-rigid-as-possible term
β_{lc}	linear constraint weight
β_{sm}	smoothness weight
α_ℓ	scaling parameter of the ℓth shape

6 Discussion

We tested some deficient versions of our example-based deformation framework. In Figure 8 we show several examples of how these partial versions of the algorithm behave. For comparison to the complete method see Figures 4 and 5. We notice that the most important part of the proposed framework is the construction of the dictionary from multiple examples. If only one example is used *(A)*, as in [17], the deformation algorithm fails when the shape has many degrees of freedom.

Although the method is robust and usually performs very well, some limitations and failures in particular cases do exist. Despite the usually pleasing to the eye deformations of the proposed example-based approach, sometimes undesirable artifacts might occur. This is the result of the collinearity between different dictionary atoms. As for the performance of the algorithm, the deformation can be produced in real time but the algorithm cannot accommodate for video applications with many objects that need to be simultaneously deformed. This problem can be solved by using the proposed algorithm only for objects for which a previous pose cannot be used for the initialization of the current one. Another drawback is that if the example shapes do not incorporate enough information for extracting the right rotation clusters, then, the algorithm will ultimately fail. Also, the current evaluation system does not prevent self-intersections.

7 Conclusions

We applied the concept of overcomplete dictionary representation to the problem of shape deformation. The proposed example-based deformation approach extends the subspace of physically-plausible deformations, while controlling the smoothness of the reconstructed mesh. The blended transformations enable us to find a new pose from a small number of known feature points without any additional information. It is well-suited for real-time applications as well as offline animation and analysis systems. In the future, we plan to apply the proposed framework to various problems from the field of shape understanding, such as gesture recognition, registration of MRI images, and prior based object reconstruction from depth images.

References

1. M. Alexa. Differential coordinates for local mesh morphing and deformation. *The Visual Computer*, 19(2):105–114, 2003.
2. M. Alexa, D. Cohen-Or, and D. Levin. As-rigid-as-possible shape interpolation. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, pages 157–164, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

3. B. Allen, B. Curless, and Z. Popović. The space of human body shapes: Reconstruction and parameterization from range scans. *ACM Trans. Graph.*, 22(3):587–594, July 2003.

4. B. Amberg, S. Romdhani, and T. Vetter. Optimal step nonrigid icp algorithms for surface registration. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–8, June 2007.

5. D. Anguelov, P. Srinivasan, H. cheung Pang, D. Koller, S. Thrun, and J. Davis. The correlated correspondence algorithm for unsupervised registration of nonrigid surfaces. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 33–40. MIT Press, 2005.

6. F. Bogo, J. Romero, M. Loper, and M. J. Black. Faust: Dataset and evaluation for 3d mesh registration. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

7. S. Bouaziz, S. Martin, T. Liu, L. Kavan, and M. Pauly. Projective dynamics: Fusing constraint projections for fast simulation. *ACM Trans. Graph.*, 33(4):154:1–154:11, July 2014.

8. S. Bouaziz, Y. Wang, and M. Pauly. Online modeling for real-time facial animation. *ACM Transactions on Graphics (TOG)*, 32(4):40, 2013.

9. A. M. Bronstein, M. M. Bronstein, and R. Kimmel. *Numerical geometry of non-rigid shapes*. Springer, 2008.

10. I. Chao, U. Pinkall, P. Sanan, and P. Schröder. A simple geometric model for elastic deformations. In *ACM SIGGRAPH 2010 Papers*, SIGGRAPH '10, pages 38:1–38:6, New York, NY, USA, 2010. ACM.

11. K. G. Der, R. W. Sumner, and J. Popović. Inverse kinematics for reduced deformable models. *ACM Trans. Graph.*, 25(3):1174–1179, July 2006.

12. T. K. Dey, P. Ranjan, and Y. Wang. Eigen deformation of 3d models. *The Visual Computer*, 28(6-8):585–595, 2012.

13. M. Elad. *Sparse and redundant representations: From theory to applications in signal and image processing*. Springer Science & Business Media, 2010.

14. W.-W. Feng, B.-U. Kim, and Y. Yu. Real-time data driven deformation using kernel canonical correlation analysis. In *ACM SIGGRAPH 2008 Papers*, SIGGRAPH '08, pages 91:1–91:9, New York, NY, USA, 2008. ACM.

15. S. Frhlich and M. Botsch. Example-driven deformations based on discrete shells. *Computer Graphics Forum*, 30(8):2246–2257, 2011.

16. E. Grinspun, A. N. Hirani, M. Desbrun, and P. Schröder. Discrete shells. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '03, pages 62–67, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.

17. A. Jacobson, I. Baran, L. Kavan, J. Popović, and O. Sorkine. Fast automatic skinning transformations. *ACM Trans. Graph.*, 31(4):77:1–77:10, July 2012.

18. A. Jacobson, I. Baran, J. Popović, and O. Sorkine-Hornung. Bounded biharmonic weights for real-time deformation. *Commun. ACM*, 57(4):99–106, Apr. 2014.

19. D. L. James and C. D. Twigg. Skinning mesh animations. In *ACM SIGGRAPH 2005 Papers*, SIGGRAPH '05, pages 399–407, New York, NY, USA, 2005. ACM.

20. L. Kaufman and P. Rousseeuw. *Clustering by means of medoids*. North-Holland, 1987.

21. L. Kavan, P.-P. Sloan, and C. O'Sullivan. Fast and efficient skinning of animated meshes. *Computer Graphics Forum*, 29(2):327–336, 2010.

