Computer Aided Geometric Design 35–36 (2015) 82–94

Contents lists available at ScienceDirect

Computer Aided Geometric Design

www.elsevier.com/locate/cagd

Spectral pose transfer

Mengxiao Yin [a,b], Guiqing Li [a,*], Huina Lu [a], Yaobin Ouyang [a], Zhibang Zhang [a], Chuhua Xian [a]

[a] School of Computer Science and Engineering, South China University of Technology, Guangzhou, China
[b] School of Computer, Electronics and Information, Guangxi University, Nanning, China

ARTICLE INFO

Article history:
Received 20 February 2015
Accepted 16 March 2015
Available online 25 March 2015

Keywords:
Pose transfer
Spectral decomposition
Medium-scale pose
Coupled quasi-harmonic bases

ABSTRACT

In spectral decomposition of a 3D mesh model, it is well known that eigenvectors with respect to small eigenvalues determine its main pose while eigenvectors associated with large eigenvalues encode its surface details. Based on this property, given two meshes with different connectivities, one can use coupled quasi-harmonic technique to transfer the pose of one mesh onto the other by exchanging the low-frequency coefficients in their spectral representations. However, directly synthesizing the new low frequencies with old high frequencies usually exhibits two vital artifacts: one is detail shearing and shape collapsing, and the other is medium-scale pose missing. This paper reformulates the pose transfer as a deformation problem with low-frequency coefficients as handles. It finally leads to a non-linear optimization with the coefficients as data constraint and Laplacian coordinates as regularity term for preserving details. Meanwhile, a hierarchical pose transfer framework is introduced to capture the medium-frequency poses. To reduce the computation complexity and enhance the stability we further solve the problem in a subspace defined by mean-value coordinates.

1. Introduction

3D mesh editing has been exhaustedly investigated in digital geometry processing (Gain and Bechmann, 2008; Xu and Zhou, 2009). Example-based mesh modeling is a commonly used strategy in this field. Deformation transfer, as an example-based deformation method, is an analogue in which two poses of a reference model are given and therefore the underlying motion of the two poses of the model can be extracted as a geometric transformation which can drive a source model deforming in the same manner (Baran et al., 2009; Sumner and Popović, 2004). In this case, the source model is usually required to have a similar pose with the first pose of the reference one.

However, sometimes we may encounter a more succinct problem: given two mesh models, named reference and source models respectively for example, which have different poses and usually possess different connectivities, how can we make the source one deform to a pose similar to that of the reference model? This problem is referred to as a pose transfer in the literature (Kovnatsky et al., 2013; Lévy, 2006). Since neither deforming course of the reference model nor pose registration between source and reference meshes is required, pose transfer can greatly expand the scope of application of example-based mesh manipulation compared to deformation transfer. From the viewpoint of human, it is not difficult to

* Corresponding author.
E-mail addresses: 42045234@qq.com (M. Yin), ligq@scut.edu.cn (G. Li), luhuina2009@163.com (H. Lu), 549088764@qq.com (Y. Ouyang), cheungbon@163.com (Z. Zhang), chhxian@scut.edu.cn (C. Xian).

http://dx.doi.org/10.1016/j.cagd.2015.03.016
0167-8396/

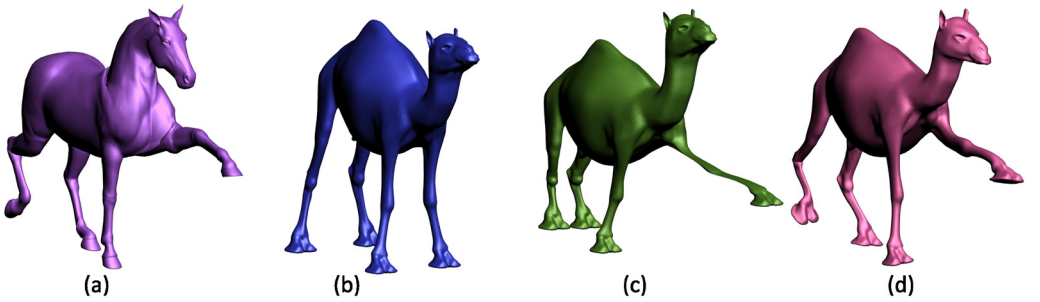

Fig. 1. Naively transferring low-frequency coefficients fails to correctly capture the reference pose and source details: (a) and (b) are respectively reference and source meshes; (c) result by copying coefficients with bases optimization (Kovnatsky et al., 2013); (d) our pose transfer.

imitate all kinds of poses of other people or even creatures even if we do not know how they move to such pose. It is a challenge to automatically conduct a natural pose transfer in mesh editing, however, due to the difficulty in describing a pose. To our knowledge, no deformation process has been proposed to sophisticatedly tackle this issue.

Fortunately, the spectral analysis technique over a 3D mesh exposes that the low-frequency components of the Laplace matrix of a mesh usually encode its pose information. Once the mesh is decomposed as the linear combination of eigenvectors of the matrix, altering the coefficients of the low-frequency components can change the pose of the model (Dey et al., 2012). The coupled quasi-harmonic base technique described in Kovnatsky et al. (2013) first sheds light on transferring poses between two mesh models with different connectivities by replacing the low-frequency components of a mesh with that of the other one. However, as it is impossible to completely decouple pose and shape of a model, this method not only fails to reproduce the reference pose in all scales but also loses geometric details of the target model in general. We present an example to show this in Fig. 1 in which the first two images (Fig. 1(a)–(b)) are respectively reference and source poses. The approach by Kovnatsky et al. (2013) exhibits serious artifacts of distortion and details missing as shown in Fig. 1(c). Fig. 1(d) depicts our result which is better than Fig. 1(c) both in keeping details and in capturing multi-level poses.

Spectral analysis based on Laplace operator has achieved great success as a digital geometry processing tool in 3D modeling applications such as compression (Karni and Gotsman, 2000), quadrangulation (Huang et al., 2008), deformation (Dey et al., 2012; Rong et al., 2008), and shape analysis (Song et al., 2014) to name a few, due to its harmonic and multi-scale properties. One can find more applications in excellent surveys (Lévy and Zhang, 2010; Zhang et al., 2007). We investigate pose transfer in this paper by unifying the advantage of the frequency components of Laplace matrix in coding the rough pose, and the ability of Laplace coordinates in representing details into a framework. Specifically, we extract the low-frequency components of the reference model to drive the source model deforming based on the coupled quasi-harmonic bases and prevent details missing by preserving Laplacian coordinates of the source model up to a rotation during deformation. In addition, we organize the reference pose into a simple hierarchical structure and apply the same pose transfer strategy to learn medium-scale poses. Our main contributions are listed as follows:

- We establish a new Laplace editing framework which regards the frequency components as handles to address the pose transfer issue. It results in a non-linear optimization.
- Observing that solution of the above optimization leads to a dense linear system, we integrate the cage-based subspace technique into the framework. It is not only able to cut down the solution space greatly but also guarantee the stability of the solutions by reducing deforming degree of freedoms.
- Revealing the multi-scale property of the pose of a 3D model, we devise a simple hierarchical strategy to transfer the different-scale poses of the reference model onto the source one progressively.

2. Related work

Considering that 3D pose transfer can be regarded as a special mesh deformation/editing operation, we first recall some classical deformation methods in the beginning of this section. Immediately following, we mention some mesh manipulating methods based on subspace solution. Deformation transfer is semantically most close to pose transfer among all kinds of 3D mesh manipulations. Functional maps are the foundation of the proposed method. We also make a brief survey on these two fields. Finally, we sketch some key ingredients upon which our framework is built.

Mesh deformation Given new positions or transformations of a subset of vertices on a 3D mesh as constraints, the metaphor of deformation (Gain and Bechmann, 2008) is to create a deformed mesh which not only satisfies the constraints but also preserves some plausible geometric or kinetic characteristics. Here, we only cite several simple examples: the Laplacian mesh editing preserves geometric details in terms of Laplacian coordinates (Lipman et al., 2004; Sorkine et al., 2004; Zhang et al., 2011; Zhou et al., 2005); the ARAP deformation enforces the motion of each triangle of the mesh approximating rigidity so as to retain the triangle shape best (Igarashi et al., 2008; Sorkine and Alexa, 2007; Sumner et al., 2007); the example-driven deformation by Fröhlich and Botsch minimizes the change of edge length, dihedral angle and mesh volume during deformation (Fröhlich and Botsch, 2011).

