

# Efficient and realistic character animation through analytical physics-based skin deformation

Shaojun Bian<sup>a,\*</sup>, Zhigang Deng<sup>b</sup>, Ehtaz Chaudhry<sup>a</sup>, Lihua You<sup>a</sup>, Xiaosong Yang<sup>a</sup>, Lei Guo<sup>c</sup>, Hassan Ugail<sup>d</sup>, Xiaogang Jin<sup>e</sup>, Zhidong Xiao<sup>a</sup>, Jian Jun Zhang<sup>a,f</sup>

<sup>a</sup> Bournemouth University, Fern Barrow, Poole, Dorset, BH12 5BB, United Kingdom

<sup>b</sup> University of Houston, Houston, Texas, U.S.

<sup>c</sup> China Academy of Art, China

<sup>d</sup> University of Bradford, United Kingdom

<sup>e</sup> Zhejiang University, China

<sup>f</sup> Southwest Jiaotong University, China

## ARTICLE INFO

### Keywords:

Character animation  
Realistic skin deformation  
Fourier series representations  
Physics-based model  
Analytical solution

## ABSTRACT

Physics-based skin deformation methods can greatly improve the realism of character animation, but require non-trivial training, intensive manual intervention, and heavy numerical calculations. Due to these limitations, it is generally time-consuming to implement them, and difficult to achieve a high runtime efficiency. In order to tackle the above limitations caused by numerical calculations of physics-based skin deformation, we propose a simple and efficient analytical approach for physics-based skin deformations. Specifically, we (1) employ Fourier series to convert 3D mesh models into continuous parametric representations through a conversion algorithm, which largely reduces data size and computing time but still keeps high realism, (2) introduce a partial differential equation (PDE)-based skin deformation model and successfully obtain the first analytical solution to physics-based skin deformations which overcomes the limitations of numerical calculations. Our approach is easy to use, highly efficient, and capable to create physically realistic skin deformations.

## 1. Introduction

Skin deformation techniques have been used as a standardized part of many character animation applications these days in both academia and industry practices. Over the years, researchers have developed a variety of skin deformation techniques to improve these major characteristics: *efficiency*, *realism*, and *ease of use*. In particular, Despite the simplicity and efficiency, geometric skin deformation approaches typically fall short of producing highly realistic skin deformation without non-trivial additional efforts. In order to achieve high realism, example-based approaches need a sufficient number of skin examples although they are often used together with geometric skinning methods. Physics-based approaches arguably have produced most realistic skin deformation results to date, but they often rely on heavy numerical calculations and specialized user knowledge and skills. Most existing physics-based skin deformation techniques work in a discrete vertex space to obtain discrete numerical solutions of skin deformation, causing the aforementioned limitations. Analytical solutions, if obtainable,

can overcome the limitations of numerical calculations of physics-based method. However, obtaining an efficient analytical solution to physics-based skin deformation is a non-trivial task.

Inspired by the above challenges, in this paper we develop a novel approach based on an analytical solution to physics-based modeling, which is capable of producing physically-realistic skin deformation with high efficiency. Our method requires at least two example meshes obtained from either captured or artist-sculpted shapes; any two adjacent examples constitute one sequence to create intermediate skin deformations (an example is shown in Fig. 1). The advantages of our approach include: high efficiency, high realism, and ease of use, as detailed below. *High efficiency*. With the obtained analytic solution to the formulated PDE-based physics model, our approach transforms a discrete example mesh into its continuous Fourier series representations, avoiding the solving of a large set of linear equations, which largely reduces data size and computing time.

*High realism*. Integrating physics-based and example-based approaches leads to more realistic skin deformations. We utilize a small number of provided example meshes in our physics-based mathemati-

\* Corresponding author.

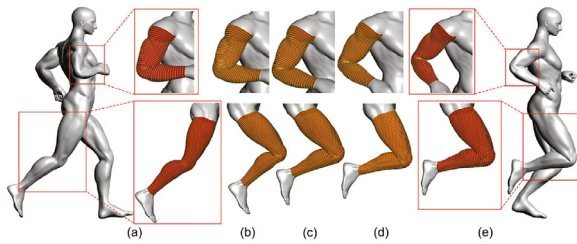
E-mail addresses: [bians@bournemouth.ac.uk](mailto:bians@bournemouth.ac.uk) (S. Bian), [zdeng4@uh.edu](mailto:zdeng4@uh.edu) (Z. Deng), [echaudhry@bournemouth.ac.uk](mailto:echaudhry@bournemouth.ac.uk) (E. Chaudhry), [LYou@bournemouth.ac.uk](mailto:LYou@bournemouth.ac.uk) (L. You), [xyang@bournemouth.ac.uk](mailto:xyang@bournemouth.ac.uk) (X. Yang), [sixstoneshi@qq.com](mailto:sixstoneshi@qq.com) (L. Guo), [H.Ugail@Bradford.ac.uk](mailto:H.Ugail@Bradford.ac.uk) (H. Ugail), [jin@cad.zju.edu.cn](mailto:jin@cad.zju.edu.cn) (X. Jin), [ZXiao@bournemouth.ac.uk](mailto:ZXiao@bournemouth.ac.uk) (Z. Xiao), [jzhang@bournemouth.ac.uk](mailto:jzhang@bournemouth.ac.uk) (J.J. Zhang).

<https://doi.org/10.1016/j.gmod.2019.101035>

Received 18 February 2019; Received in revised form 28 May 2019; Accepted 3 June 2019

Available online 7 June 2019

1524-0703/© 2019 Elsevier Inc. All rights reserved.



**Fig. 1.** Our proposed method can generate physically realistic character animation efficiently given two example poses, shown in (a) and (e). (b)–(d) are intermediate poses generated by our approach.

cal model, in order to determine the correct force field acting on skin surfaces and achieve the high realism of skin deformations.

**Explicit solver.** Our solution of physics-based skin deformations is fully analytical and able to explicitly and quickly determine physically realistic skin deformations with high automation. Due to the fully analytical nature and few manual operations, it is easy to implement and use without the need of physics-based knowledge and skills.

## 2. Related work

**Geometric skin deformation.** Linear Blend Skinning (LBS), also called Skeleton Subspace Deformation (SSD), is the most well-known geometric skinning algorithm [27] due to its efficiency and simplicity. However, its limitations include collapsing elbow, candy-wrapper effects, and failure of secondary deformation [24]. A novel skinning algorithm based on linear combination of dual quaternions was presented in [17] to tackle some drawbacks of LBS. By introducing an extra scalar weight function per bone, a simple modification of the LBS formulation was presented in [13] to enable stretching and twisting without changing the existing skeleton rig or bone weights. Le and Hodgins [22] choose to pre-compute the optimized center of rotation for each vertex, and use these centers to interpolate rigid transformations. Vaillant et al. [37] proposed a pure geometric method to handle skin contact and muscle bulge problems in real-time, but it fails to address deep self-intersections. To solve this problem, later Vaillant and Colleagues [38] use new composition operators to enable blending effects and local self-contact between implicit surfaces. Zhao et al. [44] presents a robust automatic algorithm to deform the meshes of arbitrary shapes into polycube form. Despite various efforts in recent years, geometric skinning still need to correct various artifacts to simulate the skin deformations.

**Example-based skin deformation** is employed to address the realism issue of geometric skinning, by learning deformation dynamics from a set of given examples [19,35]. An automated framework was presented in [29] to fit the parameters of a deformation model using a set of examples consisting of skeleton configurations paired with the deformed geometry as static meshes. Park and Hodgins [31] presented a data-driven technique to synthesize skin deformation from skeletal motion. Eulerian representation of skin was proposed in [25] to simulate thin hyper-elastic skin that can stretch and slide over the underlying body structures such as muscles, bones, and tendons. One automated algorithm, called Smooth Skinning Decomposition with Rigid Bones (SSDR), was introduced in [21] to extract the LBS from a set of example poses in good accuracy and applicability. Pons-Moll et al. [32] use a second-order auto-regressive model that predicts soft-tissue deformations based on previous deformations and realistically represents the dynamics of soft tissue. Wampler [39] describes a new approach for example-based inverse kinematic mesh manipulation, which generates high quality deformations for a wide range of inputs, and in particular works well even for provided stylized or cartoony examples, but it needs to develop a fast specialized solver to deal with the cases of many handles and input shapes. Gao et al. [9] introduced a new rotation invariant mesh difference representation to encode mesh deformations, and a novel re-

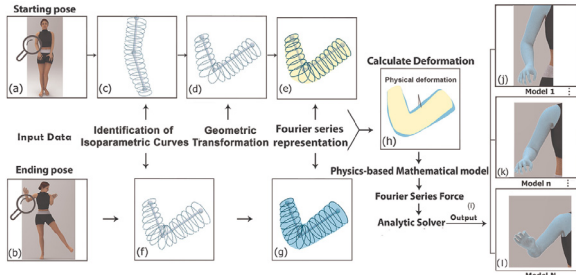
construction algorithm that efficiently solves for the vertex positions and local rotations simultaneously. The representation allows combining multiple deformations by a linear combination. But the limitation of this data-driven method is that it may not produce realistic results if the dataset is not large enough to cover target deformations or scans. Thus, one major limitation of example-based skin deformation methods is the need for sufficient example skin shapes in order to produce realistic skin deformations.

