Modelling and Simulation of Lily flowers using PDE Surfaces

Ehtzaz Chaudhry
Jian Chang
NCCA, Bournemouth University, Poole, UK
echaudhry@bournemouth.ac.uk

Hassan Ugail
Center for Visual Computing, University of Bradford, Bradford, UK
h.ugail@bradford.ac.uk

Alexander Malyshev
Department of Mathematics, University of Bergen, Bergen, Norway
alexander.malyshev@uib.no

Alfonso Carriazo
Faculty of Mathematics, University of Seville, Sevilla, Spain
carriazo@us.es

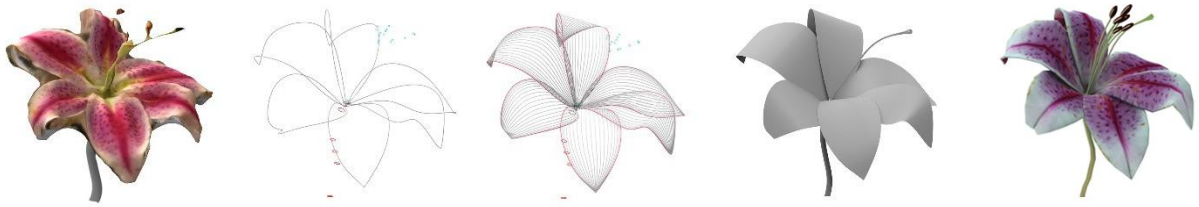
Andres Iglesias
Department of Applied Mathematics and Computational Sciences, University of Cantabria, Spain
iglesias@unican.es

Zulfiqar Habib
Allah Bux Sargano
Department of Computer Science, COMSATS University Islamabad, Lahore, Pakistan
drzhabib@cuilahore.edu.pk

Habibollah Haron
Faculty of Computing, Universiti Teknologi Malaysia, Johor Bahru, Malaysia
habib@utm.my

Algirdas Noreika
Indeform Ltd, Kaunas, Lithuania
algirdas.noreika@indeform.com

Lihua You
Jian Jun Zhang
NCCA, Bournemouth University, Poole, UK
lyou@bournemouth.ac.uk



(a) *(b)* *(c)* *(d)* *(e)*

Fig 1: Process of our method: (a) shape of lily flowers constructed from photos. (b) extracted profile curves describing various part of lily flowers. (c) PDE-based surface creation. (d) geometrical model constructed from our method. (e) refined textured model.

***Abstract*— This paper presents a partial differential equation (PDE)-based surface modelling and simulation framework for lily flowers. We use a PDE-based surface modelling technique to represent shape of a lily flower and PDE-based dynamic simulation to animate blossom and decay processes of lily flowers. To this aim, we first automatically construct the geometry of lily flowers from photos to obtain feature curves. Second, we apply a PDE-based surface modelling technique to generate sweeping surfaces to obtain geometric models of the flowers. Then, we use a physics-driven and data-based method and introduce the flower shapes at the initial and final positions into our proposed dynamic deformation model to generate a realistic deformation of flower blossom and decay. The results demonstrate that our proposed technique can create realistic flower models and their movements and shape changes against time efficiently with a small data size.**

Keywords*— *Flower modelling, sweeping surfaces, flower blossom, flower decay, deformations, partial differential equations, finite difference solution, data-driven methods.

I. INTRODUCTION

The shape representation of flowers is one of the most challenging tasks due to the complex multi-layer structures (petals, stigma, and stems) which experience geometrical deformations such as bending, stretching, shrinking and curling. Modelling complex flower shapes and simulating their movements and deformations are usually a laboursome and expensive task. The challenge is how to simplify the modelling task, create 3D models quickly, and animate them realistically and efficiently.