Spectral-based deformation (Rong et al., 2008) minimizes the change of the first and second fundamental forms, which is actually a simplified elastic deformable model. To reduce computational complexity, the authors decomposed the mesh model into two layers, i.e., a smoothed eigen-skeleton and details and found the deformed eigen-skeleton in the frequency space spanned by low-frequency eigenvectors of the original model. The details are then added back to the deformed smoothed model using deformation transfer techniques. The eigen-deformation (Dey et al., 2012) follows a similar framework, however an initial deformed eigen-skeleton is computed by rigidly transforming the specified region, and then approximated by a smoothed (deformed) one in the low-frequency space.

Subspace deformation There are three types of subspace methods for 3D deformation. The essence of these methods is driving the 3D model using a simple geometric agent such as cages, simplified meshes, and skeletons. In an interactive system, one can change the 3D model by manually updating the agent. Barycentric coordinates editing generally represents a 3D mesh as the combination of the vertices of its cage and deforms it by moving the cage vertices (Ju et al., 2005; Li et al., 2013; Lipman et al., 2008). Cages are usually used as geometric agents of meshes to be manipulated. In this case, cages are optimized for driving the meshes deforming. As a cage generally has far fewer vertices than its corresponding mesh model the complexity of the solution space often decreases greatly (Huang et al., 2006). Solving deformation problem in a hierarchical manner (Kircher and Garland, 2005; Kilian et al., 2007; Winkler et al., 2010) can also improve efficiency and enhance robustness, however, it is usually more complex in implementation. Skeletal skinning-based deformation drives the mesh to change in a near rigid manner by estimating the transformation of mesh vertices as the linear combination of the transformations of bones (Capell et al., 2002; Weber et al., 2007).

Deformation transfer Sumner and Popović (2004) copied the deformation suggested by two given reference meshes with different shapes/poses onto a source mesh which usually bears connectivity different from the reference ones. They represented the deformation between the two reference meshes as a set of transformations, and then mapped these transformations onto the source one to obtain a target (deformed) mesh via establishing a correspondence between the reference and source meshes. Baran et al. further considered semantic properties (Baran et al., 2009) during transferring. Zhou et al. addressed the deformation transfer between objects with unconnected components (Zhou et al., 2010). By deforming the space in which the reference model is embedded, Ben-Chen et al. enabled the deformation transfer from reference poses to a source model with different representations (Ben-Chen et al., 2009). Cage-based deformation transfer first solves a cage deformation using the reference meshes and then drives the source mesh deforming based on barycentric coordinate editing (Chen et al., 2010). It should be noticed that all aforementioned methods require multiple reference poses as input for defining the deformation of the reference model.

Functional maps To investigate the correspondence between two mesh models, Ovsjanikov et al. introduced linear functional maps between two function spaces defined on the two meshes (Ovsjanikov et al., 2012). In a continuous viewpoint, a correspondence between functions defined on two shapes uniquely determines a matching between points on the two shapes. As square-integrable functions can be represented as linear combinations of eigenfunctions of the Laplacian operators on the two shapes, the compatibility of two groups of eigenfunctions is vital for many applications (Huang et al., 2014; Lévy, 2006). Completely consistent behaving of two groups of eigenfunctions can only be guaranteed for a very restrictive shape pair, namely, isometric shapes with simple spectrum (i.e., multiplicity of eigenvalues equal to one) (Ovsjanikov et al., 2012).

Pose transfer Compared to deformation transfer, pose transfer takes use of less information to modify the shape of a 3D mesh model. Only one reference mesh is given in the scenarios to specify the target pose from which a given source mesh should learn. Lévy revealed that if projecting the shape space of a mesh, namely the position vector consisting of all vertices of the mesh, onto the harmonic bases of its Laplacian matrix, the coefficients corresponding to low-frequency bases can reflect its pose. The author performed a rough pose transfer between two meshes with the same connectivity by exchanging their low-frequency coefficients. Kovnatsky et al. proposed the notion of coupled quasi-harmonic bases (Kovnatsky et al., 2013) to construct compatible eigen-bases based on functional maps for two mesh shapes with different connectivities and poses. The authors showed the possibility to perform pose transfer by exchanging low-frequency coefficients of two meshes. In both methods, the naïve coefficient exchanging operation tends to impair the geometric details of the original source model. Furthermore, the transferred results are not able to capture the medium-scale poses of the reference shape too.

Our pose transfer framework also employs coupled quasi-harmonic bases to code the global pose, however, we force the whole model undergoing a global deformation under the guidance of low-frequency coefficients with Laplacian coordinates to preserve geometric features. As low-frequency coefficients only code the global pose of an object, only copying them cannot transfer the poses of different scales of the reference model to the source model. As shown in Fig. 1(a), the pose of the left-rear leg of the horse can be regarded as global while the deformation of the head and the right-rear leg of the horse is relatively small and their poses can be viewed as medium-size ones. We can observe in Fig. 1(c) that the low-frequency coefficients of the horse can only drive the camel model deformed according to the global pose. To address the issue, we cut out the parts of the reference model with medium-size pose and the corresponding components of the source one. In each component pair, their original medium-size poses now become global ones and we can perform pose transfer based on our above formulation of global deformation. This process can be conducted in a nested manner if necessary. After all parts from the source model are completely deformed under the guidance of the reference model, we glue them together to form the final result.

3. Preliminary

Before describing the proposed method in detail, we first introduce some terminologies and notations. Given a triangular mesh $M = < V, E, F >$, where $V = \{v_0, v_1, \ldots, v_{|V|-1}\}$ is the vertex set and E, F are respectively the edge and face sets.

Laplacian matrix Laplacian matrix of M is defined as $L = D^{-1}W$ (Kovnatsky et al., 2013), where $D = \mathrm{diag}\{1/a_0, 1/a_1, \ldots, 1/a_{|V|-1}\}$ with a_i the Voronoi area of vertex i, and W is a symmetric matrix with cotangent weights as its entries:

$$w_{ij} = \begin{cases} \frac{1}{2}(\cot\alpha_{ij} + \cot\beta_{ij}), & i \neq j \\ -\sum_{k\neq i} w_{ik}, & i = j, \end{cases} \tag{1}$$

with α_{ij} and β_{ij}, respectively, the opposite angles of edge (i, j) with respect to two triangles sharing the edge. Suppose $L\phi_i = \lambda_i\phi_i$, $i = 0, \cdots, |V| - 1$, where λ_i and ϕ_i are respectively eigenvalues and eigenvectors of L. We say $\Phi = \{\phi_i\}$ $(i = 0, \cdots, |V| - 1)$ the harmonic bases of M. Notice that Eq. (1) may result in non-positive weights for triangular meshes of bad quality (Bobenko and Springborn, 2007). Refining them into intrinsic Delaunay triangulations can remedy the artifact as described in Fisher et al. (2006) and Liu et al. (2014). We can employ these methods to calculate the Laplacian weights if necessary.

Functional maps Given two mesh shapes $M = < V, E, F >$ and $M' = < V', E', F' >$ which are usually triangular meshes with different connectivities, bijective map $T : M \to M'$ induces a functional map (Ovsjanikov et al., 2012) $\mathcal{F} : \mathcal{L}^2(M) \to \mathcal{L}^2(M')$ with $g(v') = \mathcal{F}(f)(v') = f(T^{-1}(v'))$. Denote $\Phi = \{\phi_i\}$ $(i = 0, \cdots, |V| - 1)$ and $\Phi' = \{\phi'_i\}$ $(i = 0, \cdots, |V'| - 1)$ the two groups of harmonic bases corresponding to M and M', respectively. In the rest of this paper, we always assume that harmonic bases have been sorted according to their eigenvalues in an increasing order. Suppose

$$\mathcal{F}(\phi_i) = \sum_{j=0,\ldots,|V'|-1} c_{ij}\phi'_j, \ i = 0, \cdots, |V| - 1, \tag{2}$$

where $\Gamma = (c_{ij})_{|V|\times|V'|}$ is called a functional matrix.