**Physics-based skin deformation** methods are designed to simulate the underlying physics for creating realistic skin deformations. In [2], a volume preserving method was presented to avoid extra bulges or wrinkles, which uses vector field integration to avoid self-collisions. However, it is computationally expensive. Kavan and Sorkine [18] presented an elasticity-based skinning method to generate high quality deformations through the optimization of skinning weights and the introduction of joint-based deformer. A sophisticated biomechanical model of the human upper body was developed in [23], where a coupled finite element model is employed to simulate biomechanically realistic flesh deformations. McAdams et al. [28] introduced a novel discretization of corotational elasticity over a hexahedral lattice to achieve near-interactive simulation of skeleton-driven, high resolution elasticity models, but it still needs several seconds per animation frame. Since it is hard to model the different elastic behaviors of muscle, fat, and skin using simple volumetric meshes, Deul and Bender [7] introduced one novel multi-layer model to simulate them, but it fails to handle collisions for fast motions. Jacobson et al. [11] presented a well-coordinated combination of a reduced deformable body model with nonlinear finite elements, a linear-time algorithm for skeleton dynamics, and explicit integration, to boost the simulation speed. In order to reduce the computational cost of physics-based skin deformations, model reduction has been introduced into static [19] and dynamic [36] deformation simulations. It can achieve real-time performance, but reduce the computational accuracy and increase the implementation complexity. Elastic animation editing with spacetime constraints was proposed in [26], where optimized control forces are added to a linearized dynamic model, and material properties are optimized to better match user constraints and provide plausible and consistent motion. Recently, Wang et al. [40] design linear deformation subspaces by minimizing quadratic deformation energy. Murai et al. [30] presents an algorithm that generates skin deformation (movement and jiggle) from joint angle data sequences by two steps: identification of parameters for a quasi-static muscle deformation model, and simulation of skin deformation. Roussellet et al. [33] proposes an efficient way to model muscle primitives with implicit surfaces. Inspired by [43] that uses a curve-defined model to animate skin deformation, Chaudhry et al. [5] integrate geometric transformations, example-based, and physics-based approaches to simulate dynamic skin deformations. Xu and Barbic [41] integrate physics-based and pose-space skin deformations, where the latter combines SSD [27] with artist-corrected pose shapes, and introduce pose-dependent model reduction to accelerate the finite element simulation for real-time applications, but more complicated implementation. Similar to [41] in spirit, this paper integrates SSD, example-based, and physics-based skin deformations to maximize their strengths.

## 3. Approach overview

Unlike numerical solutions that are essentially discrete representations, analytical solutions are continuous representations. In order to develop analytical solutions, discrete polygon models must be firstly transformed into continuous surfaces. Two mathematical representations can be used to define continuous surfaces: parametric and implicit surfaces. Since parametric surfaces are easier to develop analytical solutions than implicit surfaces, we select parametric surfaces to describe 3D skin surfaces.

Physics-based skin deformation approaches determine how skin deformations change with external forces. When external forces are



**Fig. 2.** The pipeline of our method consists of two parts, including Fourier series conversion and Physics-based skin deformation.

applied slowly, skin deformations can be treated as static (called static skin deformations), which does not consider the effects of acceleration (inertial forces) and velocity (damping forces) on skin shape changes. Otherwise, skin deformations can be treated as dynamic (called dynamic skin deformations) to include the influences of acceleration and velocity.

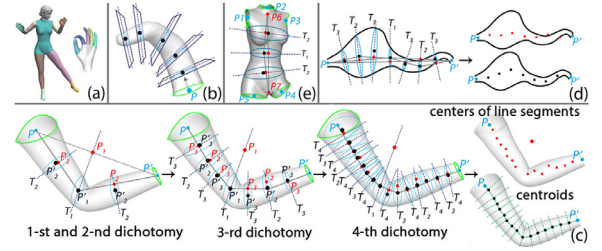
Parametric skin surfaces, written as  $S(u, v)$ , where  $u$  and  $v$  are two parametric variables.  $S_0(u, v)$  and  $S_1(u, v)$  stand for the meshes at any two adjacent poses called the starting pose and the ending pose, respectively, the shape change from  $S_0(u, v)$  to  $S_1(u, v)$  can be decomposed into two parts: *geometric deformation* that is resulted from affine transformations, and *physical deformation* that is due to the force field acting on the skin surface. With a time variable  $t$ , dynamic *physical deformation* can be written as  $D(u, v, t)$ , where a time-dependent force field  $F(u, v, t)$  is applied onto the skin surface, through the underlying physics described by differential equations. Assuming a differential operator  $L$  stands for the operations of the differential equations, the relationship between dynamic skin deformation and the force field can be described as  $L(D(u, v, t)) = F(u, v, t)$ , whose concrete form will be presented in Section 5.1.

Fourier series developed by the French mathematician Joseph Fourier provides a powerful tool for developing an analytical solution to differential equations. In order to develop an analytical solution to physics-based deformations, we will use Fourier series to represent the dynamic deformation  $D(u, v, t)$  and force field  $F(u, v, t)$  in Section 4.

As discussed above, the proposed approach consists of two parts: the first part, **Fourier series conversion** (Section 4), mainly including *identification of vertices on isoparametric curves* and *Fourier series representation*. The second part, **physics-based skin deformation** (Section 5), mainly including *mathematical model of physics-based skin deformation* and *analytical solver*, which derives a closed form analytical solution to physics-based skin deformation. The approach pipeline is illustrated in Fig. 2, two example meshes (a) and (b) are inputted to our system first in the **Fourier series conversion** part. Then, the vertices at the isoparametric curves are quickly identified to define the isoparametric curves in (c) and (f). After that, the LBS is introduced to deform (c) into (d), which together with (f) are transformed into continuous Fourier series representations indicated in (e) and (g), then inputted into a deformation calculator to obtain the physical deformation shown in (h). In the **Physics-based skin deformation** part, as shown in (i), a physics-based mathematical model is first developed for both dynamic and static skin deformations. Then, the force field causing skin deformation is represented with Fourier series. After that, the mathematical model is solved analytically and a closed form analytical solution is obtained. Finally, the obtained analytic solution is used to create skin deformations at any poses (j)–(l) between the starting and ending pose.

#### 4. Fourier series conversion

The first part, Fourier series conversion, is designed and tailored to avoid unnecessary manual operations, reduce the non-negligible amount of preprocessing time, and fast obtain the Fourier series rep-



**Fig. 3.** Determination of intersecting curves: (a) segmentation of character models into parts, (b) determination of intersecting curves for parts with one seam-curve, (c) for parts with two seam-curves, (d) for irregular parts with two seam-curves, and (e) for irregular parts with more than two seam-curves.  $T_i$  means the intersecting planes when  $i$ th dichotomy, the red points denotes centers of lines, and the black points denotes centroid of intersecting curves. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

resentations that are needed for follow-up physics-based deformation determination.

##### 4.1. Identification of isoparametric Curves

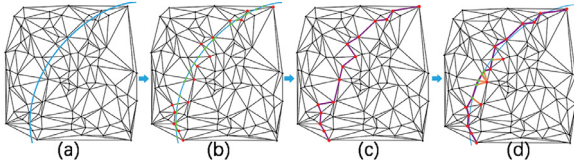
The identification of isoparametric curves includes: i) automatic segmentation of 3D models, ii) determination of the intersecting curves of the starting pose, and iii) extracting the isoparametric curves  $\{u = u_i\}$  according to intersecting curves at the starting pose, which will be used to determine their counterparts at the ending pose.

**Automatic segmentation.** Many existing algorithms can be used to automatically segment 3D mesh models. A comprehensive literature review on mesh segmentation algorithms can be found in [34]. Quantitative evaluation of mesh segmentation algorithms was also reported in [6]. Due to the reported efficiency of the random walks method [20], we employ it to automatically segment 3D models (specifically, 3D character models in this work). The intersecting curves between two adjacent segmented parts are called *seam-curves*. After segmenting a 3D model into parts, we obtain the segmented part models with one seam-curve (highlighted in green in Fig. 3(b)), two seam-curves (illustrated in Fig. 3(c), (d)), and more than two seam-curves (illustrated in Fig. 3(e)).

**Determination of intersecting curves.** The determination of intersecting curves on one part model can be divided into cases with one seam-curve, two seam-curves, and more than two seam-curves. For the case of one seam-curve on a part model (illustrated in Fig. 3(b)), we first calculate the centroid  $P$  (blue) of the seam-curve. From the centroid and the normal of the seam-curve, we specify a step size to determine next point to create intersecting plane that passes through the point and perpendicular to the normal, intersect the part model to obtain an intersecting curve. A new centroid is determined from this intersecting curve. The direction determined by the two adjacent centroids is taken to be the direction of next step. This process repeats until arriving at the farthest end of the part model.

For the case of two seam-curves on a part model (Fig. 3(c)), we use a dichotomy algorithm to obtain intersecting curves. For the first dichotomy, shown as Fig. 3(c), we determine the centroid  $P$  and  $P'$  of the two green seam-curves, and find the middle point  $P_1$  of the line segment  $PP'$ . From the red middle point  $P_1$ , an intersecting plane  $T_1$  perpendicular to the line  $PP'$  is created to intersect the part model, which leads to an intersecting curve (blue). Then, calculate the black centroid  $P'_1$  of the intersecting curve. For the second dichotomy, we connect the black centroid  $P'_1$  with  $P$  and  $P'$ , calculate the red middle points  $P_2$  of the line segments  $P'_1P$  and  $P'_1P'$ , create two planes  $T_2$  to intersect the model, obtain two intersecting curves (blue) and calculate two corresponding centroid  $P'_2$  of the blue curves. Repeat the iteration until reaching a specified number of iterations or the threshold of Euclidean distances between red center points and the black centroid. For the  $i$ th dichotomy, we can ob-





**Fig. 4.** Illustration of Extracting isoparametric curves: (a) shows one mesh intersected with one plane and get the intersected curve (blue). Green points in (b) are the intersected points, and the red vertex on each mesh edge is the edge end having smaller distance with green intersected points. Connecting these red ends of edges, isoparametric curve (purple) is extracted shown as (c). Improve the isoparametric curve by distance threshold, more regular isoparametric curve (purple) can be obtained shown as (d). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

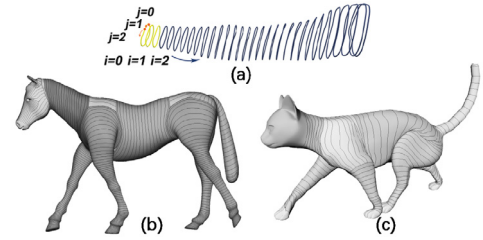
tain  $2^{(i-1)}$  centroid (black), connect all the black centroids after the  $i$ th dichotomy to generate a new curve skeleton shown as the last image of Fig. 3(c), use it to create new intersecting planes (green), which is more sensitive with character surface shape, then obtain new intersecting curves. Fig. 3(d) demonstrates the application of above dichotomy algorithm in one irregular model segmentation.