Although various techniques (Data-driven, Sketch-based, Point-based and Image-based) for flower modelling are very popular, they face some challenges such as geometry of high fidelity and missing-captured data. Data-driven techniques introduce example shapes which are used for interpolation. The more example shapes, the more realistic the results. One of the challenges for this approach is that it relies heavily on key example shapes to derive a generalisation of deformation, and this becomes its major disadvantage since it is an expensive and time-consuming process to create many key example shapes. Sketch-based techniques map 2D sketches to a 3D modelling operation [1]. The process is difficult and ambiguous, and requires perception and skills of animators. Point-based techniques use a laser scan to reconstruct the geometry. Scanning a flower geometry using standard scanning techniques

978-1-7281-2741-5/19/$31.00 ©2019 IEEE

is the most challenging task because flower consists of multi-layer complex topology with a significant amount of self-collision with the entangled petals, stems and interior parts. Image-based techniques use a set of photographs to build 3D models. There is a lot of work on surface reconstruction and various techniques have been developed. In film and game production, facial markers, camera arrays, and structured light projectors are used to obtain 3D geometry of high fidelity. However, the results of the point cloud may be noisy and incomplete.

As described above, each method has its strengths and weaknesses and there is no a single framework for flower modelling and simulation that can meet all the requirements of animation industry. 3D scanning is the most popular technique to represent flower shapes. However, it suffers from various imperfections including noise, missing data, and outliers [2]. The scanned data require a lot of work to refine the mesh structure, remove noise and fill holes caused by occlusion in scanning [2]. Such a model may lack flexibility for further editing, designing and recreation. Moreover, it requires special knowledge and skills of a modeller because it is a laborious and time-consuming task. The scanned 3D models have a large unstructured data size and are usually not suitable for animation directly. A skilful modeller has to remodel or reduce the size of scanned 3D models for animation. This is a relentless and tedious task.

To overcome these challenges, we present a PDE-based surface modelling and blossom and decay animation technique of lily flowers. It can create physically realistic deformable surfaces. The greatest advantage of our technique is that it provides a single framework for shape modelling and simulation which uses PDE-based surface modelling technique to represent the geometry of a natural flower shape and uses PDE-driven and data-based dynamic simulation to animate natural flower blossom and decay process.

In the following sections, we first introduce the related work on flower modelling and simulation followed by shape reconstruction of real lily flowers. Section 4 describes PDE surface-based modelling. Section 5 investigates a PDE-driven and data-based simulation approach for flower blossom and decay. Section 6 gives experimental results. Finally, Section 7 concludes the work given in this paper.

II. Related Work

The existing work on flower modelling and simulation can be divided into the following aspects.

A. Flower Modelling

Zhang et al. [3] presented a Data-driven modelling approach for flower petal fitting to handle occluded shape and maintain correct 3D spatial relations. This approach builds a scale-invariant morphable model of flower petal shapes from different flower species. It faces two main challenges. First, it relies on finding correspondence variations of a flower species and requires capturing real-world data [3]. Second, this technique only focuses on the parametric modeling of flower petals and does not consider other intricate structures of flowers such as stigma, and stems. Sketch-based modelling of flowers was introduced by Ijiri et al. [4] in which the user can easily model flowers, trees and land using freehand sketches. The biggest challenge which this technique faces is how to interprete the sketch shapes and map them into 3D objects. Ijiri et al. [5] introduced a flower modelling system that supports seamless transformations from an initial sketch to a detailed three-dimensional (3D) model. The system heavily relies on an artist's skill to draw nature flower shapes and does not facilitate fine-tuning of a user. Later, Ding et al. [6] presented sketch-based solid modelling of flowers. This approach was based on the work by Ijiri et al. [7]. The work focuses on interactive sketching interfaces for designing 3D objects using 2D sketches and implicit gestures to allow users to freely and easily edit the geometrical parameters directly on floral diagrams and inflorescence diagrams using the predefined implicit gestures. Petrenko et al. [8] proposed a point-based approach for Interactive flower modelling with 3Gmap L-systems. In recent years, Paulus et al. [9] used a laser scanning system for three-dimensional modelling of saffron flowers. Image-based modelling was investigated by Yan et al. [10]. The technique reconstructs flower models from a single photograph. It significantly requires user interaction for realism.

