

Efficient and Physics-based Facial Blendshapes based on ODE sweeping Surface and Newton's second law

Junheng Fang

The National Center for Computer
Animation

Bournemouth University
Bournemouth, UK

jfang@bournemouth.ac.uk

Shaojun Bian

The National Center for Computer
Animation

Bournemouth University,
Bournemouth, UK

sbianbian@163.com

Jon Macey

The National Center for Computer
Animation

Bournemouth University,
Bournemouth, UK

jmacey@bournemouth.ac.uk

Andres Iglesias

Department of Applied Mathematics
and Computational Sciences

University of Cantabria
Cantabria, Spain

iglesias@unican.es

Hassan Ugail

The Centre for Visual Computing
University of Bradford

Bradford, UK

h.ugail@bradford.ac.uk

Alexander Malyshev

Department of Mathematics
University of Bergen

Norway

alexander.malyshev@uib.no

Ehtaz Chaudhry

The National Center for Computer
Animation

Bournemouth University
Bournemouth, UK

echaudhry@bournemouth.ac.uk

Lihua You

The National Center for Computer
Animation

Bournemouth University
Bournemouth, UK

L.You@bournemouth.ac.uk

Jian Jun Zhang

The National Center for Computer
Animation

Bournemouth University
Bournemouth, UK

jjzhang@bournemouth.ac.uk

Abstract—Online games require small data of 3D models for low storage costs, quick transmission over the Internet, and efficient geometric processing to achieve real-time performance, and new techniques of facial blendshapes to create natural facial animation. Current geometric modelling and animation techniques involve big data of geometric models and widely applied facial animation using linear interpolation cannot generate natural facial animation and create special facial animation effects. In this paper, we propose a new approach to integrate the strengths of ODE (ordinary differential equation) sweeping surfaces and Newton's second law-based facial blendshapes to create 3D models and their animation with small data, high efficiency, and ability to create special facial effects.

Keywords—wireframe extraction; ODE sweeping surface-represented 3D models; Newton's second law-based facial blendshapes

I. INTRODUCTION

Recently, the two most significant and interlinked challenges, which limit the development of game production, are geometric modelling and computer animation. As the game industry focuses more and more on scene realism and performance, the game capacity has shown geometric multiplier growth especially after the 3D model techniques have been widely used. For instance, *Super Mario Bros.*, released in 1985, spent only 64 KB to store the whole game. After over 10 years, *Super Mario Bros. Deluxe*, released in platform GBC in 1999, took 342 KB, which is only approximately 6 times the size of the former game, remaining the 2D scene. However, when it came to the 21st century, as the 3D models are introduced into the game industry, the size of *New Super Mario Bros.*, released in platform NDS in 2006, came to 10 MB, which is nearly 30

times compared with the 1999's one. Markedly, the size surged 250 times i.e. 2.5 GB in 2012 released *New Super Mario Bros. U*, with both the scene and models using 3D techniques.

As is widely agreed, to achieve accurate and fine models and natural animation, rich details are required for good realism, which involves abundant design parameters, i.e. surface vertices of polygon models.

Nevertheless, problems are deservedly brought out when managing those massive data, causing a large increase of the time used in shape manipulation, animation disposing and rendering. Besides, transmitting is also an important issue of online games. As the large data of geometric models or rendered images transmit among the game players over the Internet, the transmission time is noticeably increased, and the real-time performance of game playing is reduced due to the limitation of the communication technology. Apart from the above problems, the size of the current games, with high storage cost, has placed a great burden on the hard drives, like video game *Call of Duty: Infinite Warfare* which uses over 130 GB of storage to maintain its game, while the average capacity of the mainstream SSD is only 128 GB. Sufficient example models with good realism and abundant details are required by advanced games based on data-driven techniques, which spend heavy human involvement and high cost and cause inefficient game production.

As for animation, if it follows the underlying physics of object movements and de-formations, the requirement of high computational resources and much computational time, which would directly affect the performance of real-time games, is inevitable. These are all challenges for techniques related to real-time animation, including modelling and skin deformation.

The widely used surface creation approaches are polygon, patch modelling like subdivision and NURBS. The polygon technique manipulates surface vertices of simple geometric primitives to generate complex models. The patch technique divides the complicated models into plenty of simple patches and after processing them separately, combines them together to fulfil the model. The subdivision technique separates the polygonal faces into smaller pieces by approximating or interpolating schemes to build a denser mesh of the model. However, these techniques require further improvements, for example how to reduce tedious and time-consuming manual operations when stitching separated patches together as well as considering continuity between adjacent ones, how to manipulate the global shape of complicated patches, and how to apply physics of object deformations in the recreated model.