In the situation of more than two seam-curves on a part model as indicated in Fig. 3(e), calculate each centroid of green seam-curves, and separate them into two clusters using  $K$ -means clustering method. Among each cluster, we compute the average value of these seam-curve centers, named as  $P_6$  and  $P_7$ . After this treatment, we can transfer this case into two seam-curves problem (based on  $P_6$  and  $P_7$ ), and use the above dichotomy algorithm to obtain intersecting curves. The obtained intersecting curves in small regions near the seam-curves for the situation of more than two seam-curves on a part model may be closed or open. Since our proposed Fourier series representation and skin deformation algorithm are applicable to both closed and open curves, all obtained curves, whether closed or open, can be effectively and correctly processed. With the above three algorithms of determining intersecting curves, we can control the number of intersecting curves by taking different step sizes and effectively deal with irregular meshes with high frequency details and/or noise. *Extraction of isoparametric curves.* The blue intersecting curve in Fig. 4(a) is obtained by the aforementioned *Determination of intersecting curves*, consists of the green points in Fig. 4(b) which are intersecting points generated by the intersecting the edges of triangle facets. For each edge which is intersected by the intersecting plane, we calculate the distances from the intersecting point to the two end points of the edge, and take the end point with a smaller distance as an extracted vertex. Repeating this operation, we extract the red vertices in Fig. 4(b) and connect them together to get a purple isoparametric curve depicted in Fig. 4(c). As can be seen from Fig. 4(c), the extracted purple isoparametric curve is irregular when meshes are not regular. In order to obtain regular isoparametric curves, we introduce a threshold to discard some extracted vertices but keep green intersecting points as follows. We first calculate the distance from each of the extracted vertices to the intersecting curve. If the distance is larger than the threshold, the extracted vertex is discarded. The intersecting point(s) on the intersecting curve (green) are kept to generate the new isoparametric curve (purple), shown as Fig. 4(d).

Using the random walks method, various models are segmented into three types: segmentations with one seam-curve, segmentations with two seam-curves, and segmentations with more than two seam-curves, as illustrated in Fig. 5. We only need to determine the isoparametric curves of the starting pose. Afterward, the corresponding isoparametric curves on other meshes can be automatically determined by the corresponding vertices index. For any intersecting point, we use the same ratio to determine the intersecting point on the ending example. Fig. 6 shows the identified isoparametric curves of a human arm (a), a horse (b) and a cat (c). The extracted isoparametric curves on the surfaces of



**Fig. 5.** Segmentation results of representative models in the Princeton Segmentation Benchmark [6], including three types: blue portions denote one seam-curve segmentations, the yellow denotes two seam-curves segmentations, and the green denotes more than two seam-curves segmentations. It can be observed that our approach could be employed for general models. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



**Fig. 6.** Isoparametric curve identification algorithm transforms mesh into discrete points on isoparametric curves  $\tilde{S}_0(u_i, v_j)$  ( $i = 0, 1, 2, \dots, I$ ) and the discrete points on the  $i$ th isoparametric curve are numbered as  $\tilde{S}_0(u_i, v_j)$  ( $i = 0, 1, 2, \dots, I$ ) ( $j = 0, 1, 2, \dots, J$ ) in an arm model (a), a horse model (b), and a cat model (c).

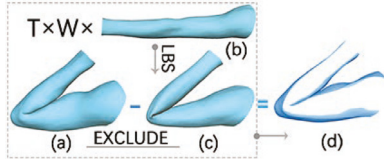
some models in Fig. 6 such as on the head and body near the legs of the horse model have been changed into open or other shapes to present visually pleasing curve networks and demonstrate our proposed Fourier series representation and skin deformation algorithm are applicable to all the curves in the curve networks. The extracted isoparametric curves on some surfaces of the models in Fig. 6 such as on the head and the body near the legs of the horse model have been changed into open or other shapes to present visually pleasing curve networks and demonstrate our proposed Fourier series representation and skin deformation algorithm are applicable to all the curves in the curve networks. Table 3 gives the computational times for identifying isoparametric curves. The method proposed in [3] can also be used to extract isoparametric curves from a triangular mesh. However, this method: 1) requires users to manually specify the extreme points of a harmonic function to guide the conversion; 2) its computational efficiency is less optimal. For example, it requires 6.21 s to process a mesh with 6966 triangles into its polar-annular mesh representation using 100 slices on an off-the-shelf computer [3]. Therefore, we propose the above approach to better serve our specific aim for Fourier Series Representation.

#### 4.2. Geometric Transformation

As discussed in Section 3, our physics-based skinning model does not include affine transformations such as rotation and translation, we must exclude these transformations first. Although various skinning methods such as LBS, PSD, and dual quaternion can be used to achieve this goal, we choose LBS to exclude these affine transformations due to its efficiency. The identified vertices  $\tilde{S}_0(u_i, v_j)$  at the starting pose are transformed to  $S_0(u_i, v_j)$  at the ending pose through a linear combination of bone transformation matrices  $T_m$  as follows.

$$S_0(u_i, v_j) = \left( \sum_{m=1}^M w_{ij,m} T_m \right) \tilde{S}_0(u_i, v_j) \quad (1)$$

where  $w_{ij,m}$  are the blending weights, and its subscript indicates the  $j$ th vertex on the isoparametric curve at  $u_i$ . Since we have two example poses, the transformation matrices  $T_m$  are straightforwardly fitted by transforming the animation skeleton at the starting pose to fit the



**Fig. 7.** Removal of transformations between two example poses. With the LBS model, we transform the starting pose (b) to obtain the transformed mesh in (c). The purely geometric transformations are excluded by calculating the difference between the mesh of the ending pose (a) and the transformed mesh (c), as shown in (d).

animation skeleton at the ending pose. We assume the skeletons have been given by users before hand, and among various weights calculation methods. Bounded Biharmonic Weights (BBW) [12] are smooth, shape-aware but need volume discretization. To maximize the high computational efficiency of this analytical solution, we use the method [42] to calculate  $w_{ij, m}$  in the above Eq. (1), and illustrate this computing process in Fig. 7.

#### 4.3. Fourier series representation

In order to obtain an analytical solution to physics-based skin deformations, we need to convert example meshes consisting of discrete vertices  $S_k(u_i, v_j)$  ( $k = 0, 1$ ) into continuous Fourier series representations  $C_k^F(u_i, v)$  ( $k = 0, 1$ ):

$$C_k^F(u_i, v) = c_0^k(u_i) + \sum_{n=1}^N [c_{2n-1}^k(u_i) \cos 2n\pi v + c_{2n}^k(u_i) \sin 2n\pi v] \quad (k = 0, 1) \quad (2)$$

where  $C_k^F(u_i, v)$  ( $k = 0, 1$ ) are continuous Fourier series representations of example models,  $k = 0$  denotes the starting pose,  $k = 1$  denotes the ending pose, the superscript  $F$  indicates Fourier series,  $u_i$  denotes the  $i$ th isoparametric curve in each pose,  $v$  is a variable in the range of  $[0, 1]$ ,  $c_n^k(u_i)$  ( $k = 0, 1; n = 0, 1, \dots, N$ ) are Fourier coefficients, and the subscript  $n$  denotes the  $n$ th term of the Fourier series. In order to determine the unknown Fourier coefficients  $c_n^k(u_i)$ , we first discuss how to relate the parametric variable  $v$  in Eq. (2) to the points on isoparametric curves. Assuming that the length of the isoparametric curves  $S_k(u_i, v_j)$  ( $k=0, 1$ ) from the point  $j = 0$  to the points  $j = 1, 2, \dots, J$  is  $L_{ij}^k$ , and the total length of the isoparametric curves from the point  $j = 0$  to the point  $j = J$  is  $L_{iJ}^k$  where the subscript  $J$  indicates the last point on the isoparametric curves, the value  $v_{ij}^k$  of the parametric variable  $v$  at the points  $S_k(u_i, v_j)$  ( $k=0, 1$ ) is determined by the ratio between the length  $L_{ij}^k$  and the total length  $L_{iJ}^k$ , i. e.,

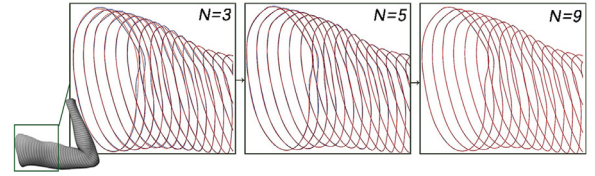
$$v_{i0}^k = 0 \quad v_{ij}^k = L_{ij}^k / L_{iJ}^k \quad (j = 1, 2, 3, \dots, J), \quad (3)$$

where

$$L_{im}^k = \sum_{l=1}^m d(S_k(u_i, v_l), S_k(u_i, v_{l-1})), \quad (k = 0, 1; i = 0, 1, 2, \dots, I; m = i, J; j = 1, 2, \dots, J) \quad (4)$$

where  $d$  is the Euclidean distance. After relating the parametric variable  $v$  in Eq. (2) to the points on the isoparametric curves, we obtain the points  $C_k^F(u_i, v_j)$  given by the Fourier series representations correspond to the points  $S_k(u_i, v_j)$  on the isoparametric curves. The error caused by the Fourier series representations can be quantified by the squared sum of all the Euclidean distances between the points  $C_k^F(u_i, v_j)$  and the points  $S_k(u_i, v_j)$  with the following equation:

$$E_k = \sum_{j=0}^{J-1} [d(C_k^F(u_i, v_j), S_k(u_i, v_j))]^2 \quad (5)$$



**Fig. 8.** Comparison between original isoparametric curves (blue) and Fourier series curves (red) (a). When different Fourier series terms are employed, such as  $N = 3, 5, 9$ , the Fourier series representation (red) from Eq. (2) are increasingly approaching to the isoparametric curves (blue) of the scanned arm. When  $N$  is set to 9, Fourier series representation approaches the isoparametric curves very closely. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

where

$$C_k^F(u_i, v_j) = c_0^k(u_i) + \sum_{n=1}^N [c_{2n-1}^k(u_i) \cos 2n\pi v_j + c_{2n}^k(u_i) \sin 2n\pi v_j] \quad (6)$$

We minimize Eq. (5) over the Fourier coefficients to determine the unknown Fourier coefficients  $c_n^k(u_i)$ .