Pose transfer Using the above notations, we can expand the vertex vectors of two meshes as follows:

$$V = \sum_{i=0}^{|V|-1} \alpha_i\phi_i, \ V' = \sum_{i=0}^{|V'|-1} \alpha'_i\phi'_i. \tag{3}$$

Noticing that α_i with respect to small eigenvalues, as low-frequency coefficients, usually determine the whole pose of a model, Lévy (2006) found that exchanging low-frequency coefficients of two expansions in Eq. (3) can transfer the pose of one mesh to the other, provided that α_i and α'_i are expressed in the same "language". For example, replacing V with

$$\tilde{V} = \sum_{i=0}^{K} \alpha'_i\phi_i + \sum_{i=K+1}^{|V|-1} \alpha_i\phi_i \tag{4}$$

can transfer the pose of M' to M. Considering that the two groups of harmonic bases are generally incompatible, Kovnatsky et al. (2013) introduced coupled quasi-harmonic bases which are consistent in the sense that functional matrix Γ is close to diagonal with positive entries.

4. Spectrum-driven deformation

Given source and reference shapes M and M' which can possess different connectivities, pose transfer requires M not only imitating the pose of M' but also preserving the details of M, in an as-rigid-as-possible manner, for example. We found via experiments that neither Lévy (2006) nor Kovnatsky et al. (2013) can achieve a true pose transfer in this sense. Actually, Eq. (4) used by both methods only copies low-frequency coefficients of M' to M while the other frequency coefficients remain unchanged the therefore the details is added with any geometric transformations. This inevitably exhibits artifacts of local pose missing and detail shearing.

We cast pose transfer to a deformation problem with low-frequency coefficients as handles. We firstly establish a formulation to reconstruct the deformed shape of M combining the global pose of M' and detail features of M together. To reduce the computational complexity, we further introduce the subspace technique based on barycentric coordinates. To address the local pose issue, we then discuss a segmentation-based strategy to convert local pose transfer into global pose transfer.

4.1. Spectral reconstruction constrained with Laplacian coordinates

Using the notations in Section 3, we assume that Φ and Φ' have been optimized to be coupled quasi-harmonic and expansions in Eq. (3) have also been performed. We investigate how to combine low-frequency coefficients $\{\alpha'_i\}$ $(i = 0, 1, \ldots, K)$ from M' and Laplacian coordinates $\delta = LV = (\delta_0, \delta_1, \ldots, \delta_{|V|-1})^T$ of M to reconstruct a target pose $\tilde{M} = < \tilde{V}, \tilde{E}, \tilde{F} >$ of M, where $|\tilde{V}| = |V|$, $\tilde{E} = E$, and $\tilde{F} = F$. The first condition indicates that the projection of vertex

vector $\tilde{V}$ onto ϕ_i $(0 \le i \le K)$ should be $(\tilde{V}, \phi_i) = \alpha_i'$, $i = 0, 1, \ldots, K$, where $(,)$ denotes inner product. This leads to a low-frequency energy

$$\mathcal{E}_{LF} = \sum_{i=0}^{K} \|\tilde{V}^T \phi_i - \alpha_i'\|^2. \tag{5}$$

On the other hand, to preserve the Laplacian coordinates we set the following Laplacian energy (Lipman et al., 2004; Sorkine et al., 2004)

$$\mathcal{E}_{LC} = \sum_{i=0}^{|V|-1} \|R_i(\tilde{V})\delta_i - \tilde{\delta}_i\|^2, \tag{6}$$

where $\tilde{\delta}_i$ are Laplacian coordinates on $\tilde{M}$, and R_i are matrices rotating δ_i according to the orientation of the new pose. This finally leads to our pose transfer energy

$$\mathcal{E}_{PT} = \mathcal{E}_{LF} + \lambda \mathcal{E}_{LC}. \tag{7}$$

4.2. Formulation in cage subspaces

Different from the ordinary Laplacian surface editing described in Lipman et al. (2004), Sorkine et al. (2004), minimizing $\mathcal{E}_{PT}$ in Eq. (7) will result in a dense linear system since energy item $\mathcal{E}_{LF}$ defined in Eq. (5) is a global constraint. To reduce the computational complexity, we employ the subspace method. Let C be a cage of M, our goal is to find a new pose of C denoted by $\tilde{C}$, such that $\tilde{M}$ reconstructed from $\tilde{C}$ based on barycentric coordinates editing can mimic the low-frequency pose of M' well. In the following, we also use the same notation to denote the vertex vector of a cage.

Let Ψ be the barycentric coordinates matrix of mesh M with respect to C, namely, we have $V = \Psi C$. Once $\tilde{C}$ is found, we can compute the vertex vector of $\tilde{M}$ as $\tilde{V} = \Psi\tilde{C}$ (Ju et al., 2005). Substituting $\tilde{V}^T$ for $\tilde{C}^T\Psi^T$ in Eqs. (5) and (6) leads to

$$\mathcal{E}_{LF}^M = \sum_{i=0}^{K} \|\tilde{C}^T\Psi^T\phi_i - \alpha_i'\|^2 \tag{8}$$

and

$$\mathcal{E}_{LC}^M = \sum_{i=0}^{|V|-1} \|R_i(\tilde{C}^T\Psi^T)\delta_i - \tilde{\delta}_i(\tilde{C}^T\Psi^T)\|^2, \tag{9}$$

respectively, where superscript M indicates the constraint is performed on vertices of the source mesh M. In our experiments, we always set $K = 5$.

Minimization of the combined energy of Eqs. (8) and (9) poses over-constraint on the cage and easily yields distorted results. We therefore suppress the distortion of cage $\tilde{C}$ by substituting Laplacian coordinates of C for Laplacian coordinates preserving item of M in Eq. (9):

$$\mathcal{E}_{LC}^C = \sum_{i=0}^{|C|-1} \|R_i(\tilde{C})\delta_i^C - \tilde{\delta}_i^{\tilde{C}}\|^2, \tag{10}$$

where δ_i^C and $\tilde{\delta}_i^{\tilde{C}}$ are respectively Laplacian coordinates of C and $\tilde{C}$.

We adapt the algorithm by Sorkine et al. (2004) to solve the minimization of the combined energy of Eq. (8) and Eq. (10). According to Sorkine et al. (2004), entries of both $R_i(\tilde{C}^T\Psi^T)$ and $R_i(\tilde{C})$ can be expressed in terms of vertices of $\tilde{C}$ (see Appendix A for more details). We find by experiments that directly applying their approach to our setting usually exhibits non-uniform scaling artifact in different regions of the model to be edited as shown in Fig. 2 in which the reference, source and result are shown from left to right. We therefore introduce a penalty which explicitly constrains the scaling factor $s_i(\tilde{C})$ to 1:

$$\mathcal{E}_{SF}^C = \sum_{i=0}^{|\tilde{C}|-1} \|s_i(\tilde{C}) - 1\|^2. \tag{11}$$

Notice that $s_i(\tilde{C})$ can be expressed as the function of vertices in $\tilde{C}$ (see Appendix A).

Since $\tilde{V} = \Psi\tilde{C}$, we constrained the cage Laplacian coordinates as regularity term $\mathcal{E}_{LC}^C$, this leads to our final energy

$$\mathcal{E}_{PT}^C = \mathcal{E}_{LF}^M + \lambda_1\mathcal{E}_{LC}^C + \lambda_2\mathcal{E}_{SF}^C, \tag{12}$$

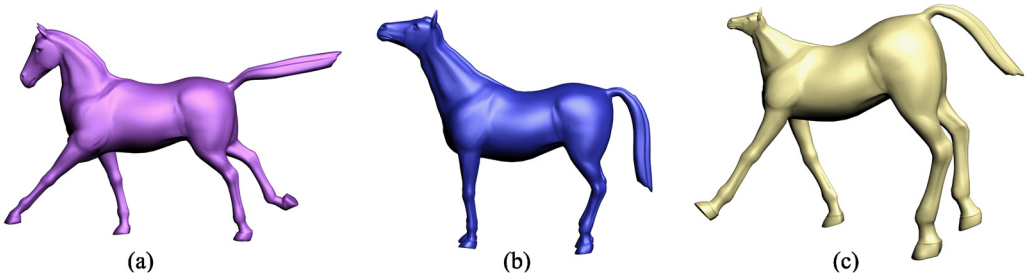

Fig. 2. Directly applying the original algorithm of Sorkine et al. (2004) to our scenarios tends to non-uniformly scale the source model: reference, source and resulting meshes are shown from left to right.

where λ_1 and λ_2 are weights and all energies are functions of vertices of unknown cage $\tilde{C}$. A brief explanation on how to minimize Eq. (12) is presented in Appendix B. In our current implementation, weights λ_1 and λ_2 are manually chosen in order to make trade-off between pose learning and details preserving. For example, λ_1 and λ_2 are set to 0.05 in Fig. 3 while $\lambda_1 = \lambda_2 = 5 \times 10^{-4}$ in Fig. 9. If the details are not distorted obviously, we increase the two weights. Conversely, if the poses are not learned completely, we have to decrease the weights.