Both the academia and industry practices in the animation field regard skin de-formation techniques as the standardized approaches when forming real-time animation. Among all the different skin deformation methods, the major characteristics have not been changed, realism and efficiency. To achieve realism, more and more details are required to be stored, i.e. sufficient vertices, which causes plenty of redundancy that is not used often in computation. Linear interpolation is still popular in facial blendshapes and determining skin deformations. Physics-based techniques cannot avoid numerical computations, which costs high sources of CPU, and reduces efficiency.

Therefore, how to solve the above challenges has become an urgent obstacle in developing advanced game techniques. In order to address these challenges, this paper will: (1) extract a 3D surface model into a wireframe-defined one to reduce data size of 3D models, (2) use an ordinary differential equation (ODE)-based method to recreate 3D models with small data from the extracted wireframes, (3) propose a new and physics-based technique of facial blendshapes based on the Newton's second law to generate new wireframes from existing ones, and (4) combine the ODE sweeping surface-represented 3D models with the physics-based facial blendshapes to create natural facial animation efficiently.

The remaining parts of this paper consist of five sections. Existing work is reviewed in Section II. The theory and method of creating ODE sweeping surface-represented 3D models are introduced in Section III. A new, efficient, and the physics-based algorithm of facial blendshapes is examined in Section IV. Experiments are made in Section V. Conclusions and future work are presented in Section VI.

II. RELATED WORK

For commercially available graphics package, the widely recognized method to build models is polygon modelling and skin surface creation, which could exhibit details and branches, as well as assign UV texture coordinates, given in [1]. Nevertheless, it is difficult to create curved surfaces. Instead, approximating curved surfaces with many small flat planes is used to visualize them, which reduces the precision. Though polygon modelling is more easily to generate hard edges, NURBS modelling could create curved objects with smoother surfaces and more readily edit the patches with fewer design parameters, investigated in [2]. This advantage makes it welcome in building realistic models. However, stitching

adjacent patches requires tedious manual operations to tackle the continuity problem. Another method to build models is subdivision modelling. It first creates a rough polygonal model, and then uses approximating or interpolation to subdivide each face into smaller ones to finally get a denser mesh. It could easily create complicated geometry with more efficiently rendering, whilst the insufficient underlying parametrization makes it hard to elaborate the accuracy. Besides, Várady et al. have declared a method based on in-curved network for further shape controls, which is particularly used to revise the inside of transfinite patches in [3].

For generating more realistic models, physics-based modelling is provided with the physical laws underlying its surface deformation. Nealen et al. have made a review of multifarious physics-based modelling methods in [4], including finite element method, finite difference method, mass-spring system and reduced deformable models using modal analysis etc.

ODEs, as an important branch of modern mathematics and an effective tool for people to solve various practical problems, have been generally used to represent the underlying physics in the computer graphics field. Take free-fall motion of a bouncing ball as an instance, Newton's second law $F = dP/dt = d(mv)/dt = mdv/dt = ma$ actually uses the derivative of velocity v with respect to time t to denote acceleration a , and then times mass m to describe force F . The introduction of ordinary differential equations into geometric modelling provides a potential to generate physically realistic facades as well as deformations of 3D models. In the existing work, ODE-based sweeping surfaces [5], ODE-based surface deformations [6] and ODE-based sweeping blending [7] have been developed. ODE-based methods create, manipulate, and animate 3D models with the solution to a suitably selected vector-valued ordinary differential equation subjected to user's specified boundary conditions. The created ODE sweeping surface-represented 3D models have small data and good continuities between two adjacent patches, whilst the traditional surface modelling approaches use a vast number of surface vertices or control points. Furthermore, Polygon models cannot approximate curved surfaces with many flat facets. NURBS and other patch-based modelling require many manual operations to stitch two adjacent patches together. It is difficult even impossible to achieve high-order continuities.

Skin deformations play an important role in character animation and facial animation. Chaudhry et al. made a review of main skin deformation techniques in [8]. In comparison with volume-based techniques, surface-based techniques are widely used in computer animation field as they are much simpler. Surface-based skin deformation techniques can be divided into purely geometric, physics-based, and data-driven. Purely geometric techniques, which ignore the underlying physics but only manipulate vertices to change shapes. Physics-based techniques create more natural and realistic appearances but require heavy computational calculations. If example models are of high quality, data-driven techniques can also create realistic skin deformations but require a lot of effort to prepare high quality example models.

The simplest method of purely geometric deformation is linear interpolation. Other methods like joint-related approaches, which treat the skin by using an explicit function of the skeleton to move the shell. One such approach examined in [9] is called skeleton subspace deformation, which introduces vertex weights for smooth transformation of bones. Example-based techniques, which use existing poses as the base to interpolate the keyframes in [10].