The above Fourier series conversion algorithm is applicable to both closed and open isoparametric curves. For closed isoparametric curves,  $S_k(u_i, v_J) = S_k(u_i, v_0)$  and Eq. (5) can be used directly. For open isoparametric curves, we use  $C_k^F(u_i, v_0) = S_k(u_i, v_0)$  and  $C_k^F(u_i, v_J) = S_k(u_i, v_J)$  to achieve the positional continuity at the two end points and determine two vector-valued unknown Fourier coefficients in Eq. (2). If tangential or higher continuities at the two end points are required, the continuity conditions of the first or higher derivatives are introduced to determine more unknown constants in Eq. (2). For the positional continuity, we change  $j$  in Eq. (5) starting from 0 into 1 to determine the remaining unknown constants.

The comparisons between the isoparametric curves (blue) of a scanned arm and corresponding Fourier series curves (red) with different terms are shown in Fig. 8. They indicate that the proposed Fourier series representations describe local details well.

#### 4.4. Deformation Calculation

The deformations from the starting pose to ending pose excluding purely geometric transformations are the difference between the transformed starting pose  $C_0^F(u_i, v)$  and the ending pose  $C_1^F(u_i, v)$ , described below.

$$\tilde{D}(u_i, v) = C_1^F(u_i, v) - C_0^F(u_i, v) \quad (7)$$

Substituting Eq. (2) into Eq. (7) and putting the same coefficients together, the skin deformations  $\tilde{D}(u_i, v)$  can be rewritten as the following Fourier series:

$$\tilde{D}(u_i, v) = d_0(u_i) + \sum_{n=1}^N [d_{2n-1}(u_i) \cos 2n\pi v + d_{2n}(u_i) \sin 2n\pi v] \quad (8)$$

$$d_n(u_i) = c_n^1(u_i) - c_n^0(u_i) \quad (n = 0, 1, 2, 3, \dots, 2N) \quad (9)$$

where unknown constants  $c_n^0(u_i)$  and  $c_n^1(u_i)$  are obtained in Section 4.3.

#### 5. Physics-based skin deformation

Physics-based skin deformation determines how skin shapes change with external forces. There are two types of physics-based skin deformation techniques: the first type such as the deformable body dynamics in [14] directly uses a 3D manifold triangle mesh as a skin surface without considering the flesh. underneath the skin surface, and the second type uses more realistic anatomical models which includes the skin and flesh, such as the anatomically-inspired volumetric template model in

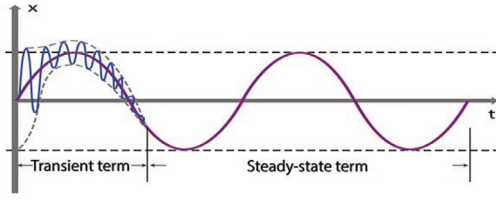


Fig. 9. Effects of damping and force on vibration [10].

[16] which consists of skin, underlying generic soft tissue, muscles, tendons, and skeleton. This paper follows the first type and only considers skin surfaces. In order to tackle both dynamic and static skin deformations, the physics-based skin deformation model should include the contributions of acceleration, velocity, and static skin deformation. In this section, we will propose such a physics-based skin deformation model to describe how skin surfaces change their shapes under the influence of a time-dependent force field. Since the deformation between the ending and starting poses has been represented in the Fourier space through Eq. (8), we also represent the force field in the same Fourier space to develop an analytical solution for creating realistic skin deformation efficiently.

### 5.1. Mathematical model of physics-based skin deformation

Similar with the treatment in [15], velocity and acceleration have effects on the shape change of skin surfaces. We formulate this effect by the vector-valued motion equation:

$$(\rho \partial^2 / \partial t^2 + \eta \partial / \partial t + \mathfrak{R}) \mathbf{D}(u, v, t) = \mathbf{F}(u, v, t) \quad (10)$$

where  $\rho$  and  $\eta$  are the density and damping coefficient of skin surfaces, and the corresponding terms describe the effects of acceleration and velocity on the shape variations of skin surfaces,  $\mathfrak{R}$  is an internal deformation force which is the function of skin deformations  $\mathbf{D}(u, v, t)$ , and  $\mathbf{F}(u, v, t)$  is a time-dependent force field driving the skin deformations. Skin deformations are similar to the elastic bending of thin plates. Accordingly, the internal force for skin bending deformations is similar to the internal force of the elastic bending of thin plates, and mathematically described by

$$\mathfrak{R} \mathbf{D}(u, v, t) = (\chi \partial^4 / \partial u^4 + \gamma \partial^4 / \partial u^2 \partial v^2 + \xi \partial^4 / \partial v^4) \mathbf{D}(u, v, t) \quad (11)$$

Here,  $\chi$ ,  $\gamma$ ,  $\xi$  are called shape control parameters which control the shape change of skin surfaces. Apart from the internal force for bending deformations, another force which resists tensile or compressive deformations should also be considered. This force can be regarded as a restoring force. Like the deformation of a spring, the restoring force is proportional to the tensile or compressive deformations, which can be approximated as  $\varsigma \mathbf{D}(u, v, t)$ . From the above discussion, the internal deformation force  $\mathfrak{R}$  should be the sum of the internal force for bending and the restoring force for tensile or compressive deformations. Using them to replace  $\mathfrak{R}$  in the linear elastodynamic Eq. (10) and introducing the differential operator  $\mathbf{L}$  defined below.

$$\mathbf{L} = \rho \partial^2 / \partial t^2 + \eta \partial / \partial t + \chi \partial^4 / \partial u^4 + \gamma \partial^4 / \partial u^2 \partial v^2 + \xi \partial^4 / \partial v^4 + \varsigma \quad (12)$$

The linear elastodynamic Eq. (10) is transformed into a vector-valued fourth order dynamic partial differential equation (PDE):

$$\mathbf{L} \mathbf{D}(u, v, t) = \mathbf{F}(u, v, t) \quad (13)$$

As show in Fig. 9, the forced vibrations include a transient part and a steady-state part. The transient part has an exponentially decreasing amplitude indicated by the blue curve. After sufficient time, the transient part is damped out, leaving the motion described by the steady-state part only indicated by the purple curve whose vibration frequency is totally determined by the frequency of the external forces [10]. Since the transient part only occurs in a short time we focus on the analytical

solution to the steady-state part of the physics-based skin deformation model in this paper. We propose to lower the fourth order dynamic PDE to the second order by using the following differential operator Eq. (14), it should be noted that lowering the dynamic PDE order may reduce the prediction accuracy of skin deformations. However, this can be well compensated by introducing example skin shapes.

$$\mathbf{L} = \rho \partial^2 / \partial t^2 + \eta \partial / \partial t + \chi \partial^2 / \partial u^2 + \gamma \partial^2 / \partial u \partial v + \xi \partial^2 / \partial v^2 + \varsigma \quad (14)$$

When considering the shape changes on the iso-parametric curve  $u_i$ , the parametric variable  $u$  becomes a constant. Accordingly, the deformation function  $\mathbf{D}(u, v, t)$  and the force functions  $\mathbf{F}(u, v, t)$  in Eq. (13) become  $\mathbf{D}(u_i, v, t)$  and  $\mathbf{F}(u_i, v, t)$ , the partial derivatives of  $\mathbf{D}(u, v, t)$  with respect to the parametric variable  $u$  drop, and the differential Eq. (13) becomes

$$\mathbf{L} \mathbf{D}(u_i, v, t) = \mathbf{F}(u_i, v, t), \quad (15)$$

where

$$\mathbf{L} = \rho \partial^2 / \partial t^2 + \eta \partial / \partial t + \xi \partial^2 / \partial v^2 + \varsigma \quad (16)$$

When two example meshes are known, we can set the deformations  $\mathbf{D}(u_i, v, t)$  to zero at  $t = 0$  and  $\tilde{\mathbf{D}}(u_i, v)$  at  $t = 1$ . If a mesh is motionless at  $t = 0$ , the deformation rate  $\partial \mathbf{D}(u_i, v, t) / \partial t$  is zero. If a mesh is in motion at  $t = 0$ , the deformation rate  $\partial \mathbf{D}(u_i, v, t) / \partial t$  should take the value of the deformation rate at the instant. If we use  $\dot{\mathbf{D}}_0$  to indicate the deformation rate at  $t = 0$ , these constraint conditions can be formulated as

$$t = 0 \quad \mathbf{D}(u_i, v, t) = 0, \quad \partial \mathbf{D}(u_i, v, t) / \partial t = \dot{\mathbf{D}}_0 t = 1 \quad \mathbf{D}(u_i, v, t) = \tilde{\mathbf{D}}(u_i, v) \quad (17)$$

After defining boundary conditions, the mathematical model is obtained by combining Eq. (15) with Eq. (17), which can be analytically solved and generate smooth deformation between two adjacent sequences as described below.

