ODE-Driven Sketch-Based Organic Modelling

Ouwen Li[1](✉), Zhigang Deng[2], Shaojun Bian[1], Algirdas Noreika[3], Xiaogang Jin[4], Ismail Khalid Kazmi[5], Lihua You[1], and Jian Jun Zhang[1]

[1] National Centre for Computer Animation, Bournemouth University,
Bournemouth, Dorset, UK
oli@bournemouth.ac.uk

[2] University of Houston, Houston, Texas, USA

[3] Indeform Ltd., K. Petrausko g. 26, 44156 Kaunas, Lithuania

[4] State Key Lab of CAD & CG, Zhejiang University, Hangzhou 310058, China

[5] Teesside University, Middlesbrough, Tees Valley, UK

Abstract. How to efficiently create 3D models from 2D sketches is an important problem. In this paper we propose a sketch-based and ordinary differential equation (ODE) driven modelling technique to tackle this problem. We first generate 2D silhouette contours of a 3D model. Then, we select proper primitives for each of the corresponding silhouette contours. After that, we develop an ODE-driven and sketch-guided deformation method. It uses ODE-based deformations to deform the primitives to exactly match the generated 2D silhouette contours in one view plane. Our experiment demonstrates that the proposed approach can create 3D models from 2D silhouette contours easily and efficiently.

Keywords: Organic models · Sketch-guided modelling · ODE-driven deformations

1 Introduction

Mainstream modelling approaches such as polygon and NURBS can create detailed 3D models. However, they require good knowledge and skills to use them, involve heavy manual operations, and take a lot of time to complete modelling tasks. In order to address these problems, various sketch-based modelling (SBM) approaches have been developed in the past several decades [13].

Ordinary Differential Equations (ODE) have been widely used to describe various physical laws in scientific computing and engineering applications. For example, fourth-order ODEs have been used to describe the lateral bending of elastic beams in structural engineering. Therefore, ODE-driven modelling is physics-based and able to generate more realistic appearances and deformations [19]. In order to generate such physically realistic surfaces, in this paper we introduce ODE-based modelling to develop primitive deformers to deform primitives to match user's drawn sketches.

M. Gavrilova et al. (Eds.): CGI 2019, LNCS 11542, pp. 453–460, 2019.
https://doi.org/10.1007/978-3-030-22514-8_44

The main contributions of our approach is that we develop an efficient, ODE-driven, and sketch-guided deformation method to create 3D models quickly. It can deform primitives to exactly match the generated silhouette contours. Compared to existing methods, it automates shape manipulation, avoids tedious manual operations, can deform primitives to match the generated silhouette contours quickly, and is effective in achieving different shapes of a primitive.

2 Related Work

The work proposed in this paper is related to sketch-based modelling, especially the primitive-based systems, and ODE-based geometric processing. In the following, we briefly review the existing work in these fields.

Sketch-Based Modelling. Sketch-based-modelling (SBM) can be broadly divided into direct mesh generation and primitive-based mesh creation.

In the category of direct mesh generation, several systems have been proposed to generate organic models. The surface inflation technique extrudes a polygonal mesh from a given skeleton outwards and does a good job in modelling stuffed toys. One trend is to inflate free-form surfaces to create simple stuffed animals and other rotund objects in a SBM fashion [9,10,12]. BendSketch [11] complements those works by enabling complex curvature patterns on surfaces. In order to give the bending information, users need to draw a set of lines that comply with what BendSketch system has specified, which mimics the hatching technique artists often utilise to express the sense of volume and curvature information on the surfaces.

Unlike the inflating systems, primitives-based systems decompose the modelling task as a process of creating a certain set of geometry primitives and further editing the primitives [5,14,17]. The idea of assembling simple geometric primitives to form 3D models is commonly used in CSG (Constructive Solid Geometry) modelling [5,14]. Structured annotations for 2D-to-3D modelling [8], on the other hand, focus on organic modelling.

ODE-Based Geometric Processing. ODEs have been widely applied in scientific computing and engineering analyses to describe the underlying physics. For example, fourth-order ODEs have been used to describe the lateral bending deformations of elastic beams. Introducing ODEs into geometric processing can create physically realistic appearances and deformations of 3D models. ODE-based sweeping surfaces [19], ODE-based surface deformations [4,20], and ODE-based surface blending [18] have also been developed previously. Although researchers studied ODE-based geometric surface creation and deformations, how to use ODE-based modelling to deform geometric primitives and create new shapes from the user's drawn sketches has been under-explored to date.