The physics-based deformation techniques are based on the anatomy and biomechanics of skin deformation deriving from the action of muscles. It uses the mass-spring system, finite element method (FEM) and finite volume method to determine contractile muscle forces and muscle geometry transformation. In [11], FEM was used to simulate dynamic deformations of a visco-elastic object, while the mass-spring system is more used for interactive animation of deformable models, carried out in [12]. The finite volume method is more used to simulate the deformable movement of skeletal muscles. These methods are all based on partial differential equation (PDE), though they approximate PDEs with different forms. Another PDE-based skin deformation model was proposed in [13], which reduces the numerical calculations and creates physically realistic skin deformations with high efficiency by including vertex identification on iso-parametric curves, Fourier series conversion and the analytic solution to a formulated model with underlying physics.

Different from the above methods, this paper will propose a new, efficient, and physics-based algorithm of facial blendshapes derived from Newton's second law, and combine it with ODE sweeping surface-based 3D modelling to create 3D facial models with small data and animate ODE sweeping surface-represented 3D models with the proposed physics-based algorithm of facial blendshapes.

III. CREATION OF ODE SWEEPING SURFACE-REPRESENTED 3D MODELS

Creation of ODE sweeping surface-represented 3D models is inspired by [14]. For completeness, this section briefly introduces the theory and method, including the mathematical model in Subsection A, accurate closed form solution in Subsection B, continuities between two adjacent patches in Subsection C, and the creation process of ODE sweeping surface-represented 3D models in Subsection D.

A. Mathematical Model

The mathematical model of ODE sweeping surfaces consists of a vector-valued ordinary differential equation and boundary conditions. Depending on different continuity requirements, the ordinary differential equation and boundary conditions have different forms.

The accurate solution of second-order, fourth-order, and sixth-order ordinary differential equations (ODEs) has two, four, and six unknown constants to satisfy the requirement of position continuity, position and tangent continuities, and position, tangent, and curvature continuities. When curvature continuity is not required, curvature can be used to manipulate surfaces efficiently. Based on these considerations, this paper uses the following vector-valued sixth-order ordinary differential equation proposed in [14]:

$$\rho \frac{d^6 \mathcal{S}(u, v_i)}{du^6} + \eta \frac{d^4 \mathcal{S}(u, v_i)}{du^4} + \frac{d^2 \mathcal{S}(u, v_i)}{du^2} = 0 \quad (1)$$

where ρ , η , and λ are called shape control parameters.

In order to make two adjacent surface patches achieve position, tangent, and curvature continuities, two adjacent ODE sweeping surfaces must share the following boundary conditions:

$$\begin{aligned} u = 0 \quad \mathcal{S}(0, v) = \mathbf{c}_1(v) \quad \frac{\partial \mathcal{S}(0, v)}{\partial u} = \mathbf{c}_2(v) \quad \frac{\partial^2 \mathcal{S}(0, v)}{\partial u^2} = \mathbf{c}_3(v) \\ u = 1 \quad \mathcal{S}(1, v) = \mathbf{c}_4(v) \quad \frac{\partial \mathcal{S}(1, v)}{\partial u} = \mathbf{c}_5(v) \quad \frac{\partial^2 \mathcal{S}(1, v)}{\partial u^2} = \mathbf{c}_6(v) \end{aligned} \quad (2)$$

where $\mathbf{c}_1(v)$ and $\mathbf{c}_4(v)$ are boundary curves of an ODE sweeping surface, $\mathbf{c}_2(v)$ and $\mathbf{c}_5(v)$ are the first partial derivatives across boundary curves, and $\mathbf{c}_3(v)$ and $\mathbf{c}_6(v)$ are the second partial derivatives across boundary curves.

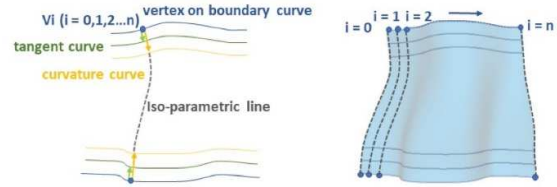


Fig. 1. ODE surface and boundary and control curves [14]

For each of the boundary curves $\mathbf{c}_1(v)$ and $\mathbf{c}_4(v)$ shown in Fig. 1, we can draw two control curves next to each of the boundary curves. From the boundary curve and two control curves, we can use the method given in [14] to obtain $\mathbf{c}_2(v)$, $\mathbf{c}_5(v)$, $\mathbf{c}_3(v)$ and $\mathbf{c}_6(v)$.