### 5.2. Fourier series force

The time-dependent force field  $\mathbf{F}$  in  $\mathbf{L} \mathbf{D} = \mathbf{F}$  can be determined from the given example skin shapes. At the starting pose  $t = 0$ , there is no skin deformations. The external force at the starting pose should be zero:  $\mathbf{F}(u_i, v, 0) = 0$ . At the ending pose  $t = 1$ , the skin deformations is  $\tilde{\mathbf{D}}(u_i, v)$  determined by Eq. (8), which are represented with Fourier series of the parametric variable  $v$ . Naturally, we can use the same Fourier series to represent the external force at the ending pose, i. e.,

$$\tilde{\mathbf{F}}(u_i, v) = \mathbf{e}_0(u_i) + \sum_{n=1}^N (\mathbf{e}_{2n-1}(u_i) \cos 2n\pi v + \mathbf{e}_{2n}(u_i) \sin 2n\pi v) \quad (18)$$

Here, the  $\mathbf{e}_0(u_i)$ ,  $\mathbf{e}_{2n-1}(u_i)$ ,  $\mathbf{e}_{2n}(u_i)$  are unknown constants that can be obtained by Analytic Solver in Section 5.3. From the known external forces at the starting and ending poses, we can obtain the following linear variation of the force field with respect to the time variable  $t$ .

$$\begin{aligned} \mathbf{F}(u_i, v, t) &= t \tilde{\mathbf{F}}(u_i, v) \\ &= t \mathbf{e}_0(u_i) + \sum_{n=1}^N (t \mathbf{e}_{2n-1}(u_i) \cos 2n\pi v + t \mathbf{e}_{2n}(u_i) \sin 2n\pi v) \end{aligned} \quad (19)$$

### 5.3. Analytic solver

The mathematical model Eq. (15) can be used to deal with dynamic skin deformations including skin vibrations. The time-dependent force field represented by Eq. (19) involves three types of terms:  $t$ ,  $t \cos 2n\pi v$ , and  $t \sin 2n\pi v$ . If it is introduced into Eq. (15),  $\mathbf{D}(u_i, v, t)$  should also contain the three types of terms. If we use  $\mathbf{D}_1(u_i, t)$ ,  $\mathbf{D}_{2n}(u_i, v, t) = \mathbf{D}_2(t \cos 2n\pi v)$  and  $\mathbf{D}_{3n}(u_i, v, t) = \mathbf{D}_3(t \sin 2n\pi v)$  to indicate the three types of terms, respectively, the deformation function  $\mathbf{D}(u_i, v, t)$  can be rewritten as

$$\mathbf{D}(u_i, v, t) = \mathbf{D}_1(u_i, t) + \sum_{n=1}^N [\mathbf{D}_{2n}(u_i, v, t) + \mathbf{D}_{3n}(u_i, v, t)] \quad (20)$$



Substituting Eq. (20) into Eq. (15), we can obtain

$$\mathbf{L}_1 \mathbf{D}_1(u_i, t) + \sum_{n=1}^N \mathbf{L}_2 [\mathbf{D}_{2n}(u_i, v, t) + \mathbf{D}_{3n}(u_i, v, t)] = \mathbf{F}(u_i, v, t), \quad (21)$$

where  $\mathbf{L}_1$  and  $\mathbf{L}_2$  are determined by

$$\mathbf{L}_1 = \rho \partial^2 / \partial t^2 + \eta \partial / \partial t + \varsigma \quad \mathbf{L}_2 = \mathbf{L}_1 + \xi \partial^2 / \partial v^2 \quad (22)$$

Introducing Eq. (19) into Eq. (21) and comparing both sides of the equation, the following equations are obtained

$$\begin{aligned} \mathbf{L}_1 \mathbf{D}_1(u_i, t) &= t \mathbf{e}_0(u_i) \\ \mathbf{L}_2 \mathbf{D}_{2n}(u_i, v, t) &= t \mathbf{e}_{2n-1}(u_i) \cos 2n\pi v \\ \mathbf{L}_2 \mathbf{D}_{3n}(u_i, v, t) &= t \mathbf{e}_{2n}(u_i) \sin 2n\pi v \\ (n &= 1, 2, 3, \dots, N) \end{aligned} \quad (23)$$

The remaining details of analytically solving the homogenous general and particular solutions to Eq. (21) are given in Appendix A. Finally, the analytical solution is as follows:

$$\begin{aligned} \mathbf{D}(u_i, v, t) &= \gamma_0(t) d_0(u_i) \\ &+ \sum_{n=1}^N \gamma_n(t) [d_{2n-1}(u_i) \cos 2n\pi v + d_{2n}(u_i) \sin 2n\pi v] \end{aligned} \quad (24)$$

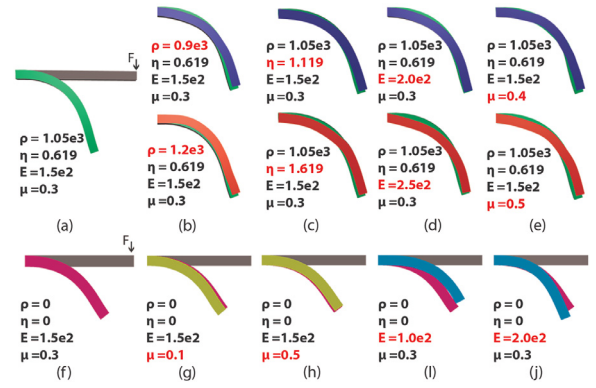
The above Eq. (24) indicates that once the skin deformations  $\mathbf{d}_n(u_i) (n = 0, 1, 2, \dots, 2N)$ , determined from the skin shapes at the starting pose  $\mathbf{c}_n^0(u_i)$  and the ending pose  $\mathbf{c}_n^1(u_i)$ , are known, the skin deformation at any in-between poses can be analytically determined by Eq. (24).

## 6. Experimental results and comparisons

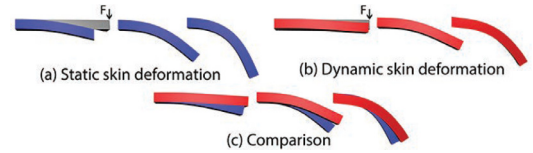
In Section 6.1, we first discuss some basic behaviors of the proposed physics-based skin deformations. In order to demonstrate our approach can produce realistic skin deformations efficiently, we compared our approach with geometric skin deformation methods (6.2), example-based skin deformation method, i.e., pose space deformation (PSD) method (6.3), and physics-based dynamic skin deformation methods (6.4 and 6.5), where 6.4 uses curve-defined skin models and 6.5 uses polygon skin models.

### 6.1. Basic behaviors

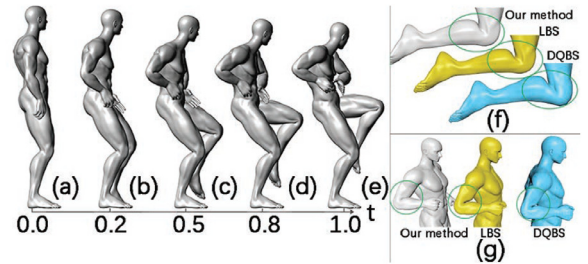
Basic behaviors to be tested include the effects of material and mechanical parameters, differences between dynamic and static skin deformations, and the reversibility of our approach. As aforementioned, skin deformations are similar to the elastic bending of thin plates and the bending deformations are described by Eq. (11), where the coefficient  $\xi$  is similar to the flexural rigidity of elastic plate bending, which is connected to Young's modulus  $E$ , Poisson ratio  $\mu$ , and skin thickness  $h$  through the equation  $\xi = Eh^3 / [12(1 - \mu^2)]$ . Young's modulus, Poisson ratio and the tensile stiffness  $\varsigma$  are mechanical parameters of the skin. The density  $\rho$  and damping coefficient  $\eta$  are material parameters. All of them have influence on skin shapes. We consider a skin strip with a rectangle cross-section of height  $h$  and unit width shown in Fig. 10(a) where the left end of the skin strip is fixed and the right end is under the action of a force  $F$ . According to [1,8], these mechanical and material parameters for the skin strip can be taken to be:  $E = 1.5e2$ ,  $\mu = 0.3$ ,  $\rho = 1.05e3$ , and  $\eta = 0.619$ . With these material and mechanical parameters, the straight skin strip in grey is deformed into a new shape in green shown in Fig. 10(a). Then we investigate how each parameter of  $E$ ,  $\mu$ ,  $\rho$  and  $\eta$  affect skin deformations, shown in Fig. 10(b)–(e). In Fig. 10(f),  $\rho = \eta = 0$  can achieve static skin deformations, and (f)–(j) demonstrate the effects of the Young's modulus  $E$  and the Poisson's ratio  $\mu$  on static skin deformations. Besides, in our physics-based skin deformation model, setting the density  $\rho = 0$  and the damping coefficient  $\eta = 0$ , the obtained analytical solution can be used to create static skin



**Fig. 10.** (a) shows the initial values for the proposed model, (b) shows how the performance of dynamic vibration when density  $\rho$  is decreased or increased; (c) shows how the damping coefficient  $\eta$  influences the results when overdamped; (d) shows the amplitude of the object decreases when the Young's modulus  $E$  is increased; (e) shows how Poisson ratio  $\mu$  affects the result. In (f),  $\rho = \eta = 0$  can achieve static skin deformation; (g)–(h) show the results when fixing the Young's modulus  $E$  but changing the Poisson's ratio  $\mu$ , (i)–(j) show the different results when fixing the Poisson's ratio  $\mu$  but changing the Young's modulus  $E$  and (f)–(j) demonstrate the effects of the Young's modulus  $E$  and the Poisson's ratio  $\mu$  on static skin deformations.