B. Flower Simulation

Lu et al. [11] presented an approach for simulation of opening flowers based on deformation which emulates its growth by changing the deformation parameters. Qin et al. [12] introduced a flower model using an L-system and Bezier surfaces. Ijiri et al. [13] presented simulation of blooming flowers using an elastic triangular mesh to represent petals and simulate their growth in a semiautomatic way according to user-specified parameters. Li et al. [14] introduced a physics-based simulation method for flower blossom. This method employs fewer growth parameters and follows boundary-dominant blooming mechanism. All the techniques, described above require user to specify parameters to obtain reasonable simulation results. Hence, they cannot produce realistic effects such as wrinkles and shrinking. Recently, Zheng et al. [15] proposed a template-based dynamic tracking algorithm to capturing blooming flower sequences from 3D scanners. This technique provides the motion information for different petals, which is represented as a set of transformations. This allows users to easily manipulate flower blooming motion for special effects. However, a complex flower with densely packed petals, such as (Rose and Peony) is the main challenge which this technique faces because the scanned data of these complex flowers are incomplete.

C. Plant/Tree Modeling and Simulation

Some researchers provided the methods in plant/tree modeling and simulation. Quan et al. [16] proposed a semi-automatic image based plant modelling. The 3D model can be reconstructed from multiple views and images. However, this method is dependent on the user segmentation. Tan et al. [17] presented an image based natural-looking tree generation method without the need for user intervention. Li et al. [18] proposed a method for detecting bifurcations of indoor plants from 4D point cloud data. The problem was the complexity caused by the changing of point cloud density and occlusions among structures. Livny et al. [19] presented an automatic method to reconstruct tree skeletal structures from point clouds. L-systems, presented by Prusinkiewicz and Lindenmayer et al.

[20] provide a way to model organic elements such as trees and flowers and animate their growth.

Physically-based tree animation and leaf deformation using CUDA in real time were proposed by Yang et al. [21]. Interactive authoring of simulation-ready plants was investigated by Zhao et al. [22]. A venation skeleton-based method for modelling and animating plant leaf wilting was examined by Lu et al. [23]. Kider et al. [24] presented a physically-based approach to simulate the biological aging and decay process in fruits. The method uses some numerical reaction diffusion equations to form fungal and bacterial colonies, and a transpiration model to drive the volume collapse. These calculations affect the underlying maps and substrate meshes as the fruit decays. Jeong et al. [25] explored the changes in morphology drying and presented a triangle-based double-layer structure to model the fine wrinkles caused by inhomogeneous shrinkage due to the water loss. The osmotic water flow simulates the internal water flow through leaves so that dehydrated regions of drying leaves advance forward the veins.

III. Shape Reconstruction of Real Lily Flowers

The structure of lily flowers is shown in Figure 2. In order to create highly realistic animation of blossom and decay of lily flowers, it is important to generate high-quality and detailed flower models. Reconstruction from real images provides an effective method to obtain high-quality flower models. This section will discuss the capture of flower shape, bloom, decay and texture changes using time-lapse video and reconstruction of lily flowers from the captured video.

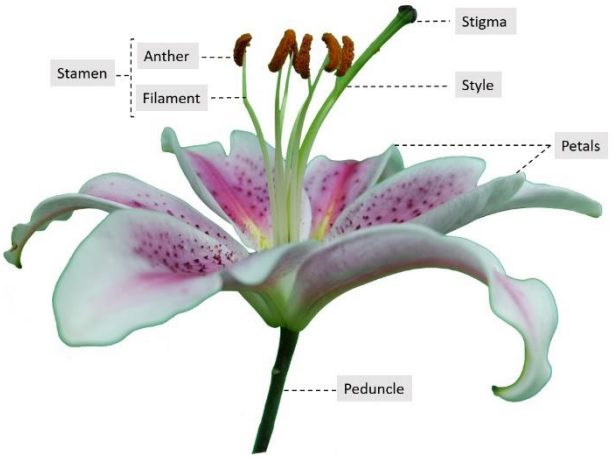


Fig. 2. Structure of lily flowers

A. Capture of lily flowers with time-lapse video

As shown in Figure 3, various cameras were used to record a time-lapse video from front and top view, and 20 photos from different views were selected to define the shapes of three flower development stages: bud, bloom and decay. The time-lapse video helps us to study the shape changes associated with flower parts and to understand the simulation of lily flowers.