3 System Overview

As shown in the Fig. 1, one can extract 2D silhouette contours from sketches, or to directly draw 2D silhouette contours.

After generating 2D silhouette contours, the next step is to set super-ellipsoid as 3D primitives. We sample the silhouette contours into point cloud and we use radial Euclidean distance to format the global distortion [6] to determine the parameters of the super-ellipse(ϵ_z, r_x, r_y). If one only feeds the system with the 2D sketch in one view not providing z-axis depth information, then the radius of z-axis r_z and ϵ_{xy} requires users to manually set, and any necessary translations in z-axis should be done manually. Since the parameter determination took place in object space, when we transform the superellipsoid primitives into the world space, they will align with its silhouette in world space.

Then the ODE-driven deformation method described in Sect. 4 is applied to deform the primitives to exactly match the corresponding 2D silhouette contours.

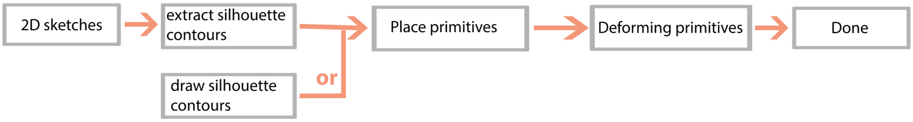


Fig. 1. System overview.

4 Primitive Deformers

Here we propose an ODE-driven and sketch-guided primitive deformation method. It is developed from a simplified version of the Euler-Lagrange PDE (Partial Differential Equation), which is widely used in physically-based surface deformations and briefly reviewed below.

As discussed in [1], the main requirement for physically-based surface deformations is an elastic energy which considers the local stretching and bending of two-manifold surfaces called thin-shells. When a surface $\mathbb{S} \subset \mathbb{R}^3$, parameterized by a function $\mathbf{P}(u, v) : \Omega \subset \mathbb{R}^2 \mapsto \mathbb{S} \subset \mathbb{R}^3$, is deformed to a new shape $\mathbb{S}'$ through adding a displacement vector $\mathbf{d}(u, v)$ to each point $\mathbf{P}(u, v)$, the change of the first and second fundamental $I(u, v)$, $\mathit{\Pi}(u, v) \in \mathbb{R}^{2\times 2}$ forms in differential geometry [7] yields a measure of stretching and bending, as described in [15]:

$$E_{shell}(\mathbb{S}') = \int_{\Omega} k_s \|I' - I\|_F^2 + k_b \|\mathit{\Pi}' - \mathit{\Pi}\|_F^2 \,\mathrm{d}u\,\mathrm{d}v, \tag{1}$$

where I' and $\mathit{\Pi}'$ are the first and second fundamental forms of the surface $\mathbb{S}'$, $\|.\|$ indicates a (weighted) Frobenius norm, and the stiffness parameters k_s and k_b are used to control the resistance to stretching and bending.

Generating a new deformed surface requires the minimization of the above Eq. 1, which is non-linear and computationally expensive for interactive applications. In order to avoid the nonlinear minimization, the change of the first and second fundamental forms is replaced by the first and second order partial derivatives of the displacement function $\mathbf{d}(u, v)$ [2,16], i. e.,

$$\tilde{E}_{shell}(\mathbf{d}) = \int_{\Omega} k_s(\|\mathbf{d}_u\|^2 + \|\mathbf{d}_v\|^2) + k_b(\|\mathbf{d}_{uu}\|^2 + 2\|\mathbf{d}_{uv}\| + \|\mathbf{d}_{vv}\|^2)\, \mathrm{d}u\, \mathrm{d}v, \tag{2}$$

where $d_x = \frac{\partial}{\partial x}$ and $d_{xy} = \frac{\partial^2}{\partial x \partial y}$. The minimization of the above equation can be obtained by applying variational calculus, which leads to the following Euler-Lagrange PDE:

$$-k_s \triangle \mathbf{d} + k_b \triangle^2 \mathbf{d} = 0, \tag{3}$$

where $\triangle$ and $\triangle^2$ are the Laplacian and the bi-Laplacian operators, respectively,

$$\begin{aligned} \triangle \mathbf{d} &= div \nabla \mathbf{d} = \mathbf{d}_{uu} + \mathbf{d}_{vv}, \\ \triangle^2 \mathbf{d} &= \triangle(\triangle \mathbf{d}) = \mathbf{d}_{uuuu} + 2\mathbf{d}_{uuvv} + \mathbf{d}_{vvvv}. \end{aligned} \tag{4}$$