B. Accurate closed form solution

For each of components x , y , and z , (1) presents one ordinary differential equation. We use $\mathcal{S}_\phi(u, v_i)$ ($\phi = x, y, z$) to represent the components of the surface $\mathcal{S}(u, v_i)$, and let $\mathcal{S}_\phi(u, v_i) = e^{ru}$, (1) is transformed into an algebra equation. Solving it and submitting the obtained roots back to $\mathcal{S}_\phi(u, v_i) = e^{ru}$, we can obtain the solution of ODE (1). Substituting the obtained accurate closed form solution into boundary conditions (2), all the unknown constraints in the solution are determined, and the accurate closed form solution take the form of:

$$\mathcal{S}(u, v) = g_1(u)\mathbf{c}_1(v) + g_2(u)\mathbf{c}_2(v) + g_3(u)\mathbf{c}_3(v) + g_4(u)\mathbf{c}_4(v) + g_5(u)\mathbf{c}_5(v) + g_6(u)\mathbf{c}_6(v) \quad (3)$$

where the known functions $g_i(u)$ ($i = 1, 2, 3, 4, 5, 6$) are given in [14].

C. Continuity between adjacent patches

Since the accurate closed form solution exactly satisfies boundary conditions, two ODE patches obtained from the solution share the same boundary conditions and naturally achieve position, tangent, and curvature continuities on their shared boundary curves at $u = 0$ and $u = 1$.

On the boundary curves at $v = 0$ and $v = 1$, two ODE sweeping surfaces can be written as $\bar{S}_\phi(u, 0)$ and $\bar{S}_\phi(u, 1)$. And the position, tangent, and curvature continuities are obtained by satisfying the following conditions:

$$\begin{aligned} S_\phi(u, 0) &= \bar{S}_\phi(u, 1) \\ \partial S_\phi(u, 0)/\partial v &= \partial \bar{S}_\phi(u, 1)/\partial v \\ \partial^2 S_\phi(u, 0)/\partial^2 v &= \partial^2 \bar{S}_\phi(u, 1)/\partial^2 v \end{aligned} \quad (4)$$

D. Creation process of ODE sweeping surface-represented 3D models

The above method can be used to create 3D models from scratch or convert polygon, NURBS and subdivision models into ODE sweeping surface-represented models. We briefly introduce these methods below.

When 3D models will be created from scratch, the following steps will be followed. First, we draw boundary curves and control curves and determine the boundary conditions (2) from boundary curves and control curves. Then we solve ODE (1) subject-ed to boundary conditions (2) to obtain the accurate closed form solution determine the unknow constants in the closed form solution. After that, the obtained closed form solutions for two adjacent ODE surfaces are substituted into (4) to satisfy the continuity requirements. Finally, a 3D model is created from a network of boundary curves called a wireframe.

When a polygon, NURBS and subdivision 3D model will be converted into an ODE sweeping surface-represented model, we follow the following steps. First, we segment the 3D model into patches. Then we extract the boundary curves of these patches to obtain wireframe of the 3D model and draw two control curves for each of extracted boundary curves. From the boundary and control curves, we determine boundary conditions (2). After that, we solve ODE (1) subjected boundary conditions (2) to obtain the accurate closed form solution and use (4) to satisfy the continuity requirements. The obtained closed form solution is used to create one ODE sweeping patch. An ODE sweeping surface-represented 3D model is created after the wireframe is filled with ODE patches. Fig. 2 shows a converted ODE sweeping surface-represented dog model.



Fig. 2. ODE surface and boundary and control curves [14].

IV. NEWTON'S SECOND LAW-BASED FACIAL BLENDSHAPES

In this section, we develop a new, efficient, and physics-based algorithm of facial blendshapes based on Newton's second law. In what follows, we first present New-ton's second

law, then derive its accurate closed form solution, and finally determine the unknow constants involved in the accurate closed form solution.

In a three-dimensional space, Newton's second law can be written as:

$$m\mathbf{a} = \mathbf{f} \quad (5)$$

where m stands for mass, \mathbf{a} denotes the acceleration which has three components $a^{(i)}$ ($i = 1, 2, 3$) with $a^{(1)} = a_x$, $a^{(2)} = a_y$ and $a^{(3)} = a_z$, and \mathbf{f} stands for the external force also with three components in each direction $f^{(i)}$ ($i = 1, 2, 3$) with $f^{(1)} = f_x$, $f^{(2)} = f_y$ and $f^{(3)} = f_z$.

The acceleration \mathbf{a} is related to the position change \mathbf{x} as:

$$\mathbf{a} = \frac{d^2\mathbf{x}}{dt^2} \quad (6)$$

where the position change \mathbf{x} has three components $x^{(i)}$ ($i = 1, 2, 3$) with $x^{(1)} = x$, $x^{(2)} = y$, and $x^{(3)} = z$.