B. Reconstruction of lily flowers from photos

After capturing the time-lapse video at the three flower development stages, the photos were identified and input Autodesk Recap to automatically reconstruct the 3D polygon models of lily flowers at the three development stages.

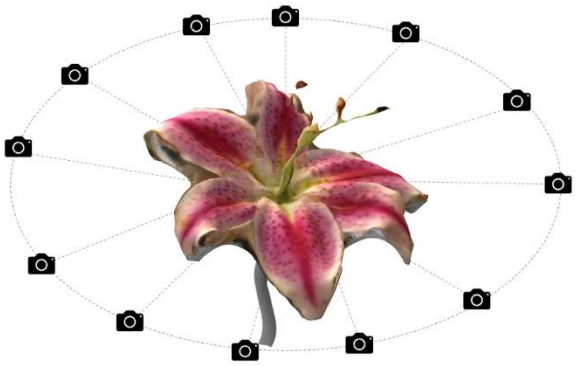


Fig. 3. Reconstruction of flowers from photos

Figure 4a and 4b show the 3D model reconstructed by Autodesk ReCap. This base geometry is incomplete with large missing parts due to self-occlusions and inaccessible parts.

a b

Fig. 4. Reconstruction of flowers from photos

C. Obtaining Curve Defined Model

As pointed out above, the reconstructed geometry of lily flowers is incomplete with large missing parts. In addition, the reconstructed geometry is represented with polygon models which involves a large data size causing large storage requirements and slow simulation efficiency. In order to tackle these problems, we transform the reconstructed polygon models into PDE surfaces.

PDE surfaces are defined by boundary curves and some other boundary constraints such as boundary tangents. Extracting boundary curves are an essential step of transforming the polygon models into PDE surfaces.

There are two approaches which can be used to extract boundary curves from the reconstructed flower mesh. One is to draw an outline curves on the built 3D model, and the other is to project curves onto the built 3D model as used by Olsen et al. [1]. In this paper, the first method is used to obtain boundary curves of the reconstructed flower geometry. Figure 5 shows the result of extracted boundary curves from the reconstructed flower geometry.

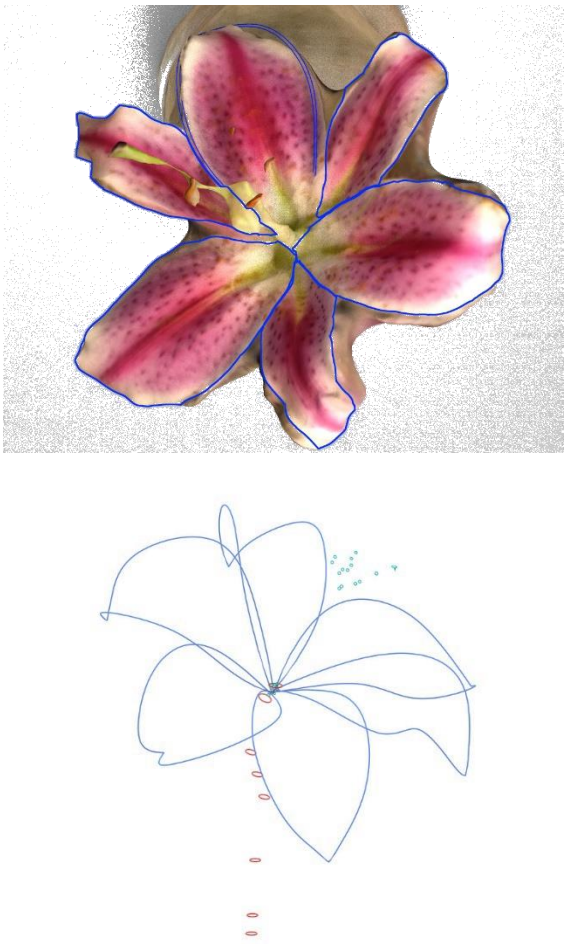


Fig. 5. Curve Defined Model

IV. PDE SURFACE-BASED MODELING

PDE surfaces have the ability of describing a complicated shape with a single PDE patch. Therefore, representing 3D models with PDE surfaces can greatly reduce the data size of geometric modelling. In this section, we investigate PDE surface-based shape modelling of lily flowers.