Using the sketched 2D silhouette contours shown in Fig. 2 to change the shape of the primitive can be transformed to the generation of a sweeping surface which passes through the two sketched 2D silhouette contours. The generator that creates the sweeping surface is a curve of the parametric variable u only, and the two silhouette contours are trajectories. If Eq. (3) is used to describe the generator, the parametric variable v in Eq. (3) drops, and we have $\mathbf{d}_{vv} = 0$ and $\mathbf{d}_{vvvv} = 0$. Substituting $\mathbf{d}_{vv} = 0$ and $\mathbf{d}_{vvvv} = 0$ into Eq. (4), we obtain the following simplified version of the Euler-Lagrange PDE, seen as (5), which is actually a vector-valued ODE.

$$k_b \frac{\partial^4 \mathbf{d}}{\partial u^4} - k_s \frac{\partial^2 \mathbf{d}}{\partial u^2} = 0. \tag{5}$$

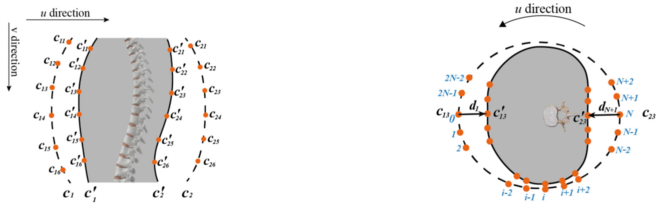
As pointed out in [3], the finite difference solution to ODEs is very efficient, we here investigate such a numerical solution to Eq. (5). For a typical node i, the central finite difference approximations of the second and fourth order derivatives can be written as:

$$\begin{aligned} \frac{\partial^2 \mathbf{d}}{\partial u^2}|_i &= \frac{1}{\triangle u^2}(\mathbf{d}_{i+1} - 2\mathbf{d}_i + \mathbf{d}_{i-1}), \\ \frac{\partial^4 \mathbf{d}}{\partial u^4}|_i &= \frac{1}{\triangle u^4}[6\mathbf{d}_i - 4(\mathbf{d}_{i-1} + \mathbf{d}_{i+1}) + \mathbf{d}_{i-2} + \mathbf{d}_{i+2}]. \end{aligned} \tag{6}$$

Introducing Eq. (6) into Eq. (5), the following finite difference equation at a representative node i can be written as:

$$(6k_b + 2k_s h^2)\mathbf{d}_i + k_b \mathbf{d}_{i-2} + k_b \mathbf{d}_{i+2} - (4k_b + k_s h^2)\mathbf{d}_{i-1} - (4k_b + k_s h^2)\mathbf{d}_{i+1} = 0. \tag{7}$$

For organic models, the 3D shape defined by two silhouette contours is closed in the parametric direction u as indicated in Fig. 2b. Therefore, we can extract some closed curves each of which passes through the two corresponding points on the two silhouette contours. Taking the silhouette contours in Fig. 2a as an example, we find two corresponding points $\mathbf{C}_{13}$ and $\mathbf{C}_{23}$ on the original silhouette contours $\mathbf{c}_1$ and $\mathbf{c}_2$, and two corresponding points $\mathbf{C}'_{13}$ and $\mathbf{C}'_{23}$ on the deformed silhouette contours $\mathbf{c}'_1$ and $\mathbf{c}'_2$ as shown in Fig. 2b. Then, we extract a closed curve $\mathbf{c}(u)$ passing through the two corresponding points $\mathbf{C}_{13}$ and $\mathbf{C}_{23}$ from the 3D model in Fig. 2a and depict it as a dashed curve in Fig. 2b. Assuming that the deformed shape of the closed curve $\mathbf{c}(u)$ is $\mathbf{c}'(u)$, the displacement difference between the original closed curve and the deformed closed curve is $\mathbf{d}(u) = \mathbf{c}'(u) - \mathbf{c}(u)$.