22. V. G. Kim, Y. Lipman, and T. Funkhouser. Blended intrinsic maps. In *ACM SIGGRAPH 2011 Papers*, SIGGRAPH '11, pages 79:1–79:12, New York, NY, USA, 2011. ACM.

23. Y. Koyama, K. Takayama, N. Umetani, and T. Igarashi. Real-time example-based elastic deformation. In *Proceedings of the 11th ACM SIGGRAPH / Eurographics Conference on Computer Animation*, EUROSCA'12, pages 19–24, Aire-la-Ville, Switzerland, Switzerland, 2012. Eurographics Association.

24. P. G. Kry, D. L. James, and D. K. Pai. Eigenskin: Real time large deformation character skinning in hardware. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '02, pages 153–159, New York, NY, USA, 2002. ACM.

25. B. H. Le and Z. Deng. Robust and accurate skeletal rigging from mesh sequences. *ACM Trans. Graph.*, 33(4):84:1–84:10, July 2014.

26. Z. Levi and C. Gotsman. Smooth rotation enhanced as-rigid-as-possible mesh animation. *IEEE transactions on visualization and computer graphics*, 21(2):264–277, 2015.

27. J. P. Lewis, M. Cordner, and N. Fong. Pose space deformation: A unified approach to shape interpolation and skeleton-driven deformation. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, pages 165–172, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

28. H. Li, R. W. Sumner, and M. Pauly. Global correspondence optimization for non-rigid registration of depth scans. *Computer Graphics Forum*, 27(5):1421–1430, 2008.

29. Y. Lipman, O. Sorkine, D. Cohen-Or, D. Levin, C. Rossi, and H. P. Seidel. Differential coordinates for interactive mesh editing. In *Shape Modeling Applications, 2004. Proceedings*, pages 181–190, June 2004.

30. L. Liu, L. Zhang, Y. Xu, C. Gotsman, and S. J. Gortler. A local/global approach to mesh parameterization. *Computer Graphics Forum*, 27(5):1495–1504, 2008.

31. N. Magnenat-thalmann, R. Laperrire, D. Thalmann, and U. D. Montral. Joint-dependent local deformations for hand animation and object grasping. In *In Proceedings on Graphics interface 88*, pages 26–33, 1988.

32. S. Martin, B. Thomaszewski, E. Grinspun, and M. Gross. Example-based elastic materials. *ACM Trans. Graph.*, 30(4):72:1–72:8, July 2011.

33. A. McAdams, A. Selle, R. Tamstorf, J. Teran, and E. Sifakis. Computing the singular value decomposition of 3×3 matrices with minimal branching and elementary floating point operations. Technical report, Technical Report, University of Wisconsin-Madison, 2011.

34. U. Pinkall and K. Polthier. Computing discrete minimal surfaces and their conjugates. *Experimental Mathematics*, 2(1):15–36, 1993.

35. C. Schumacher, B. Thomaszewski, S. Coros, S. Martin, R. Sumner, and M. Gross. Efficient simulation of example-based materials. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 1–8. Eurographics Association, 2012.

36. P.-P. J. Sloan, C. F. Rose, III, and M. F. Cohen. Shape by example. In *Proceedings of the 2001 Symposium on Interactive 3D Graphics*, I3D '01, pages 135–143, New York, NY, USA, 2001. ACM.

37. O. Sorkine and M. Alexa. As-rigid-as-possible surface modeling. In *Proceedings of EUROGRAPHICS/ACM SIGGRAPH Symposium on Geometry Processing*, pages 109–116, 2007.

38. R. W. Sumner and J. Popović. Deformation transfer for triangle meshes. *ACM Trans. Graph.*, 23(3):399–405, Aug. 2004.

39. R. W. Sumner, M. Zwicker, C. Gotsman, and J. Popović. Mesh-based inverse kinematics. In *ACM SIGGRAPH 2005 Papers*, SIGGRAPH '05, pages 488–495, New York, NY, USA, 2005. ACM.

40. C. Von-Tycowicz, C. Schulz, H.-P. Seidel, and K. Hildebrandt. Real-time nonlinear shape interpolation. *ACM Trans. Graph.*, 34(3):34:1–34:10, May 2015.

41. R. Y. Wang, K. Pulli, and J. Popović. Real-time enveloping with rotational regression. *ACM Trans. Graph.*, 26(3), July 2007.

42. Y. Wang, A. Jacobson, J. Barbič, and L. Kavan. Linear subspace design for real-time shape deformation. *ACM Transactions on Graphics (TOG)*, 34(4):57, 2015.

43. T. Winkler, J. Drieseberg, M. Alexa, and K. Hormann. Multi-scale geometry interpolation. *Computer Graphics Forum*, 29(2):309–318, 2010.

44. D. Xu, H. Zhang, Q. Wang, and H. Bao. Poisson shape interpolation. *Graphical Models*, 68(3):268–281, 2006.

45. H. Zhang, A. Sheffer, D. Cohen-Or, Q. Zhou, O. Van Kaick, and A. Tagliasacchi. Deformation-driven shape correspondence. *Computer Graphics Forum*, 27(5):1431–1439, 2008.

46. W. Zhang, J. Zheng, and N. M. Thalmann. Real-time subspace integration for example-based elastic material. In *Computer Graphics Forum*, volume 34, pages 395–404. Wiley Online Library, 2015.

47. H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.