4.3. Hierarchical pose transfer

In spectral pose transfer, only low-frequency coefficients of the source model are conveyed to the target one, the secondary pose change of the source model cannot be captured by the target one. To address the issue, we interactively segment the corresponding components from two models whose poses are not identical by using easy mesh cutting (Ji et al., 2006). These medium-scale poses of the original model are then converted into low-frequency ones. It indicates that we can perform the pose transfer scheme described in Section 4.2 on these pairs of components. If necessary, we can also further decompose these local components into smaller parts in order to achieve complete pose learning. Namely, this is a multi-level or hierarchical process.

More specifically, suppose we have obtained new pose $\tilde{M}$ of M by learning low frequency of pose M'. Let $\tilde{P}$ and P' be two corresponding parts of $\tilde{M}$ and M', respectively. We enforce $\tilde{P}$ deformed to $\tilde{\tilde{P}}$ by learning pose P'.

It should be noticed that the segmented parts are usually too coarse to obtain usable coupled quasi-harmonic bases. We employ the butterfly subdivision (Dyn, 1990) to refine them once or twice depending on their vertex numbers. After successfully transferring poses between the reference and source parts, we remove the inserted vertices such that the new pose has the same connectivity as the original one.

After multi-level transfer, each source part (for convenience, we call the part submesh) has been deformed to the similar pose with the reference one, but sometimes, it may be different from the reference counterpart in orientation. In order to achieve good effect, we use a simple strategy to register the components from two consecutive levels. First, we obtain a local frame for both the source part and the reference counterpart. Both local frames are constructed by three points, namely, the average point of the whole submesh, the average point of the cut boundary vertices and a point that is far away from the boundary. After that, we calculate the rotation matrix between the two local frames, and then, apply the transformation to all vertices of the deformed source submesh. Finally, we smooth the boundary area by keeping the Laplacian coordinates of boundary vertices of the deformed source submesh before rotation and the Laplacian coordinates of inner vertices of the deformed source submesh after rotation.

5. Implementation and results

We have implemented the proposed framework with Visual Studio 2010. All results are obtained by a notebook computer with Intel Core i7-3520M 2.90GHz CPU, 8.00GB RAM memory. We call the Matlab code provided by Kovnatsky et al. (2013) in our C++ program to compute the coupled quasi-harmonic bases. In Kovnatsky et al. (2013), the given p corresponding functions $g_i = f_i \circ t^{-1}$ are represented in the discrete setting by matrices $\mathbf{F} = (\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_p)$ and $\mathbf{G} = (\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_p)$ of size $|V| \times p$ and $|V'| \times p$. Indicator functions of k-ring neighborhoods of selected vertices for correspondence are employed as the p corresponding functions $\mathbf{f}_i$ and $\mathbf{g}_i$, where k is usually one of $3, 4$ and 5. We always set $k = 3$ for reference meshes and choose the value of k for source meshes such that it minimizes the difference between the normalized areas of the two corresponding regions. Besides, in our experiments p is about 30 for the whole models.

In the rest of this section, we first explain the whole process of our approach using an example. Following that, we present a variety of examples to show the effect for various topological and geometric configurations, and meanwhile compare our results with that of methods described in Lévy (2006), Kovnatsky et al. (2013). We also analyze the efficiency of our approach by presenting the running time of key steps and finally discuss its limitations. To save space, some basic information such as vertex number and face number for all examples is summarized in Table 1 together with running time.

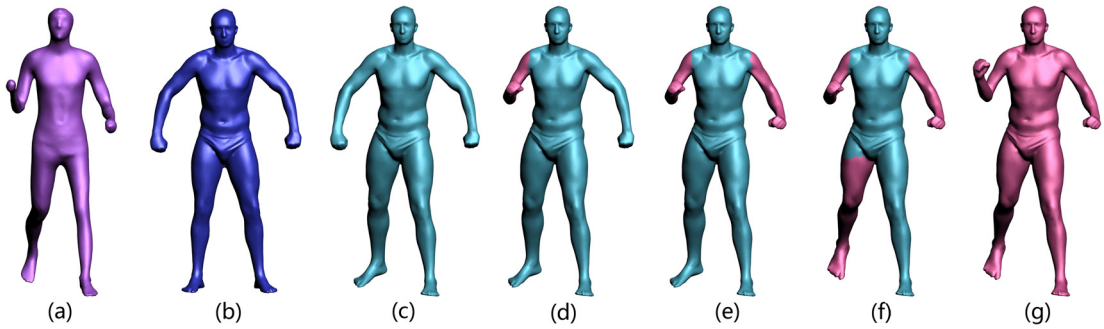

Fig. 3. Hierarchical pose transfer: (a) the reference pose; (b) the source pose; (c) the result of the whole pose transfer; (d) the right arm medium-scale pose transfer, (e) the left arm medium-scale pose transfer, (f) the right leg medium-scale pose transfer and (g) the final result after adjusting the orientation of each part.

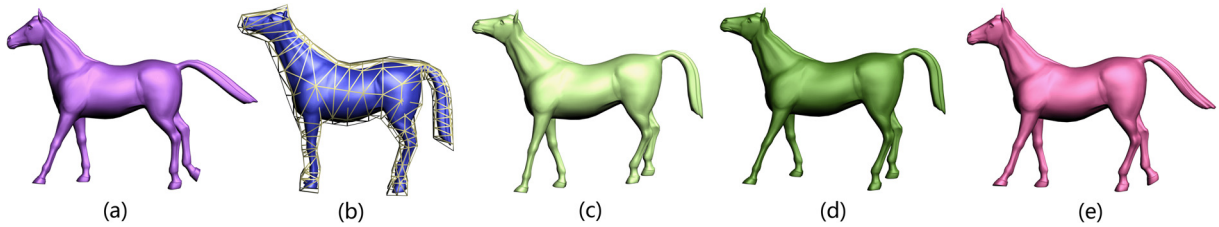

Fig. 4. Pose transfer between meshes with the same connectivity of the same object: (a) reference pose; (b) source pose and cage; (c) the result of Lévy (2006); (d) the result of Kovnatsky et al. (2013); (e) two-level pose transfer by our method.

Hierarchical pose transfer Fig. 3 presents an example to show different-scale pose transfer step by step. In this example, two levels of hierarchical pose transfer are performed. Firstly, the first level pose transfer is performed for the whole model (Fig. 3(c)) and then the second level pose transfer are successively conducted for the right arm (Fig. 3(d)), the left arm (Fig. 3(e)) and the right leg (Fig. 3(f)), respectively. The parts that are segmented from the entirely model for local pose transfer are shown in red color in the figure. Notice that we reuse the resulting cage from the last transfer as the cage of all local parts to be processed instead of constructing a sole cage for each of them. Nevertheless, we always fix those cage vertices that are far away from the part to be processed.