Substituting (6) into (5), the following formula could be obtained in the form of components:

$$m \frac{d^2 x^{(i)}}{dt^2} = f^{(i)} \quad (i = 1, 2, 3) \quad (7)$$

The next step is to determine initial and boundary conditions of solving the above second-order ordinary differential equations. This can be achieved by considering the positions at the initial and final poses and the velocity at the initial pose.

Assuming a 3D model is defined by the polygon vertices $\bar{\mathbf{x}}_n$ ($n = 1, 2, 3, \dots, N$) at the initial pose $t = 0$, and $\bar{\bar{\mathbf{x}}}_n$ ($n = 1, 2, 3, \dots, N$) at the final pose $t = 1$, the position change is $\mathbf{x}_n = 0$ at the initial pose $t = 0$ and $\mathbf{x}_n = \bar{\bar{\mathbf{x}}}_n - \bar{\mathbf{x}}_n$ at the final pose $t = 1$.

If we assume that the rate of the position change at $t = 0$ is zero, i.e. $dx_n/dt = 0$, the initial and boundary conditions could be gained:

$$\begin{aligned} t = 0 \quad & \mathbf{x}_n = 0 \\ & \frac{d\mathbf{x}_n}{dt} = 0 \\ t = 1 \quad & \mathbf{x}_n = \bar{\bar{\mathbf{x}}}_n - \bar{\mathbf{x}}_n \\ & (n = 1, 2, 3, \dots, N) \end{aligned} \quad (8)$$

In the form of components, (8) could be changed into:

$$\begin{aligned} t = 0 \quad & \mathbf{x}_n^{(i)} = 0 \\ & \frac{d\mathbf{x}_n^{(i)}}{dt} = 0 \\ t = 1 \quad & \mathbf{x}_n^{(i)} = \bar{\bar{\mathbf{x}}}_n^{(i)} - \bar{\mathbf{x}}_n^{(i)} \\ & (i = 1, 2, 3; n = 1, 2, 3, \dots, N) \end{aligned} \quad (9)$$

Replacing $\mathbf{x}^{(i)}$ with $\mathbf{x}_n^{(i)}$ in (9), we get:

$$m \frac{d^2 \mathbf{x}_n^{(i)}}{dt^2} = f_n^{(i)} \quad (10)$$

$$(i = 1, 2, 3; n = 1, 2, 3, \dots, N)$$

Dividing both sides of (10) by m , we obtain:

$$m \frac{d^2 \mathbf{x}_n^{(i)}}{dt^2} = f_n^{(i)} \quad (11)$$

$$(i = 1, 2, 3; \quad n = 1, 2, 3, \dots, N)$$

Integrating (11) with respect to time t twice, we obtain:

$$\mathbf{x}_n^{(i)} = \frac{f_n^{(i)} t^2}{2m} + c_0 t + c_1 \quad (12)$$

$$(i = 1, 2, 3; \quad n = 1, 2, 3, \dots, N)$$

where c_0 and c_1 are unknown constants.

Introducing (12) into (9), and solving for the two unknown constants c_0 and c_1 as well as the external force $f_n^{(i)}$, we have:

$$c_0 = c_1 = 0$$

$$f_n^{(i)} = 2m[\bar{\mathbf{x}}_n^{(i)} - \bar{\mathbf{x}}_n^{(i)}] \quad (13)$$

$$(i = 1, 2, 3; \quad n = 1, 2, 3, \dots, N)$$

Substituting (13) into (12), we reach the following formula to calculate the position change:

$$\mathbf{x}_n^{(i)} = [\bar{\mathbf{x}}_n^{(i)} - \bar{\mathbf{x}}_n^{(i)}] t^2 \quad (14)$$

$$(i = 1, 2, 3; \quad n = 1, 2, 3, \dots, N)$$

Finally, by adding the position change $\mathbf{x}_n^{(i)}$ to the position values $\bar{\mathbf{x}}_n^{(i)}$ of each vertex of the 3D model at the initial pose $t = 0$, we get the following formulae which will be used to determine the position values $\hat{\mathbf{x}}_n^{(i)}$ at any poses between the initial and final poses:

$$\hat{\mathbf{x}}_n^{(i)} = \bar{\mathbf{x}}_n^{(i)} + [\bar{\mathbf{x}}_n^{(i)} - \bar{\mathbf{x}}_n^{(i)}] t^2 \quad (15)$$

$$(i = 1, 2, 3; \quad n = 1, 2, 3, \dots, N)$$

With time t changing in the interval $[0, 1]$, we use the above formulae to obtain a set of different position values of the 3D model. Then we use these position values to create facial blendshapes between the initial and final poses.