(a) side view

(b) top view

Fig. 2. Finite difference nodes for local shape manipulation from sketches in top and side view planes, respectively.

In order to use the finite difference method to find the displacement difference $\mathbf{d}(u)$, we uniformly divide the closed curve into $2N$ equal intervals as indicated in Fig. 2b. With the displacement difference at node 0 and node N already known, i. e. $\mathbf{d}_0 = \mathbf{C}'_{13} - \mathbf{C}_{13}$ and $\mathbf{d}_N = \mathbf{C}'_{23} - \mathbf{C}_{23}$, we can form a $2N$ linear algebra equations derived from (7) for each of these nodes' displacement. Solve the equations and add all the displacement differences to the original curve $\mathbf{c}(u)$, we can then obtain the deformed curve $\mathbf{c}'(u)$, and depict it as a solid curve in Fig. 2b. Repeating the above operations for all other points on the two silhouette contours, we obtain all deformed curves that describe a new 3D deformed shape. This method also applies to deformations responding to free form curves, which can be seen from the creation of a human leg shown in Fig. 3b.

The aforementioned method was developed in python on the Houdini FX Education Edition 16.5.323 package, and ran on a dual boot Linux PC with 23 GB memory and 64 bits Intel(R) Xeon(R) CPU E5-1650 0 @ 3.20 GHz CPU. The average time for deforming a primitive is 0.17 s, which ensures a smooth real-time modelling user experience.

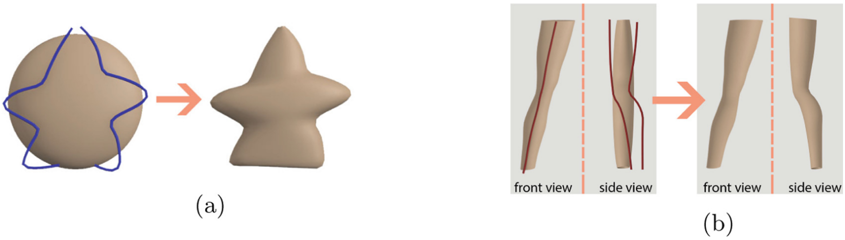


Fig. 3. ODE-driven Sketch-based deformations: (a) the deformation process of an organic shape represented by an ellipsoid and its 2D silhouette contour, and the deformed shape of the ellipsoid, (b) a leg that has been deformed by single-view sketch strokes before, now get further deformed in accordance with free form red-colored curves (Color figure online)

5 Conclusion and Future Work

In this paper, we present an ODE-driven and sketch-based modelling approach to create 3D models from 2D sketches efficiently, more examples are displayed in Fig. 4. By introducing ODE-driven geometric modelling, we proposed a primitive deformation method to deform primitives to match the input 2D silhouette contours exactly. Our approach has the advantages of: (1) easiness for beginners to use, (2) avoiding heavy manual operations, and (3) high efficiency in creating sketch-based 3D organic models.

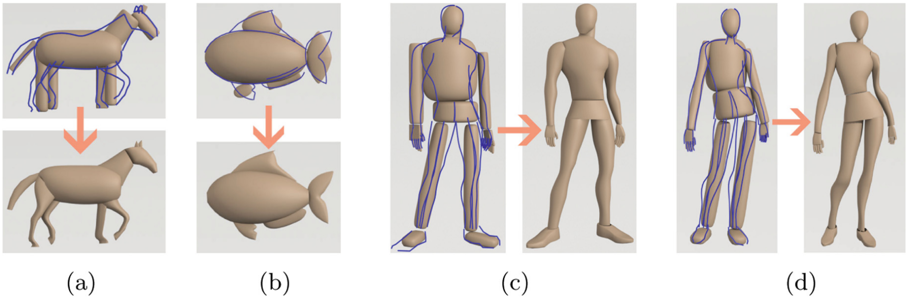


Fig. 4. Examples of the primitive deformation method

Acknowledgements. This research is supported by the PDE-GIR project which has received funding from the European Unions Horizon 2020 research and innovation programme under the Marie Skodowska-Curie grant agreement No 778035, and Innovate UK (Knowledge Transfer Partnerships Ref: KTP010860). Xiaogang Jin is supported by Science and Technology Project on Preservation of Cultural Relics, Cultural Heritage Bureau of Zhejiang Province (Grant No. 2018009) and the National Natural Science Foundation of China (No. 61732015).