In the field of computer graphics, physics-based deformations are normally based on an elastic energy of thin shells [28]. As described in [29], when a surface S parameterized by a function $P(u,v)$ is deformed to a new shape S' through adding a displacement vector $d(u,v)$ to each point $P(u,v)$, the change of the first and second fundamental forms $\mathrm{I}(u,v)$ and $\Pi(u,v)$ in differential geometry [30] leads to a measure of stretching and bending.

$$E_{shell}(S') = \int_\Omega k_s \|\mathrm{I}' - \mathrm{I}\|_F^2 + k_b \|\Pi' - \Pi\|_F^2 \, du\, dv \quad (1)$$

In the above equation, I' and Π' are the first and second fundamental forms of the surface S', $\|\cdot\|_F$ indicates a (weighted) Frobenius norm, and the stiffness parameters k_s and k_b are used to control the resistance to stretching and bending.

The minimization of the above equation results in a new deformed surface. The minimization operation is nonlinear and computationally expensive. In order to avoid the nonlinear minimization, the first and second order partial derivatives of the displacement function $d(u,v)$ are introduced to replace the change of the first and second fundamental forms [31, 32], i. e.,

$$\tilde{E}_{shell}(d) = \int_\Omega k_s(\|d_u\|^2 + \|d_v\|^2) + k_b(\|d_{uu}\|^2 + 2\|d_{uv}\|^2 + \|d_{vv}\|^2)\, du\, dv \quad (2)$$

where $d_x(u,v) = \frac{\partial}{\partial x} d(u,v)$ and $d_{xy} = \frac{\partial^2}{\partial x \partial y} d(u,v)$.

Applying variational calculus to Equation (2), the minimization of the above equation is changed into the following Euler-Lagrange partial differential equation (PDE)

$$-k_s \Delta d + k_b \Delta^2 d = 0 \quad (3)$$

where Δ and Δ^2 are the Laplacian and the bi-Laplacian operator, respectively.

$$\Delta d = div \nabla d = d_{uu} + d_{vv}$$
$$\Delta^2 d = \Delta(\Delta d) = d_{uuuu} + 2d_{uuvv} + d_{vvvv} \quad (4)$$

The solution to Equation (3) represents a 3D surface which can be regarded as a set of sweeping a 3D curve called as a generator along boundary curves called trajectories. If Equation (3) is used to describe the generator, the parametric variable v in Equation (3) drops, and we have $d_{vv} = 0$ and $d_{vvvv} = 0$. Substituting $d_{vv} = 0$ and $d_{vvvv} = 0$ into Equation (3), we obtain the following simplified version of the Euler-Lagrange partial differential equation (PDE)

$$k_b \frac{\partial^4 d}{\partial u^4} - k_s \frac{\partial^2 d}{\partial u^2} = 0 \quad (5)$$

The solution to the above equation defines the generator. Since k_b and k_s greatly influences the shape of the generator, they are called shape control parameters.

After obtaining the equation of defining the generator, the next step is to define the trajectories (boundary curves) and other related boundary constraints. They can be obtained with the method below.

When two sweeping surfaces are connected together, the smooth transition between the two connected sweeping surfaces requires them to share not only the same boundary curves but also the same tangents on the boundary curves.

As shown in Figure 5, the two boundary curves $\mathbf{C}_1(v)$ and $\mathbf{C}_2(v)$ are known, we generate one more control curve next to each of the two boundary curves and obtain the boundary tangents at the boundary curve from the boundary curve and control curve. The obtained boundary tangents are indicated by $\mathbf{T}_1(v)$ and $\mathbf{T}_2(v)$.