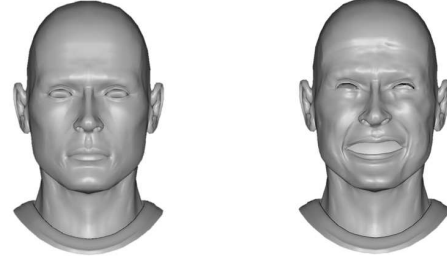
V. EXPERIMENTS

We implemented the above Compared with the animation of other parts of a virtual human character, face deformation is more crucial since it requires higher precision to be regarded as realistic. Thus, in order to demonstrate the result of the Experiments, using the source provided by [15], two face models in the same topological structure with different expressions are chosen as Fig.3. We first use the ODE-based approach to recreate the two different models and then use linear interpolation to calculate the keyframes in between. Finally, form an animation between the two different reconstructed expression models to demonstrate the algorithm implementation as well as how it could be applied in face deforming animation.

A. Patch Separation

Different facial expressions can be defined with some parameters. Paul et al. provided the Facial Action Coding

System (FACS) in [16] to use corresponding parameters to control disparate face shapes. Hamm et al. claimed that the facial appearance could be used to encode the action of each facial muscles with marginally imperative changes in [17]. Furthermore, any anatomically possible facial expression could be encoded by FACS through deconstructing it into temporal



(a) Neutral Model

(b) Laugh Model

Fig. 3. Source models in two different states.

segments and action units, which could be used for further interpolation to adapt to any intelligent environment, mentioned in [18]. Differently, we use a wireframe to define a 3D model and use ODE sweeping surfaces to recreate the 3D model from the wireframe.

In this paper, we separate the face model into seven different patches shown in Fig.4. These seven patches are: eye socket, eye bone, nose bridge, nostril, cheek, mouth, and jaw. After separating the 3D model into patches, we extract the boundary curves of the patches.

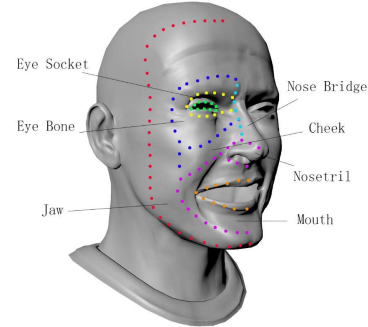


Fig. 4. Patch Separation. We separate the face into 7 different patches: eye socket, eye bone, nose bridge, nostril, cheek, mouth, and jaw.

B. Facial Recreation

After extracting all the boundary curves, we add two control curves for each of the boundary curves to determine the tangents and curvature on the extracted boundary curves which are actually the first and second partial derivatives. The left of Fig.5 shows the boundary curves and two control curves, and the right of Fig.5 shows the curves obtained from ODE sweeping surfaces.

We compare the original model with ODE sweeping surface-represented model of the neutral model in Fig.6(a) and the laugh model in Fig.6(b). In the figures, the left images are from the original model and the right images are from the ODE-sweeping surface-represented model.

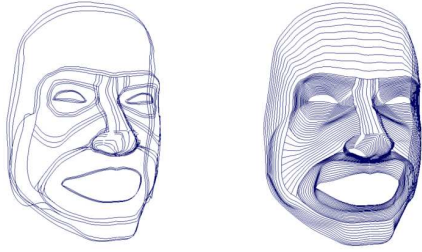


Fig. 5. Wireframes extraction. The left figure shows the boundary curves and control curves of each patch. The right figure shows the ODE-based generated curves. When converting models to ODE sweeping surfaces, the data size significantly reduces, which are more convenient for storage and speed up network transmission.

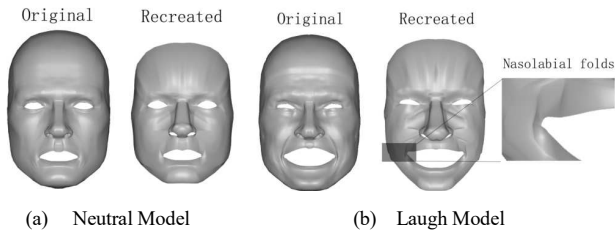


Fig. 6. Comparison of original and recreated model. The original laugh model contains details like nasolabial folds, which we use ODE-based recreated surface to restore them. ODE sweeping technique could store significant details through manipulating control curves..