The same object with the same connectivity The simplest case is pose transfer between two meshes of the same object with the same connectivity. Fig. 4 illustrates such an example in which (a) and (b) are reference and source meshes with a cage drawn in wireframe. From (c) to (e) are the pose transferring results by approaches in Lévy (2006) and Kovnatsky et al. (2013) as well as our methods. The first five images in Fig. 5 illustrate the second to the sixth bases of Laplacian matrices of the reference mesh (upper row) and the source meshes (bottom row). These bases are directly used by Lévy (2006). The correspondence between two groups of bases looks good enough and therefore the pose transfer is successful to certain extent (see Fig. 4(c)). We also illustrate the functional matrix (rightmost one of Fig. 5) between coupled quasi-harmonic bases of two models used by Kovnatsky et al. (2013) and our method. It shows that the correspondence between coupled quasi-harmonic bases is also good enough. The pose transfer result by the method described in Kovnatsky et al. (2013) is comparable to that of the Lévy's method. Nevertheless, though there exists little distortion in the result (Fig. 4(c)) by Lévy (2006) and result (Fig. 4(d)) by the coupled quasi-harmonic technique (Kovnatsky et al., 2013), our pose transfer (Fig. 4(e)) is more sufficient in learning poses of the head, the tail and the rear right leg.

Different objects with the same connectivity The case of different objects with the same connectivity is relatively general. In Fig. 6, the pose of a cat (Fig. 6(a)) is transferred onto a tiger (Fig. 6(b)). Both reference and source meshes are topologically identical. Also, results by three methods, Lévy (2006), Kovnatsky et al. (2013) and ours are illustrated in Fig. 6(c)–(e), respectively. In this example, the result by Lévy (2006) is still very unsatisfactory. The original Laplacian bases used by this method are illustrated in Fig. 7 which shows that the correspondence is very poor. In this figure, the rightmost image displays the correspondence matrix between two sets of coupled quasi-harmonic bases from the two models. Though the correspondence looks good, the result by Kovnatsky et al. (2013) also exhibits serious distortion or insufficient pose transfer. Our pose transfer is better in learning poses of the head, the tail and the front right leg. Note that our result is generated with two levels of pose transfer. In the second level, we perform pose transfer on the tail, head, left rear paw and right front paw separately.

The same object with different connectivities Here the same object means that the reference and source are similar in geometric shape but different in motion pose. Fig. 8 shows that our approach also works for this configuration. Fig. 8(a) and (b) are respectively reference and source meshes which possesses different triangulation. Fig. 8(c) is generated by Kovnatsky et al. (2013). Obviously, their method cannot capture the tail pose of the reference cat. Fig. 8(d) depicts our result which is generated by the first level pose transfer of the whole model and the second level pose transfer of the tail.

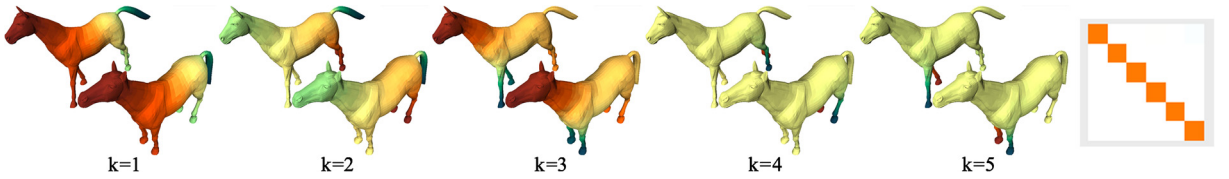

Fig. 5. The second to sixth Laplacian eigenfunctions are visualized for both reference and source meshes from the first to fifth columns. The rightmost: the functional matrix between coupled quasi-harmonic bases between two reference and source poses.

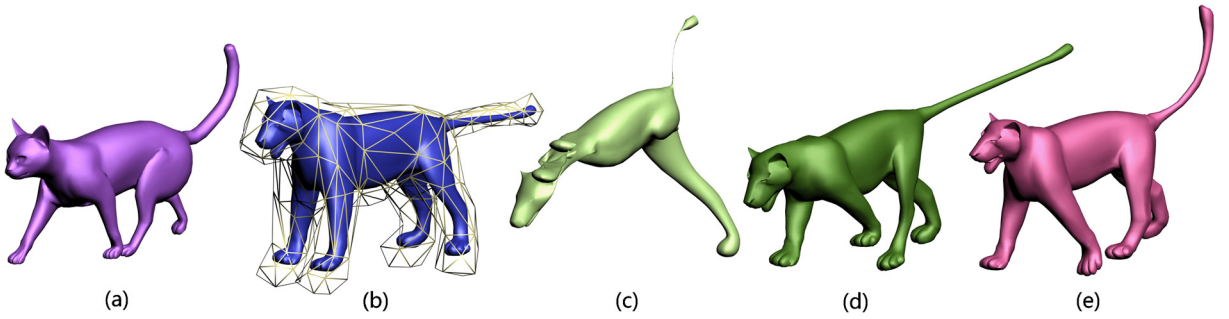

Fig. 6. Pose transfer between different objects with the same mesh connectivity. (a) reference; (b) source; (c) result by Lévy (2006); (d) result by Kovnatsky et al. (2013); and (e) result by our approach.

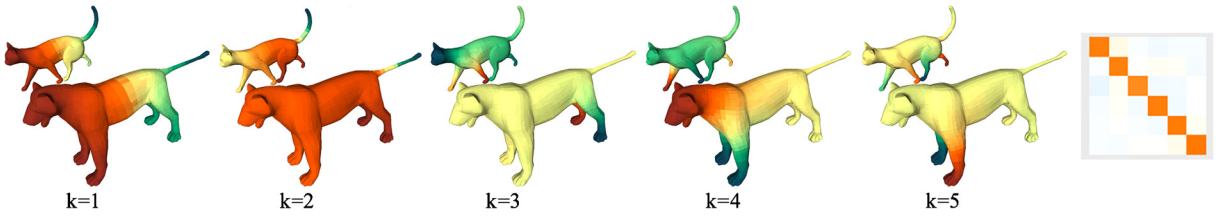

Fig. 7. From left to right is the visualization of the original bases corresponding to the second to the sixth smallest eigenvalues while the rightmost image depicts the functional matrix between two sets of coupled quasi-harmonic bases for the two models.

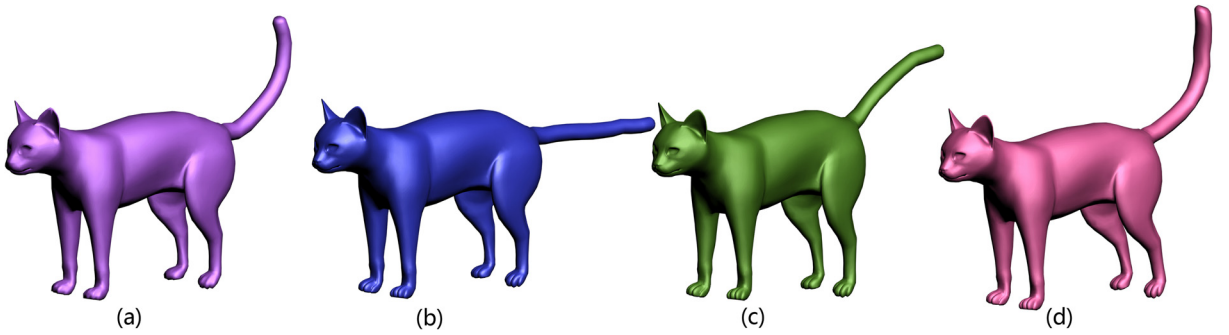

Fig. 8. Pose transfer between meshes of the same object with the different connectivities: (a) reference; (b) source; (c) the result of Kovnatsky et al. (2013); (d) our result.

General examples – different objects with different connectivities Now we discuss two general examples in which the reference and source meshes represent different objects and have different connectivities. Fig. 9 is an example for pose transfer between two different men with Fig. 9(a) as reference and Fig. 9(b) as source. Their mesh models have different geometry, topology and pose. Fig. 9(c) and (d) are respectively results by Kovnatsky et al. (2013) and our method. After the first level pose transfer, we apply the second level pose transfer in the arms, the legs and the head of the source mesh. It can be observed that the result by our method is better in learning head orientation and the poses of two legs. Finally, we present more general example in which the reference and source models are from different species. It transfers the pose of a man model (Fig. 10(a)) to the armadillo model (Fig. 10(b)). Fig. 10(c)–(e) are results of Lévy (2006), Kovnatsky et al. (2013) and this paper, respectively. We conduct the secondary pose transfer on the two legs and the head of the two models in this example. Notice that both former methods exhibit serious distortion while our result looks satisfactory.