With the above treatment, the four constraint equations at the two boundaries can be written as [26]:

$$u=0:\mathbf{S}(0,v)=\mathbf{C}_1(v):\frac{\partial \mathbf{S}(0,v)}{\partial u}=\mathbf{T}_1(v)$$
$$u=1:\mathbf{S}(1,v)=\mathbf{C}_2(v):\frac{\partial \mathbf{S}(1,v)}{\partial u}=\mathbf{T}_2(v) \quad (6)$$

where $\mathbf{C}_1(v)$ and $\mathbf{C}_2(v)$ are boundary curves at $u=0$ and $u=1$, respectively, and $\mathbf{T}_1(v)$ and $\mathbf{T}_2(v)$ are boundary tangents at the two boundary curves.

Depending on different combinations of the shape control parameters k_b and k_s, the three different solutions to ODE (2) can be obtained below.

Solution 1: When $k_s = 0$, the solution to Equation (5) can be written as the following form

$$\boldsymbol{d}(u) = \boldsymbol{e}_1 + \boldsymbol{e}_2 u + \boldsymbol{e}_3 u^2 + \boldsymbol{e}_4 u^3 \quad (7)$$

Solution 2: When $k_s \neq 0$ and $k_b k_s > 0$, the solution to ODE (2) has the form of

$$\boldsymbol{d}(u) = \boldsymbol{e}_1 + \boldsymbol{e}_2 u + \boldsymbol{e}_3 e^{\gamma u} + \boldsymbol{e}_4 e^{-\gamma u} \quad (8)$$

where

$$\gamma = \sqrt{k_s/k_b}$$

Solution 3: When $k_s \neq 0$ and $k_b k_s < 0$, the solution to ODE (2) becomes

$$\boldsymbol{d}(u) = \boldsymbol{e}_1 + \boldsymbol{e}_2 u + \boldsymbol{e}_3 \cos(au) + \boldsymbol{e}_4 \sin(au) \quad (9)$$

where

$$\gamma = \sqrt{-k_s/k_b} \quad (10)$$

Substituting each of the equations (6), (7) and (9) into the boundary constraint equation (6), we can determine all the four unknown constants $\boldsymbol{e}_1$, $\boldsymbol{e}_2$, $\boldsymbol{e}_3$, and $\boldsymbol{e}_4$.

For example, we substitute Eq. (7) into the boundary constraint equation (6), and obtain the four known constants below

$$\begin{aligned} \boldsymbol{e}_1 &= \boldsymbol{C}_1(v) \\ \boldsymbol{e}_2 &= \boldsymbol{T}_1(v) \\ \boldsymbol{e}_3 &= \boldsymbol{C}_1(v) - \boldsymbol{C}_2(v) + \boldsymbol{T}_2(v) \\ \boldsymbol{e}_4 &= 2[\boldsymbol{C}_1(v) - \boldsymbol{C}_2(v)] + \boldsymbol{T}_1(v) + \boldsymbol{T}_2(v) \end{aligned} \quad (11)$$

Substituting Eq. (11) into Eq. (7), the mathematical expression for the sweeping surface which satisfies both Equation (5) and the boundary constraint equation (6) is obtained as

$$\begin{aligned} \boldsymbol{d}(u,v) = {} & (1+u^2+2u^3)\boldsymbol{C}_1(v) + u(1+u^2)\boldsymbol{T}_1(v) \\ & -u^2(1+2u)\boldsymbol{C}_2(v) + u^2(1+u)\boldsymbol{T}_2(v) \end{aligned} \quad (12)$$

With the above PDE surface-based modelling, each part of the flower such as petals, stigma and stamp is created separately with one sweeping surface and the whole flower model is built by assembling these parts together. Figure 6 shows one lily flower created with the sweeping surfaces.