**Fig. 11.** (a) indicates static skin deformations by the proposed physics-based skin deformation model, (b) indicates dynamic skin deformations and (c) demonstrates substantial differences between the generated static and dynamic skin deformations.

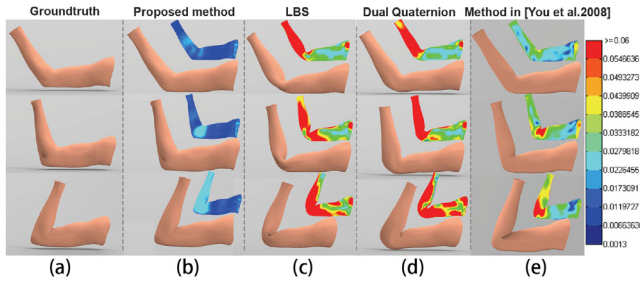


**Fig. 12.** Skin deformation generated by our approach. (a) and (e) are the input example poses. (b)–(d) are generated intermediate shapes by our approach. (f) and (g) shows our approach gives more realistic skin deformations at leg joints and arm joints.

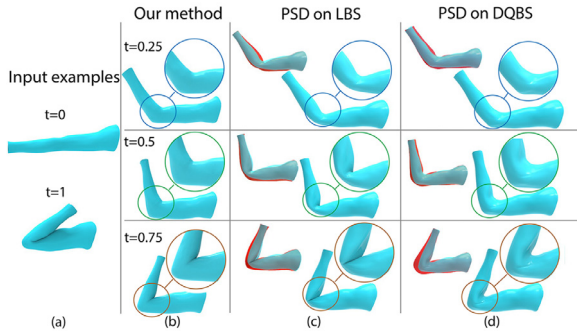
deformations. The differences between static and dynamic skin deformations are shown in Fig. 11 where Fig. 11(a) indicates static skin deformations, Fig. 11(b) indicates dynamic skin deformations, and Fig. 11(c) shows the differences between static and dynamic skin deformations.

### 6.2. Comparison with geometric skin deformation methods

In this subsection, we compared the deformed shapes created by our approach with 3D scanned groundtruth and those from baseline methods including the classical LBS and the Dual Quaternion blending skinning (DQBS) [17] which are included in standard character animation pipelines. Since LBS and DQBS have been included in Autodesk Maya, we use Maya's LBS and DQBS to determine skin deformations. Fig. 12(a)–(e) show the skin deformation result of a male model obtained by our approach. The comparison with those obtained by the two base-



**Fig. 13.** Comparisons between our method and baseline skinning methods. Three rows (from top to bottom) correspond to the poses at  $t = 0.25$ ,  $t = 0.5$ , and  $t = 0.75$ , respectively. The columns from left to right are: ground-truth from laser scanning data, our approach, the classical LBS, dual quaternion skinning [17], curve-based method [43]. The color images at top-right corners are the visualization of the vertex errors, compared to the ground-truth.

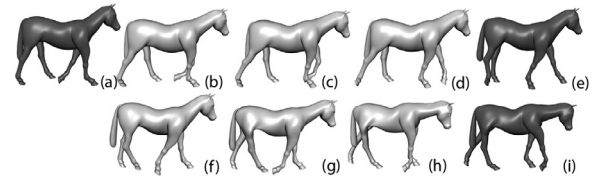


**Fig. 14.** Comparison with PSD, (a) shows the input data, (b) shows the models generated by our method, without collapsing-joint, bulging-joint and distorted normal. (c) shows the models create by PSD based on LBS. The left overlap parts show the differences between results of LBS (red) and PSD on LBS (blue). (d) shows the models generated by PSD based on DQBS. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

line methods is indicated in Fig. 12(f)–(g). Our approach gives more realistic skin deformations at the arm and the leg joints. In order to quantify the errors of various methods, we calculated the vertex errors between the 3D scanned groundtruth and those obtained by our approach and the two baseline methods, and depicted the vertex errors with different colours in Fig. 13. In the figure, our proposed method and the method by You et al. 2018 [43] use the ground-truth skin shapes at  $t = 0$  and  $t = 1$ , shown in Fig. 14(a), LBS and Dual Quaternion use the ground-truth skin shape at  $t = 0$ , also shown in Fig. 14(a), Fig. 13(b)–(e) show the calculated skin shapes at  $t = 0.25$ ,  $t = 0.5$ , and  $t = 0.75$  by our proposed method, LBS, Dual Quaternion, and the method by You et al. 2018 [43], respectively. As shown in the figure, the skin deformation by our approach is the closest to the ground-truth in this comparison. The visual perception of the difference between the ground-truth and our approach is negligible at most. In contrast, the meshes created by the classic LBS and dual quaternion blend skinning are noticeably different from the ground-truth. The Maximum Vertex Error (MVE) by our method is 0.026, while other methods have much larger MVEs: LBS (0.385), the dual quaternion blend skinning (0.278) and the method of You et al. 2018 [43] (0.169).

### 6.3. Comparison with example-based skin deformation methods

As above discussed, example-based skin deformation methods are often used together with geometric skinning, and realistic skin deformations require sufficient examples. Since the Pose Space Deformation (PSD) method [24] is a widely-used example-based skin deformation



**Fig. 15.** Skin deformation result of a horse model. (a), (e), and (i) are three input example poses, (b)–(d) are in-between poses generated by our method based on (a) and (e). (f)–(h) are in-between poses generated by our approach based on (e) and (i). During these two adjacent sequences, the shapes show a smooth transition from the first sequence (a)–(e) to the second sequence (e)–(i).

**Table 1**

Runtime comparison between this method and [43].

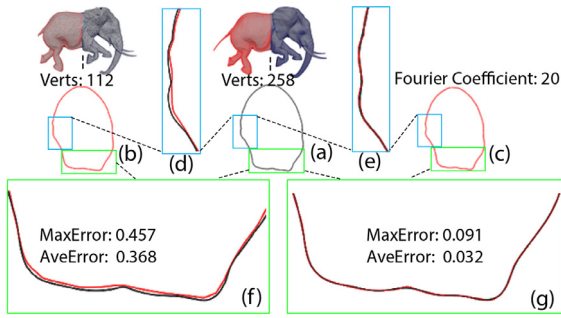
Data	Verts	Our method(fps)	[43] (fps)
Arm	3840	531	23
Cat	7207	273	17
Horse	10,128	205	14
Runner	32,185	86	4

method, we compare our approach with PSD in this subsection and demonstrate our proposed physics-based approach can be used to improve the realism of example-based skin deformations. All the three methods: our method, PSD on LBS, and PSD on DQBS, use the ground-truth arm skin shapes at  $t = 0$  and  $t = 1$  shown in Fig. 14(a). The obtained skin shapes at  $t = 0.25$ ,  $t = 0.5$  and  $t = 0.75$  are depicted in Fig. 14(b)–(d). In the figure, (b) is from our approach, and (c) and (d) are from the PSD tool added to Maya 2016 Extension 2. We can find that both PSD on LBS and PSD on DQBS improve the skin deformation results purely on LBS and DQBS, but still can not avoid the collapsing-joint, bulging-joint and distorted normal effects, whereas our approach creates more realistic skin deformations without these artifacts.

### 6.4. Comparison with physics-based skin deformation methods using curve defined models

Although several previous curve-based skin deformation methods [5,43] have been proposed, they have the following limitations. 1) The curves defining skin surfaces are manually extracted, leading to heavy and time-consuming human involvements. 2) A large linear system must be solved to obtain skin deformations at finite difference nodes [5] or unknown constants involved in analytical mathematical expressions of skin deformations [43]. 3) The animation is not smooth at the connecting poses between adjacent animation sequences. The above limitations 1) to 2) make previous curve-based skin deformation methods less suitable for those applications requiring high animation frame rates, and the limitation 3) would seriously affects the quality of resulting animations. Our approach overcomes all the above three limitations. Since both deformation continuity and deformation rate continuity are introduced into Eq. (17), our method only requires at least two example meshes to produce smooth and physically realistic animations as demonstrated by the horse model in Fig. 15 and the accompanying video. In contrast, the previous approaches such as [43] would fail to produce smooth animations if multiple example poses are given, as demonstrated by the animation result in the accompanying video. Among various existing curve-based skin deformation approaches, the work of [43] is the most efficient. We compare the computing time of our approach with [43] through four different models: scanned arm, cat, horse, and human, presented in Table 1 which shows that our approach is about 12–18 times faster than [43].





**Fig. 16.** The high-resolution curve (a) in polygon mesh reduced to (b) with large accuracy lose in (d), (f). But Fourier series representation (c) keep high realism with sharply decreasing data-size.

**Table 2**

Comparison of variables, FC: Fourier Coefficients Terms. The second row shows the total vertices of character models. The third to fifth rows show the total vector-valued Fourier coefficients when using different Fourier series terms to describe the models.

	Arm	Cat	Horse	Dancer	Runner
Verts	3840	7207	10,128	13,201	32,185
FC $N = 3$	448	1681	2363	3080	5632
FC $N = 5$	704	2642	3713	4840	8850
FC $N = 9$	1216	4563	6414	8360	15,287

**Table 3**

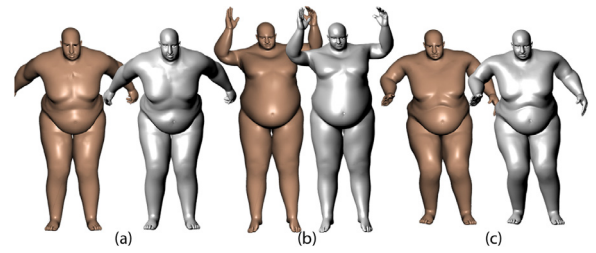
Runtime breakdown of our approach. FC: Fourier Coefficients, variables of Fourier series representations used in these experiments. IIL: Identification of isoparametric curves. GT: Geometric Transform. FSC: Fourier Series Representation. DC: Deformation Calculation. AS: Analytic Solver. SD: Skin Deformer.