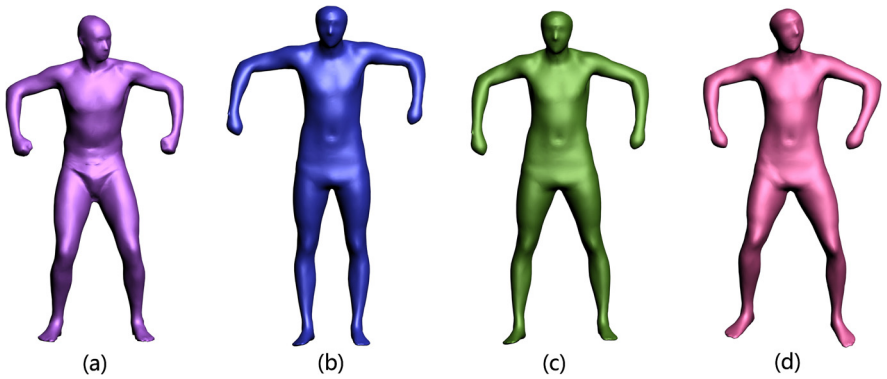

Fig. 9. Pose transfer between meshes of two men with the different connectivities: (a) reference; (b) source; (c) result by Kovnatsky et al. (2013); (d) ours.

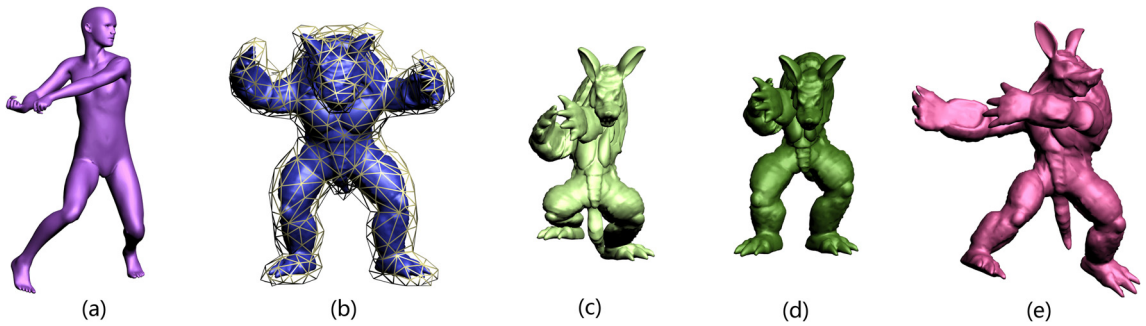

Fig. 10. Pose transfer from a man to the armadillo: (a) reference; (b) source; (c) result by Lévy (2006); (d) result by Kovnatsky et al. (2013); (e) ours.

Table 1
Timings for aforementioned examples (unit: second).

Figures	Ref. poses (V/F num.)	Source poses (V/F num.)	Cages (V/F num.)	Comput. of bases & MVC	Pose transfer opti. (iter. times)	The second level (the third level)
Fig. 1	19248/38492	10889/21774	452/900	122.079	31.479(10)	140.02 (20.706)
Fig. 3	6442/12880	7506/15008	166/328	53.135	9.812(10)	123.192
Fig. 4	8431/16858	8431/16858	348/692	32.772	7.825(5)	127.559
Fig. 6	7207/14410	7207/14410	520/1036	52.092	63.573(20)	288.196
Fig. 8	3604/7204	7207/14410	319/634	29.111	8.039(5)	33.322
Fig. 9	6241/12863	6449/12894	173/342	23.524	3.762(4)	65.052
Fig. 10	10002/20000	10495/20986	502/1000	55.474	16.775(5)	173.517

Timings To show the efficiency of the proposed approach, we report the running time of key steps such as eigen-decomposition of Laplacian matrices, computation of coupled quasi-harmonic bases, evaluation of mean-value coordinates (MVC) and pose transfer optimization, for all examples used in the paper in Table 1. We also list the size of the reference pose, the source pose and the associated cage. Moreover, we also give the iteration number in addition to the whole optimization time for the first level pose transfer while we only show the total time cost for all secondary pose transfers. As only Fig. 1 requires the third level pose transfer, we attach its consuming time to the end of the secondary pose transfer column (see the number in the small brace).

Limitations As cages used for subspace solution is generated in advance, they are not deformation-aware. Overstocked control points may lead to pose over-learning while too few control points may result in details distortion. In a word, the pose transfer quality is heavily dependent on the quality of cages to a certain extent. In Fig. 11, (a) is the reference pose, (b) and (d) are source poses with different cages, and (c) and (e) are respectively results of two cases. The cage in (b) is better than that in (d) (see the regions marked by red rectangles) and therefore its result (Fig. 11(c)) is also better than that of the other (Fig. 11(e)). Fig. 12 gives another example in which the right hand of the man in (d) is inappropriately bent.

6. Conclusion

We have described a spectral-based pose transfer approach. It is insufficient to learn the pose of a reference by only using low-frequency component. Therefore, our approach firstly creates a hierarchical pose structure to capture medium-scale poses. It then segments the components including medium-scale pose. Finally, the approach performs pose transfer

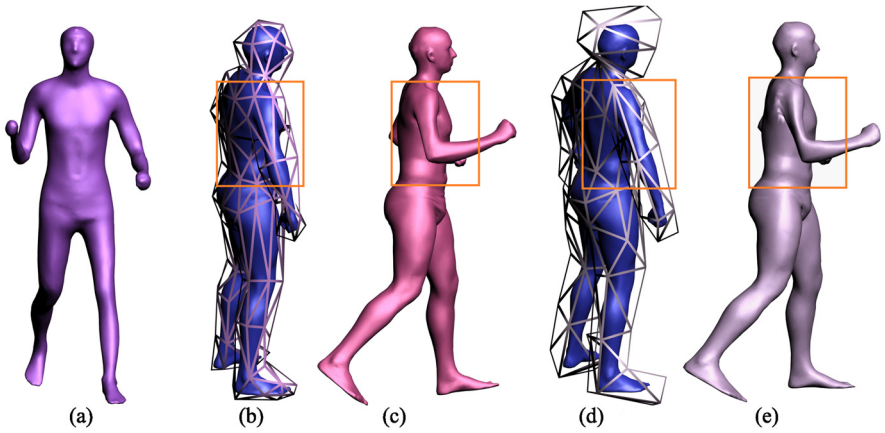

Fig. 11. Bad cage quality may result in serious distortion: (a) reference pose; (b) and (c) are respectively source pose with a smoother cage and the corresponding result; (d) and (e) are respectively source pose with another cage and corresponding result.

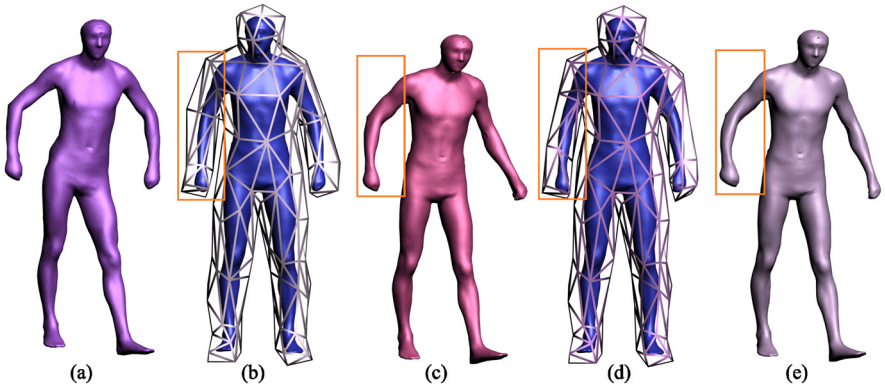

Fig. 12. Bad cage quality may result in overlearning: (a) reference pose; (b) source pose with a good cage; (c) pose transfer result of (b); (d) source pose with a bad cage; (e) the result of (d). In (d), the right hand control points of the cage cannot align the joint well, and therefore the resulting right hand is overbent (see the regions marked by rectangles).

operation for low-scale frequency on two segmented parts. To prevent distortion during transferring process, we introduce a penalty to preserve Laplacian coordinates. Based on the new deformation formulation, a framework is established for a variety of applications such as pose transfer between two meshes with different connectivities, mesh deformation, and shape interpolation.