TABLE I. COMPARISON OF SIZE BETWEEN ORIGINAL AND RECREATED MODELS

	Model Type	Original Model	Recreated Model	Reduce Rate
vertices number	neutral	8221	4764	42.05%
	laugh	8246	4764	42.23%
polygon number	neutral	15378	4236	72.45%
	laugh	15411	4236	72.51%

Although few patches are used to create ODE sweeping surface-represented 3D model, the recreated face models shown in the right images do not lose too many details compared with the original face models shown in the left images of Figs. 6(a) and 6(b). In addition to the relatively good approximation of the recreated face models to the original models, the ODE sweeping surface-represented models, shown in TABLE I, is defined by 4764 points on boundary curves and control curves and three shape control parameters involved in (1). They indicate that the design variables of the ODE sweeping surface-represented model is only about 58% of the design variables of the corresponding original model. Therefore, the ODE sweeping surface-represented models could noticeably save the data storage space with similar quality to the original model. Furthermore, the ODE sweeping surface-represented models provide a convenient way to quickly manipulate the face shapes and expressions through modifying boundary curves, control curves, and shape control parameters. Since two adjacent ODE sweeping patches share the same boundary conditions, up to curvature continuities are naturally achieved, which avoids manual operations to stitch two adjacent ODE sweeping patches

together and save a lot of time in achieving curvature continuities between adjacent patches.

C. Facial Blendshapes

Facial blendshapes of polygon models are to interpolate the vertices of the neutral model and target model. When a neutral model and a target model have different topology or vertices, a lot of work and time are required to find the correspondence between the neutral model and target model. Since many polygon vertices are used to define a polygon model, a lot of time is also required to calculate the interpolation between the neutral model and the target model.

After a neutral model and a target model have been converted into ODE-sweeping surface-represented models, facial blendshapes can be achieved through interpolate the boundary curves and control curves, which avoids the work in finding the correspondence between two models with different topology and vertices and saves the time in interpolating many polygon vertices.

In comparison with most widely used facial blendshapes based on linear interpolation, which is purely geometric, facial blendshapes use the Newton's second law. Therefore, the proposed algorithm is physics-based and has a potential to create more natural appearance of facial blendshapes. We have implemented these two algorithms and use both of them to create facial blendshapes.

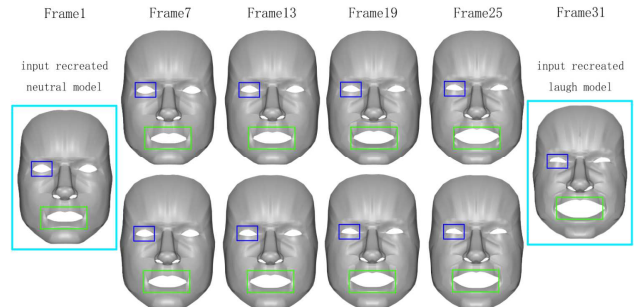


Fig. 7. 31 keyframe models blending two different interpolation methods. The first row is the facial blendshapes generated by linear interpolation while the second row is generated by the interpolation derived Newton's second law. Here we use the same size green box and blue box to respectively show the shape change of mouth and eye. After comparing the shape change of mouth and eye, it could be found that, the change acceleration of the second row keeps the same, whilst that of the first row is small at first and surges at the end i.e., more natural.

The facial blendshapes shown in the first row of Fig.7 are created with the Newton's second law-based interpolation. First, we interpolate two boundary curves and four control curves for each of ODE sweeping surface patches of a 3D model to obtain the boundary conditions at the poses between the neutral model and target model. Then, ODE sweeping surface-represented 3D models are created from the obtained boundary conditions. Plus the original neutral model and target model, 30 keyframe models are used to create smooth facial animation.

The facial blendshapes shown in the second row of Fig. 7 are created with linear interpolation. Similarly, 30 keyframe models including the original neutral model and target model are used to create smooth facial animation.

Comparing the images shown in Figs. 7, we can conclude that facial blendshapes based on linear interpolation generate equal changes between two adjacent models, i.e., move with a same velocity. Actually, facial shape changes are not in a same velocity. For example, when we open or close our mouth, two lips may move quickly or slowly at the beginning or at the end. Such an acceleration or deceleration effect could create more natural lip movements. With the interpolation derived Newton's second law, it is easy to create such acceleration and or deceleration effect and more natural blendshapes could be generated.

VI. CONCLUSION

In this paper, we have developed a new and physics-based algorithm of facial blendshapes derived from Newton's second law and combined it with ODE sweeping surfaces to represent neutral and target models, create the models between neutral and target models, and generate physics-based facial animation. The proposed method has the advantages of small data of created models, easy obtainment and natural maintenance of required continuities, efficient shape manipulation, correspondence avoidance between the neutral model and target model, natural facial blendshapes, and easy creation of special animation effects.