Model	FC	Runtime (ms)				
		IIL	GT	FSR+DC	AS+SD	Total
Arm	1216	0.35	0.215	1.19	0.13	1.885
Cat	2642	0.72	0.258	2.47	0.21	3.658
Horse	3713	0.99	0.450	3.15	0.27	4.860
Runner	15,287	1.83	0.656	8.25	0.79	11.526

### 6.5. Comparison with physics-based skin deformation methods using polygon models

As discussed before, physics-based skin deformation methods produce high-quality and realistic results, but require heavy numerical calculations. Our method significantly reduces the computing cost and manual operations. In this subsection, we will compare our proposed approach with existing physics-based skin deformation methods using polygon models. We first demonstrate our proposed Fourier series representation can greatly reduce design variables compared with polygon models but still keep high-realism, shown as Fig. 16. Table 2 gives a comparison of design variables. For polygon models, the total design variable is the total number of vertices. After representing polygon models with Fourier series, the total design variables become the total vector-valued Fourier coefficients. The data illustrate the design variables of polygon models are 4.3–8.6, 2.7–5.5, and 1.6–3.2 times of Fourier series representations, when FC set to be 3, 5, and 9, respectively. Using Fourier series representations, it reduce design variables for calculation sharply, and the specific value of Fourier coefficients used in our experiments are shown in Table 3.

Among various physics-based skin deformation approaches, the approach proposed in [41] incorporates physically based simulation into rigging/skinning to automatically produce secondary skin motion and



**Fig. 17.** Simulating skin vibration with steady-state dynamics: (a)–(c) are comparisons of three model poses created by ABAQUS steady-state dynamic simulation (gold) and by our steady-state dynamic simulation (silver).

use model reduction to accelerate numerical calculations, which sacrifices the accuracy to a certain extent. It should be noted that, since our method focuses on integrating example skin shapes to physics-based model to create high-realistic skin deformation in the steady-state part of forced vibrations. By contrast, the work described in [41] includes the calculations of vibration modes under constraints. Therefore, it is unable to make a fair comparison between our method and [41]. To accommodate large deformations around each pose, [41] augments the linear model basis with modal derivatives proposed in [4]. In [4], after frame 1000, the vertical displacement of a vertex of a spoon's simulation between full simulation and modal derivatives could reach about half of the length of the spoon. In order to obtain elastic and inertial forces, [41] needs several samples for training, which brings in fewer than 3% relative training error. Although the pose-dependent model reduction [41] divides a polar bear model of 6876 vertices into 8 local regions and uses 8 threads with OpenMP, it still takes 1.43 ms to animate skin deformations of the polar bear model. Without any multithreading, our proposed approach only takes 1.627 ms to animate skin deformations of a horse model with 10,128 vertices. It indicates that our approach is comparable with the pose-dependent model reduction, if not faster.

The method in [18] generates higher quality deformations than both linear and dual quaternion skinning through skinning weights optimization, but requires at least a few minutes to precompute the deformation weights. The method in [28] needs at least several seconds for torso and arms simulation per frame on GPU, it is still not fast enough for interactive posing. Through a quantitative comparison with ground-truth skin deformations shown in Fig. 13, the high realism of our approach has been clearly demonstrated.

Using shell elements and the same example skin shapes at extreme poses, we also compared with the finite element simulation of steady-state skin vibrations, shown in Fig. 17, which further shows a high realism achieved by our approach.

## 7. Discussion and conclusion

This paper develops an analytical approach to efficiently create physically-realistic skin deformations. It includes vertex identification on isoparametric curves, Fourier series conversion, and an analytic solution to the formulated physics-based skin deformation model. Many experiments have been conducted to validate this approach. We discussed the effects of various factors, and made direct comparisons with various skin deformation approaches including geometric skin deformation methods, example-based skin deformation methods, and physics-based skin deformation methods. These results demonstrate our approach avoids unnecessary manual operations and time-consuming numerical calculation that require specialized knowledge and skills, produces physically-realistic models and reaches high animation frame rates for real-time applications. Several improvements can be made in the future. First, we will extend our method to develop an analytical solution to fourth order differential equations to tackle elastic dynamic skin deformations including both transient and steady-state skin vibrations. Our current analytical solution to physics-based skin deformation

is derived from linearly varying forces. It can be extended to nonlinearly varying forces. We will examine which force variation can generate more realistic skin deformations in the future work.

## Declaration of Competing Interest

None.

## Acknowledgment

This research is supported by the PDE-GIR project which has received funding from the [European Union's Horizon 2020](#) research and innovation programme under the Marie Skłodowska-Curie grant agreement (No. 778035), the [National Natural Science Foundation of China](#) (Grant No. 51475394), and Innovate UK (Knowledge Transfer Partnerships KTP.010860). Shaojun Bian is also supported by China Scholarship Council. Xiaogang Jin is supported by the Key Research and Development Program of Zhejiang Province, China (No. 2018C01090) and the [National Natural Science Foundation of China](#) (No. 61732015).

## Appendix A

The homogenous general solution to each equation of [Eq. \(23\)](#) can be taken as  $\tilde{\mathbf{D}}_1(u_i, t) = e^{r_1 t}$ ,  $\tilde{\mathbf{D}}_2(u_i, v, t) = e^{q_1 t} \cos 2n\pi v$  and  $\tilde{\mathbf{D}}_3(u_i, v, t) = e^{s_1 t} \sin 2n\pi v$ . Substituting them into each of [Eq. \(23\)](#), considering the situations  $4\rho\xi - \eta^2 > 0$  and  $4(\xi - 4\zeta n^2\pi^2) - \eta^2 > 0$ , and setting

$$\begin{aligned} \alpha_r &= -\eta/(2\rho) & \beta_r &= \alpha_r \sqrt{4\rho\xi/\eta^2 - 1} \\ h_n &= \zeta - 4n^2\pi^2\xi & \beta_n &= \alpha_r \sqrt{4h_n/\eta^2 - 1} \end{aligned} \quad (\text{A.1})$$

we obtain  $r_{1,2} = \alpha_r + i\beta_r$ ,  $q_{n1,2} = s_{n1,2} = \alpha_r + i\beta_n$  and the homogenous general solutions below

$$\begin{aligned} \tilde{\mathbf{D}}_1(u_i, t) &= \mathbf{C}_1(u_i)f(t) + \mathbf{C}_2(u_i)\bar{f}(t) \\ \tilde{\mathbf{D}}_{2n}(u_i, v, t) &= [\mathbf{C}_{3n}(u_i)g_n(t) + \mathbf{C}_{4n}(u_i)\bar{g}_n(t)] \cos 2n\pi v \\ \tilde{\mathbf{D}}_{3n}(u_i, v, t) &= [\mathbf{C}_{5n}(u_i)g_n(t) + \mathbf{C}_{6n}(u_i)\bar{g}_n(t)] \sin 2n\pi v \\ f(t) &= e^{\alpha_r t} \cos \beta_r t & \bar{f}(t) &= e^{\alpha_r t} \sin \beta_r t \\ g_n(t) &= e^{\alpha_r t} \cos \beta_n t & \bar{g}_n(t) &= e^{\alpha_r t} \sin \beta_n t \end{aligned} \quad (\text{A.2})$$

Then, the particular solutions to [Eq. \(23\)](#) can be taken as  $\tilde{\mathbf{D}}_1(u_i, t) = a_1(u_i) + a_2(u_i)t$ ,  $\tilde{\mathbf{D}}_{2n}(u_i, v, t) = [a_3(u_i) + a_4(u_i)t] \cos 2n\pi v$ ,  $\tilde{\mathbf{D}}_{3n}(u_i, v, t) = [a_5(u_i) + a_6(u_i)t] \sin 2n\pi v$ . Substituting them into [Eq. \(23\)](#) to obtain

$$\begin{aligned} \tilde{\mathbf{D}}_1(u_i, t) &= (\zeta t - \eta)e_0(u_i)/\zeta^2 \\ \tilde{\mathbf{D}}_{2n}(u_i, v, t) &= (h_n t - \eta)e_{2n-1}(u_i) \cos 2n\pi v/h_n^2 \\ \tilde{\mathbf{D}}_{3n}(u_i, v, t) &= (h_n t - \eta)e_{2n}(u_i) \sin 2n\pi v/h_n^2 \end{aligned} \quad (\text{A.3})$$

Introducing [Eq. \(A.2\)](#) and [Eq. \(A.3\)](#) into [Eq. \(20\)](#), we obtain the function  $\mathbf{D}(u_i, v, t)$  which involves the unknown constants  $\mathbf{e}_0(u_i)$ ,  $\mathbf{e}_{2n-1}(u_i)$ ,  $\mathbf{e}_{2n}(u_i)$  and  $\mathbf{C}_1(u_i)$ ,  $\mathbf{C}_2(u_i)$  and  $\mathbf{C}_{kn}(u_i)$  ( $k = 3, 4, 5, 6$ ). Using the boundary conditions [Eq. \(17\)](#), we could determine these unknown constants. Then, substituting the obtained unknown constants back into  $\mathbf{D}(u_i, v, t)$ , and introducing

$$\begin{aligned} \delta_0 &= e^{\alpha_r t} [-\eta\beta_r \cos \beta_r + \zeta(1 + \eta\alpha_r/\zeta) \sin \beta_r] - \zeta\beta_r(1 - \eta/\zeta) \\ \delta_n &= e^{\alpha_r t} [-\eta\beta_n \cos \beta_n + (1 + \eta\alpha_r) \sin \beta_n] - \beta_n(1 - \eta/h_n) \\ \gamma_0(t) &= [-\eta\beta_r f(t) + \zeta(1 + \eta\alpha_r/\zeta)\bar{f}(t) - \zeta\beta_r(t - \eta/\zeta)]/\delta_0 \\ \gamma_n(t) &= [-\eta\beta_n g_n(t) + (1 + \eta\alpha_r)\bar{g}_n(t) - \beta_n(t - \eta/h_n)]/\delta_n \end{aligned} \quad (\text{A.4})$$

we obtain [Eq. \(24\)](#).

## Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:[10.1016/j.gmod.2019.101035](https://doi.org/10.1016/j.gmod.2019.101035).

## References

- [1] M.R. Amar, Estimation of mechanical properties of soft tissue subjected to dynamic impact, *Ind. Manage. Syst. Eng.* 12 (2010) 1–67.
- [2] A. Angelidis, K. Singh, Kinodynamic skinning using volume-preserving deformations, in: *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, in: SCA '07, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 2007, pp. 129–140.
- [3] J.A. Bærentzen, R. Abdrashitov, K. Singh, Interactive shape modeling using a skeleton-mesh co-representation, *ACM Trans. Graph. (TOG)* 33 (4) (2014) 132.
- [4] J. Barbič, D.L. James, Real-time subspace integration for St. Venant-Kirchhoff deformable models, in: *ACM Transactions on Graphics (TOG)*, vol. 24, ACM, 2005, pp. 982–990.
- [5] E. Chaudhry, S. Bian, H. Ugail, X. Jin, L. You, J.J. Zhang, Dynamic skin deformation using finite difference solutions for character animation, *Comput. Graph.* 46 (2015) 294–305.
- [6] X. Chen, A. Golovinskiy, T. Funkhouser, A benchmark for 3d mesh segmentation, in: *ACM Transactions on Graphics (TOG)*, vol. 28, ACM, 2009, p. 73.
- [7] C. Deul, J. Bender, Physically-based character skinning, *Virtual Reality Interactions and Physical Simulations (VRIPhys)*, 2013.
- [8] G. Elert, *The Physics Hypertextbook*, 1998.
- [9] L. Gao, Y.-K. Lai, D. Liang, S.-Y. Chen, S. Xia, Efficient and flexible deformation representation for data-driven surface modeling, *ACM Trans. Graph. (TOG)* 35 (5) (2016) 158.
- [10] G.W. Housner, D.E. Hudson, *Applied Mechanics Dynamics*, 1980.
- [11] A. Jacobson, I. Baran, L. Kavan, J. Popović, O. Sorkine, Fast automatic skinning transformations, *ACM Trans. Graph. (TOG)* 31 (4) (2012) 77.
- [12] A. Jacobson, I. Baran, J. Popović, O. Sorkine, Bounded biharmonic weights for real-time deformation, *ACM Trans. Graph. (TOG)* 30 (4) (2011) 78.
- [13] A. Jacobson, O. Sorkine, Stretchable and twistable bones for skeletal shape deformation, *ACM Trans. Graph. (TOG)* 30 (6) (2011) 165.
- [14] S. Jain, C.K. Liu, Controlling physics-based characters using soft contacts, *ACM Trans. Graph. (TOG)* 30 (6) (2011) 163.
- [15] D.L. James, D.K. Pai, DyRT: dynamic response textures for real time deformation simulation with graphics hardware, *ACM Trans. Graph. (TOG)* 21 (3) (2002) 582–585.
- [16] P. Kadlecěk, A.-E. Ichim, T. Liu, J. Křivánek, L. Kavan, Reconstructing personalized anatomical models for physics-based body animation, *ACM Trans. Graph. (TOG)* 35 (6) (2016) 213.
- [17] L. Kavan, S. Collins, J. Žára, C. O'Sullivan, Geometric skinning with approximate dual quaternion blending, *ACM Trans. Graph. (TOG)* 27 (4) (2008) 105.
- [18] L. Kavan, O. Sorkine, Elasticity-inspired deformers for character articulation, *ACM Trans. Graph. (TOG)* 31 (6) (2012) 196.
- [19] P.G. Kry, D.L. James, D.K. Pai, EigenSkin: real time large deformation character skinning in hardware, in: *Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, in: SCA '02, ACM, New York, NY, USA, 2002, pp. 153–159.
- [20] Y.-K. Lai, S.-M. Hu, R.R. Martin, P.L. Rosin, Fast mesh segmentation using random walks, in: *Proceedings of the 2008 ACM Symposium on Solid and Physical Modeling*, ACM, 2008, pp. 183–191.
- [21] B.H. Le, Z. Deng, Smooth skinning decomposition with rigid bones, *ACM Trans. Graph. (TOG)* 31 (6) (2012) 199.
- [22] B.H. Le, J.K. Hodgins, Real-time skeletal skinning with optimized centers of rotation, *ACM Trans. Graph. (TOG)* 35 (4) (2016) 37.
- [23] S.-H. Lee, E. Sifakis, D. Terzopoulos, Comprehensive biomechanical modeling and simulation of the upper body, *ACM Trans. Graph. (TOG)* 28 (4) (2009) 99.
- [24] J.P. Lewis, M. Cordner, 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*, ACM Press/Addison-Wesley Publishing Co., 2000, pp. 165–172.
- [25] D. Li, S. Sueda, D.R. Neog, D.K. Pai, Thin skin elastodynamics, *ACM Trans. Graph. (TOG)* 32 (4) (2013) 49.
- [26] S. Li, J. Huang, F. de Goes, X. Jin, H. Bao, M. Desbrun, Space-time editing of elastic motion through material optimization and reduction, *ACM Trans. Graph. (TOG)* 33 (4) (2014) 108.
- [27] N. Magnenat-Thalmann, R. Laperrière, D. Thalmann, Joint-dependent local deformations for hand animation and object grasping, in: *Proceedings on Graphics Interface '88*, Citeseer, 1988.
- [28] A. McAdams, Y. Zhu, A. Selle, M. Empey, R. Tamstorf, J. Teran, E. Sifakis, Efficient elasticity for character skinning with contact and collisions, in: *ACM Transactions on Graphics (TOG)*, vol. 30, ACM, 2011, p. 37.
- [29] A. Mohr, M. Gleicher, Building efficient, accurate character skins from examples, in: *ACM Transactions on Graphics (TOG)*, vol. 22, ACM, 2003, pp. 562–568.
- [30] A. Murai, Q.Y. Hong, K. Yamane, J.K. Hodgins, Dynamic skin deformation simulation using musculoskeletal model and soft tissue dynamics, *Comput. Visual Media* 3 (1) (2017) 49–60.
- [31] S.I. Park, J.K. Hodgins, Data-driven modeling of skin and muscle deformation, in: *ACM Transactions on Graphics (TOG)*, vol. 27, ACM, 2008, p. 96.
- [32] G. Pons-Moll, J. Romero, N. Mahmood, M.J. Black, Dyna: a model of dynamic human shape in motion, *ACM Trans. Graph. (TOG)* 34 (4) (2015) 120.

- [33] V. Roussellet, N.A. Rumman, F. Canezin, N. Mellado, L. Kavan, L. Barthe, Dynamic implicit muscles for character skinning, *Comput. Graph.* 77 (2018) 227–239.
- [34] A. Shamir, A survey on mesh segmentation techniques, in: *Computer Graphics Forum*, vol. 27, Wiley Online Library, 2008, pp. 1539–1556.
- [35] P.-P.J. Sloan, C.F. Rose III, M.F. Cohen, Shape by example, in: *Proceedings of the 2001 Symposium on Interactive 3D Graphics*, ACM, 2001, pp. 135–143.
- [36] Y. Teng, M.A. Otaduy, T. Kim, Simulating articulated subspace self-contact, *ACM Trans. Graph. (TOG)* 33 (4) (2014) 106.
- [37] R. Vaillant, L. Barthe, G. Guennebaud, M.-P. Cani, D. Rohmer, B. Wyvill, O. Gourmel, M. Paulin, Implicit skinning: real-time skin deformation with contact modeling, *ACM Trans. Graph. (TOG)* 32 (4) (2013) 125.
- [38] R. Vaillant, G. Guennebaud, L. Barthe, B. Wyvill, M.-P. Cani, Robust iso-surface tracking for interactive character skinning, *ACM Trans. Graph. (TOG)* 33 (6) (2014) 189.
- [39] K. Wampler, Fast and reliable example-based mesh IK for stylized deformations, *ACM Trans. Graph. (TOG)* 35 (6) (2016) 235.
- [40] Y. Wang, A. Jacobson, J. Barbič, L. Kavan, Linear subspace design for real-time shape deformation, *ACM Trans. Graph. (TOG)* 34 (4) (2015) 57.
- [41] H. Xu, J. Barbič, Pose-space subspace dynamics, *ACM Trans. Graph. (TOG)* 35 (4) (2016) 35.
- [42] X. Yang, J.J. Zhang, Stretch it-realistic smooth skinning, in: *Computer Graphics, Imaging and Visualisation, 2006 International Conference on*, IEEE, 2006, pp. 323–328.
- [43] L. You, X. Yang, J.J. Zhang, Dynamic skin deformation with characteristic curves, *Comput. Anim. Virtual Worlds* 19 (3–4) (2008) 433–444.
- [44] H. Zhao, N. Lei, X. Li, P. Zeng, K. Xu, X. Gu, Robust edge-preserving surface mesh polycube deformation, *Comput. Visual Media* 4 (1) (2018) 33–42.