As future work, the current computational method for coupled quasi-harmonic basis is not stable enough, particularly, when one of the mesh models is too coarse or too irregular. In addition, at this stage, we need to interactively specify parts in which pose learning is incomplete. It is desirable to automatically detecting and segmenting those parts. Besides, triangulation quality of cages is also a significant factor impacting on the transferring results. Adaptively refining cages may be a good choice to control the degree of freedom. Finally, how to automatically choose the weights in Eq. (12) is an important issue we need to tackle in the next step.

Acknowledgements

This work is partially supported by the Key Program of Natural Science Foundation of Guangdong, Province, China (S2013020012795), Doctoral Funds of Ministry of Education of China (20130172110010, 20130172120041), Natural Science Foundation of Guangxi Province, China (2014GXNSFAA118361), and Natural Science Foundation of China (61300136). We want to express gratitude to Artiom Kovnatsky for providing codes for computing coupled quasi-harmonic bases and models in Fig. 1. Thanks also go to Nils Hasler for permitting to use the human models from http://www.mpi-inf.mpg.de/resources/scandb/.

Appendix A. Express local transformations of Laplacian operators with position coordinates

The formulation by Sorkine et al. (2004) is employed in our method to represent the parameters of the transformation matrices in terms of vertices for preserving Laplacian coordinates up to a rotation for cages. We list the derivation in this appendix for completeness.

According to Sorkine et al. (2004), the transformation of the vertex i deforming cage C to $\tilde{C}$ has the following simplified form:

$$R_i(\tilde{C}) = \begin{pmatrix} s & -c & b \\ c & s & -a \\ -b & a & s \end{pmatrix},$$

where s is scale factor, and entries a, b, c determine the rotation. Parameters of $R_i(\tilde{C})$ can be expressed in terms of vertices of $\tilde{C}$ by solving the following optimization:

$$\min_{R_i(\tilde{C})} (\|R_i(\tilde{C})v_i - \tilde{v}_i\|^2 + \sum_{j\in\mathcal{N}_i} \|R_i(\tilde{C})v_j - \tilde{v}_j\|^2), \tag{13}$$

where $\tilde{v}_i$ and $\tilde{v}_j$ are the vertices of $\tilde{C}$ and $\mathcal{N}_i$ is the set of neighbors of vertex i.

Denoting the 3D coordinates of v_i as $(v_i(x), v_i(y), v_i(z))$, we have

$$\begin{aligned} R_i(\tilde{C})v_i &= \begin{pmatrix} s & -c & b \\ c & s & -a \\ -b & a & s \end{pmatrix} \begin{pmatrix} v_i(x) \\ v_i(y) \\ v_i(z) \end{pmatrix} \\ &= \begin{pmatrix} v_i(x) & 0 & v_i(z) & -v_i(y) \\ v_i(y) & -v_i(z) & 0 & v_i(x) \\ v_i(z) & v_i(y) & -v_i(x) & 0 \end{pmatrix} \begin{pmatrix} s \\ a \\ b \\ c \end{pmatrix}. \end{aligned}$$

Let

$$A_i = \begin{pmatrix} \vdots & & & \\ v_{k_x} & 0 & v_{k_z} & -v_{k_y} \\ v_{k_y} & -v_{k_z} & 0 & v_{k_x} \\ v_{k_z} & v_{k_y} & -v_{k_x} & 0 \\ \vdots & & & \end{pmatrix}, B_i = \begin{pmatrix} \vdots \\ \tilde{v}_k(x) \\ \tilde{v}_k(y) \\ \tilde{v}_k(z) \\ \vdots \end{pmatrix},$$

where vertical ellipsis dots means all $k \in \{i\} \bigcup \mathcal{N}_i$ are collected. Eq. (13) can be converted into the following minimization:

$$\min \|A_i(s \ a \ b \ c)^T - B_i\|^2.$$

It follows

$$\begin{pmatrix} s \\ a \\ b \\ c \end{pmatrix} = (A_i^T A_i)^{-1} A_i^T B_i = U B_i = \begin{pmatrix} U_0 \\ U_1 \\ U_2 \\ U_3 \end{pmatrix} B_i,$$

$$R_i(\tilde{C}) = \begin{pmatrix} U_0 B_i & -U_3 B_i & U_2 B_i \\ U_3 B_i & U_0 B_i & -U_1 B_i \\ -U_2 B_i & U_1 B_i & U_0 B_i \end{pmatrix}.$$

It is easy to see that the scale factor $s = U_0 B_i$, and this is used in the scale constraint in Eq. (11).

Appendix B. A brief explanation to minimization of Eq. (12)

It is not difficult to know that all energies in Eq. (12) are quadratic function of coordinates of cage $\tilde{C}$. Particularly,

$$\begin{aligned} \mathcal{E}_{LC}^C &= \sum_{i=0}^{|C|-1} \|R_i(\tilde{C})\delta_i^C - \tilde{\delta}_i^{\tilde{C}}\|^2 \\ &= \sum_{i=0}^{|C|-1} \left\| \begin{pmatrix} U_0 B_i & -U_3 B_i & U_2 B_i \\ U_3 B_i & U_0 B_i & -U_1 B_i \\ -U_2 B_i & U_1 B_i & U_0 B_i \end{pmatrix} \delta_i^C - \tilde{\delta}_i^{\tilde{C}} \right\|^2 \\ &= \sum_{i=0}^{|C|-1} \left\| \begin{pmatrix} U_0 \delta_i^x B_i - U_3 \delta_i^y B_i + U_2 \delta_i^z B_i \\ U_3 \delta_i^x B_i + U_0 \delta_i^y B_i - U_1 \delta_i^z B_i \\ -U_2 \delta_i^x B_i + U_1 \delta_i^y B_i + U_0 \delta_i^z B_i \end{pmatrix} - \begin{pmatrix} \tilde{\delta}_i^x \\ \tilde{\delta}_i^y \\ \tilde{\delta}_i^z \end{pmatrix} \right\|^2 \end{aligned}$$

$$= \sum_{i=0}^{|C|-1} \left\| \begin{pmatrix} U_0\delta_i^x - U_3\delta_i^y + U_2\delta_i^z \\ U_3\delta_i^x + U_0\delta_i^y - U_1\delta_i^z \\ -U_2\delta_i^x + U_1\delta_i^y + U_0\delta_i^z \end{pmatrix} B_i - \begin{pmatrix} \tilde{\delta}_i^x \\ \tilde{\delta}_i^y \\ \tilde{\delta}_i^z \end{pmatrix} \right\|^2$$

$$= \| O\tilde{C} - L\tilde{C} \|^2 = \| (O - L)\tilde{C} \|^2,$$

where L is the Laplacian matrix of cage C, $\delta_i^C = \left(\delta_i^x,\ \delta_i^y,\ \delta_i^z \right)^T$ and $\tilde{\delta}_i^{\tilde{C}} = \left(\tilde{\delta}_i^x,\ \tilde{\delta}_i^y,\ \tilde{\delta}_i^z \right)^T$ are the Laplacian coordinates of C and $\tilde{C}$, respectively. Since Ψ and ϕ_i can be computed in advance, optimizing Eq. (12) reduces to a linear least-squares problem.

Appendix C. Supplementary material

Supplementary material related to this article can be found online at http://dx.doi.org/10.1016/j.cagd.2015.03.016.

References

Baran, I., Vlasic, D., Grinspun, E., Popović, J., 2009. Semantic deformation transfer. ACM Trans. Graph. 28, 36. http://dx.doi.org/10.1145/1576246.1531342.

Ben-Chen, M., Weber, O., Gotsman, C., 2009. Spatial deformation transfer. In: Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation. ACM, pp. 67–74. http://dx.doi.org/10.1145/1599470.1599479.

Bobenko, A.I., Springborn, B.A., 2007. A discrete Laplace–Beltrami operator for simplicial surfaces. Discrete Comput. Geom. 38, 740–756.

Capell, S., Green, S., Curless, B., Duchamp, T., Popović, J., 2002. Interactive skeleton-driven dynamic deformations. ACM Trans. Graph. 21, 586–593. http://dx.doi.org/10.1145/566570.566622.

Chen, L., Sun, H., Bao, H., 2010. Cage-based deformation transfer. Comput. Graph. 34, 107–108. http://dx.doi.org/10.1016/j.cag.2010.01.003.