However, wireframes defining facial models are manually extracted in our work, which requires many manual operations and a lot of time. This can be improved by developing an automatic extraction algorithm. A template facial model with pre-defined surface patches can be introduced to automatically segment any 3D facial models, and a Maya Mel script can be implemented to automatically extract boundary curves of the segmented patches and generate control curves. Facial blendshapes include interpolation between a neutral model and a target model and blending among a neutral model and many target models. In this paper, we only investigate facial blendshapes using interpolation between a neutral model and a target model, and facial blendshapes among a neutral model and many target models have not been investigated. New algorithms of facial blendshapes based on Newton's second law can be developed to tackle this problem. In order to facilitate the applications of the proposed method, an interactive interface based on ODE sweeping surface-represented models and the interpolation using Newton's second law can be developed to create facial blendshapes easily. In the future, we will investigate these issues.

ACKNOWLEDGMENT

This research is supported by the PDE-GIR project which has received funding from the European Union Horizon 2020

research and innovation programme under the Marie Skłodowska-Curie grant agreement No 778035.

REFERENCES

- [1] M. Russo, *Polygonal modeling: basic and advanced techniques*. Jones & Bartlett Learning (2006).
- [2] L. Piegl and W. Tiller, *The NURBS book*. Springer Science & Business Media (2012).
- [3] T. Várady, P. Salvi, and A. Rockwood, "Transfinite surface interpolation with interior control," *Graphical models*, vol. 74, no. 6, pp. 311–320. Elsevier (2012).
- [4] A. Nealen, M. Müller, R. Keiser, E. Boxerman, and M. Carlson, "Physically based deformable models in computer graphics," in *EUROGRAPHICS 2005 STAR-STATE OF THE ART REPORT*. Citeseer (2005).
- [5] L. You, X. Yang, M. Pachulski, and J. J. Zhang, "Boundary constrained swept surfaces for modelling and animation," in *Computer Graphics Forum*, vol. 26, no. 3, pp. 313–322 (2007).
- [6] L. You, X. Yang, X. Y. You, X. Jin, and J. J. Zhang, "Shape manipulation using physically based wire deformations," *Computer Animation and Virtual Worlds*, vol. 21, no. 3-4, pp. 297–309 (2010).
- [7] E. Chaudhry, L. You, X. Jin, X. Yang, and J. J. Zhang, "Shape modeling for animated characters using ordinary differential equations," *Computers & graphics*, vol. 37, no. 6, pp. 638–644 (2013).
- [8] E. Chaudhry, L. You, and J. J. Zhang, "Character skin deformation: A survey," in *2010 Seventh International Conference on Computer Graphics, Imaging and Visualization*. pp. 41–48 (2010).
- [9] N. Magnenat-Thalmann, R. Laperrre, and D. Thalmann, "Joint-dependent local deformations for hand animation and object grasping," in *In Proceedings on Graphics interface'88*. Citeseer (1988).
- [10] B. Allen, B. Curless, and Z. Popović, "Articulated body deformation from range scan data," *ACM Transactions on Graphics (TOG)*, vol. 21, no. 3, pp. 612–619 (2002).
- [11] G. DeBunne, M. Desbrun, M.-P. Cani, and A. H. Barr, "Dynamic real-time deformations using space & time adaptive sampling," in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pp. 31–36 (2001).
- [12] S. M. Platt and N. I. Badler, "Animating facial expressions," in *Proceedings of the 8th annual conference on Computer graphics and interactive techniques*, pp. 245–252 (1981).
- [13] S. Bian, Z. Deng, E. Chaudhry, L. You, X. Yang, L. Guo, H. Ugail, X. Jin, Z. Xiao, and J. J. Zhang, "Efficient and realistic character animation through analytical physics-based skin deformation," *Graphical Models*, vol. 104, p. 101035 (2019).
- [14] S. Bian, G. Maguire, W. Kokke, L. You, and J. J. Zhang, "Efficient c2 continuous surface creation technique based on ordinary differential equation," *Symmetry*, vol. 12, no. 1, p. 38 (2020).
- [15] Mesh Data from Deformation Transfer for Triangle Meshes, <http://people.csail.mit.edu/sumner/research/deftransfer/data.html>, last accessed 2021/01/05.
- [16] E. Paul, H. Joseph C, and F. Wallace V, *Facial action coding system*. Salt Lake City: A Human Face (2002).
- [17] J. Hamm, C. G. Kohler, R. C. Gur, and R. Verma, "Automated facial action coding system for dynamic analysis of facial expressions in neuropsychiatric disorders," *Journal of neuro-science methods*, vol. 200, no. 2, pp. 237–256 (2011).
- [18] A. Freitas-Magalhães, *The face of lies*. Leya (2013).