References

1. Botsch, M., Sorkine, O.: On linear variational surface deformation methods. IEEE Trans. Vis. Comput. Graph. **14**(1), 213–230 (2008)
2. Celniker, G., Gossard, D.: Deformable curve and surface finite-elements for free-form shape design. ACM SIGGRAPH Comput. Graph. **25**(4), 257–266 (1991)
3. Chaudhry, E., Bian, S., Ugail, H., Jin, X., You, L., Zhang, J.J.: Dynamic skin deformation using finite difference solutions for character animation. Comput. Graph. **46**, 294–305 (2015)
4. Chaudhry, E., You, L., Jin, X., Yang, X., Zhang, J.J.: Shape modeling for animated characters using ordinary differential equations. Comput. Graph. **37**(6), 638–644 (2013)
5. Chen, T., Zhu, Z., Shamir, A., Hu, S.M., Cohen-Or, D.: 3-sweep: extracting editable objects from a single photo. ACM Trans. Graph. (TOG) **32**(6), 195 (2013)
6. Chevalier, L., Jaillet, F., Baskurt, A.: Segmentation and superquadric modeling of 3D objects (2003)
7. Do Carmo, M.P., Fischer, G., Pinkall, U., Reckziegel, H.: Differential geometry. Mathematical Models, pp. 155–180. Springer, Wiesbaden (2017). https://doi.org/10.1007/978-3-658-18865-8_10
8. Gingold, Y., Igarashi, T., Zorin, D.: Structured annotations for 2D-to-3D modeling. In: ACM Transactions on Graphics (TOG), vol. 28, p. 148. ACM (2009)
9. Igarashi, T., Matsuoka, S., Tanaka, H.: Teddy: a sketching interface for 3d freeform design. In: Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, pp. 409–416. ACM Press/Addison-Wesley Publishing Co. (1999)
10. Karpenko, O.A., Hughes, J.F.: Smoothsketch: 3D free-form shapes from complex sketches. ACM Trans. Graph. **25**(3), 589–598 (2006). https://doi.org/10.1145/1141911.1141928
11. Li, C., Pan, H., Liu, Y., Tong, X., Sheffer, A., Wang, W.: Bendsketch: modeling freeform surfaces through 2D sketching. ACM Trans. Graph. (TOG) **36**(4), 125 (2017)
12. Nealen, A., Igarashi, T., Sorkine, O., Alexa, M.: Fibermesh: designing freeform surfaces with 3D curves. ACM Trans. Graph. (TOG) **26**(3), 41 (2007)
13. Olsen, L., Samavati, F.F., Sousa, M.C., Jorge, J.A.: Sketch-based modeling: a survey. Comput. Graph. **33**(1), 85–103 (2009)
14. Shtof, A., Agathos, A., Gingold, Y., Shamir, A., Cohen-Or, D.: Geosemantic snapping for sketch-based modeling. In: Computer Graphics Forum, vol. 32, pp. 245–253. Wiley Online Library (2013)
15. Terzopoulos, D., Platt, J., Barr, A., Fleischer, K.: Elastically deformable models. In: ACM SIGGRAPH Computer Graphics, vol. 21, pp. 205–214. ACM (1987)
16. Welch, W., Witkin, A.: Variational surface modeling. In: ACM SIGGRAPH Computer Graphics, vol. 26, pp. 157–166. ACM (1992)
17. Xu, M., Li, M., Xu, W., Deng, Z., Yang, Y., Zhou, K.: Interactive mechanism modeling from multi-view images. ACM Trans. Graph. (TOG) **35**(6), 236 (2016)
18. You, L., Ugail, H., Tang, B., Jin, X., You, X., Zhang, J.J.: Blending using ode swept surfaces with shape control and c1 continuity. Vis. Comput. **30**(6–8), 625–636 (2014)

19. You, L., Yang, X., Pachulski, M., Zhang, J.J.: Boundary constrained swept surfaces for modelling and animation. In: Computer Graphics Forum, vol. 26, pp. 313–322. Wiley Online Library (2007)
20. You, L., Yang, X., You, X.Y., Jin, X., Zhang, J.J.: Shape manipulation using physically based wire deformations. Comput. Anim. Virtual Worlds **21**(3–4), 297–309 (2010)