Dey, T.K., Ranjan, P., Wang, Y., 2012. Eigen deformation of 3D models. Vis. Comput. 28, 585–595. http://dx.doi.org/10.1007/s00371-012-0705-0.

Dyn, N., 1990. A butterfly subdivision scheme for surface interpolatory with tension control. ACM Trans. Graph. 9, 160–169. http://dx.doi.org/10.1145/78956.78958.

Fisher, M., Springborn, B., Bobenko, A.I., Schroder, P., 2006. An algorithm for the construction of intrinsic Delaunay triangulations with applications to digital geometry processing. In: ACM SIGGRAPH 2006 Courses. ACM, pp. 69–74.

Fröhlich, S., Botsch, M., 2011. Example-driven deformations based on discrete shells. Comput. Graph. Forum 30, 2246–2257. http://dx.doi.org/10.1111/j.1467-8659.2011.01974.x.

Gain, J., Bechmann, D., 2008. A survey of spatial deformation from a user-centered perspective. ACM Trans. Graph. 27, 107. http://dx.doi.org/10.1145/1409625.1409629.

Huang, J., Shi, X., Liu, X., Zhou, K., Wei, L.Y., Teng, S.H., Bao, H., Guo, B., Shum, H.Y., 2006. Subspace gradient domain mesh deformation. ACM Trans. Graph. 25, 1126–1134. http://dx.doi.org/10.1145/1179352.1142003.

Huang, Q., Wang, F., Guibas, L., 2014. Functional map networks for analyzing and exploring large shape collections. ACM Trans. Graph. 33, 36. http://dx.doi.org/10.1145/2601097.2601111.

Huang, J., Zhang, M., Ma, J., Liu, X., Kobbelt, L., Bao, H., 2008. Spectral quadrangulation with orientation and alignment control. ACM Trans. Graph. 27, 147. http://dx.doi.org/10.1145/1457515.1409100.

Igarashi, T., Moscovich, T., Hughes, J.F., 2008. As-rigid-as-possible shape manipulation. ACM Trans. Graph. 24, 134–1143. http://dx.doi.org/10.1145/1186822.1073323.

Ji, Z., Liu, L., Chen, Z., Wang, G., 2006. Easy mesh cutting. Comput. Graph. Forum (Proceedings of Eurographics) 25, 283–291. http://dx.doi.org/10.1111/j.1467-8659.2006.00947.x.

Ju, T., Schaefer, S., Warren, J., 2005. Mean value coordinates for closed triangular meshes. ACM Trans. Graph. 24, 561–566. http://dx.doi.org/10.1145/1186822.1073229.

Karni, Z., Gotsman, C., 2000. Spectral compression of mesh geometry. In: Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques. ACM Press/Addison–Wesley Publishing Co., pp. 279–286. http://dx.doi.org/10.1145/344779.344924.

Kilian, M., Mitra, N., Pottmann, H., 2007. Geometric modeling in shape space. ACM Trans. Graph. 26, 64. http://dx.doi.org/10.1145/1275808.1276457.

Kircher, S., Garland, M., 2005. Progressive multiresolution meshes for deforming surfaces. In: 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 191–200. http://dx.doi.org/10.1145/1073368.1073395.

Kovnatsky, A., Bronstein, M.M., Bronstein, A.M., Glashoff, K., Kimmel, R., 2013. Coupled quasi-harmonic bases. Comput. Graph. Forum 32, 439–448. http://dx.doi.org/10.1111/cgf.12064.

Lévy, B., 2006. Laplace–Beltrami eigenfunctions towards an algorithm that "understands" geometry. In: IEEE International Conference on Shape Modeling and Applications-SMI 2006. IEEE, p. 13. http://dx.doi.org/10.1109/SMI.2006.21.

Lévy, B., Zhang, H.R., 2010. Spectral mesh processing. In: ACM SIGGRAPH 2010 Courses. ACM, p. 8. http://dx.doi.org/10.1111/j.1467-8659.2010.01655.x.

Li, X.Y., Ju, T., Hu, S.M., 2013. Cubic mean value coordinates. ACM Trans. Graph. 32, 126. http://dx.doi.org/10.1145/2461912.2461917.

Lipman, Y., Levin, D., Cohen-Or, D., 2008. Green coordinates. ACM Trans. Graph. 27, 78. http://dx.doi.org/10.1145/1360612.1360677.

Lipman, Y., Sorkine, O., Cohen-Or, D., Levin, D., Rossi, C., Seidel, H.P., 2004. Differential coordinates for interactive mesh editing. In: Shape Modeling Applications, 2004. Proceedings. IEEE, pp. 181–190. http://dx.doi.org/10.1109/SMI.2004.1314505.

Liu, Y.J., Xu, C.X., He, Y., Kim, D.S., 2014. The duality of geodesic Voronoi/Delaunay diagrams for an intrinsic discrete Laplace–Beltrami operator on simplicial surfaces. In: 26th Canadian Conference on Computational Geometry. CCCG.

Ovsjanikov, M., Ben-Chen, M., Solomon, J., Butscher, A., Guibas, L., 2012. Functional maps: a flexible representation of maps between shapes. ACM Trans. Graph. 31, 30. http://dx.doi.org/10.1145/2185520.2185526.

Rong, G., Cao, Y., Guo, X., 2008. Spectral mesh deformation. Vis. Comput. 24, 787–796. http://dx.doi.org/10.1007/s00371-008-0260-x.

Song, R., Liu, Y., Martin, R.R., Rosin, P.L., 2014. Mesh saliency via spectral processing. ACM Trans. Graph. 33, 6. http://dx.doi.org/10.1145/2530691.

Sorkine, O., Alexa, M., 2007. As-rigid-as-possible surface modeling. In: Symposium on Geometry Processing, pp. 109–116. http://dl.acm.org/citation.cfm?id=1281991.1282006.

Sorkine, O., Cohen-Or, D., Lipman, Y., Alexa, M., Rössl, C., Seidel, H.P., 2004. Laplacian surface editing. In: Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing. ACM, pp. 175–184. http://dx.doi.org/10.1145/1057432.1057456.

Sumner, R.W., Popović, J., 2004. Deformation transfer for triangle meshes. ACM Trans. Graph. 23, 399–405. http://dx.doi.org/10.1145/1015706.1015736.

Sumner, R.W., Schmid, J., Pauly, M., 2007. Embedded deformation for shape manipulation. ACM Trans. Graph. 26, 80. http://dx.doi.org/10.1145/1275808.1276478.

Weber, O., Sorkine, O., Lipman, Y., Gotsman, C., 2007. Context-aware skeletal shape deformation. Comput. Graph. Forum 26, 265–274. http://dx.doi.org/10.1111/j.1467-8659.2007.01048.x.

Winkler, T., Drieseberg, J., Alexa, M., Hormann, K., 2010. Multi-scale geometry interpolation. Comput. Graph. Forum 29, 309–318. http://dx.doi.org/10.1111/j.1467-8659.2009.01600.x.

Xu, W.W., Zhou, K., 2009. Gradient domain mesh deformation survey. J. Comput. Sci. Technol. 24, 6–18. http://dx.doi.org/10.1007/s11390-009-9209-4.

Zhang, S., Huang, J., Metaxas, D.N., 2011. Robust mesh editing using Laplacian coordinates. Graph. Models 73, 10–19. http://dx.doi.org/10.1016/j.gmod.2010.10.003.

Zhang, H., van Kaick, O., Dyer, R., 2007. Spectral methods for mesh processing and analysis. In: Proceedings of Eurographics State-of-the-Art Report, pp. 1–22. https://www.cs.sfu.ca/~haoz/pubs/zhang_eg07star_spectral.pdf.

Zhou, K., Huang, J., Snyder, J., Liu, X., Bao, H.B., Guo, B., Shum, H.Y., 2005. Large mesh deformation using the volumetric graph Laplacian. ACM Trans. Graph. 24, 496–503. http://dx.doi.org/10.1145/1186822.1073219.

Zhou, K., Xu, W., Tong, Y., Desbrun, M., 2010. Deformation transfer to multi-component objects. Comput. Graph. Forum 29, 319–325. http://dx.doi.org/10.1111/j.1467-8659.2009.01601.x.