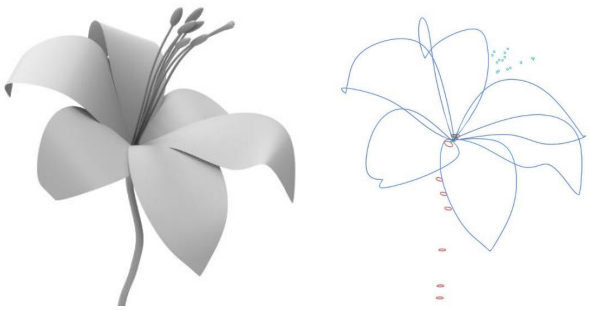


Fig. 6. Geometrical constructed model

V. PDE-DRIVEN AND DATA-BASED SIMULATION OF FLOWER BLOSSOM AND DECAY

The bud, bloom and decay processes of lily flowers involve shape changes of the sweeping surfaces. As discussed above, the sweeping surfaces are generated by sweeping a generator. Therefore, the shape changes of the sweeping surfaces can be transformed into the shape changes of the generator. The shapes of the generator at different positions are shown in Figure 7.

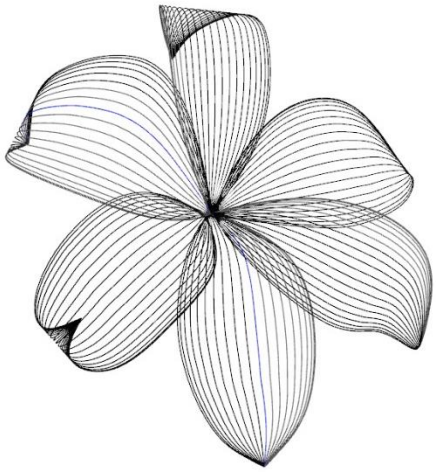


Fig. 7. Shapes of the generator at different positions

Once the shapes of the generator at different positions are obtained, we can simulate their changes with time. By doing so, simulating the shape changes of the PDE surface-based flower model is changed into simulating the changes of the shapes of the generator at different positions. With this method, we can animate a flower model efficiently to simulate the bud, bloom and decay processes.

As investigated in [27], the deformations of a curve can be described with a dynamic partial differential equation. This can be achieved by introducing a time variable t into Eq. (5), considering the initial forces $m\frac{\partial^2 \boldsymbol{d}(u,t)}{\partial v^2}$, and adding the action

of the external force $\mathbf{f}(u,t)$ which change Eq. (5) into the following form

$$k_b \frac{\partial^4 \boldsymbol{d}(u,t)}{\partial u^4} - k_s \frac{\partial^2 \boldsymbol{d}(u,t)}{\partial u^2} + m \frac{\partial^2 \boldsymbol{d}(u,t)}{\partial t^2} = \boldsymbol{f}(u,t) \qquad (13)$$

where m is the mass, and $\mathbf{f}(u,t)$ is the external force.

At the initial position $t=0$, the lily flower has no deformations, i. e., $\boldsymbol{d}_0(u) = 0$. At the final position $t=1$, the deformations of the lily flower are $\boldsymbol{d}_1(u)$. At the initial position $t=0$, the deformation rates are zero, i. e., $\partial \boldsymbol{d}_0(u)/\partial t = 0$. Therefore, the following conditions must be applied when solving the dynamic equation (13)

$$\begin{aligned} t = 0 \quad & \mathbf{d}(u,0) = \mathbf{d}_0(u) = 0 \qquad \partial \boldsymbol{d}(u,0)/\partial t = \mathbf{0} \\ t = 1 \quad & \mathbf{d}(u,1) = \mathbf{d}_1(u) \end{aligned} \qquad (14)$$

The solution of the dynamic equation (13) subjected to the conditions (14) is complicated. Here we use the finite difference method to solve it.

Among various finite difference approximations, the central finite difference approximation is the most accurate. Since Eq. (13) involves both the parametric variable u and the time variable t, we have to discretize them separately, i. e., $\Delta u = 1/I$ and $\Delta t = 1/J$. At the time instant $t_j = j\Delta t = j/J$ and the node $u_i = i\Delta u = i/I$, the central finite difference approximations for the first, second and fourth derivatives with the parametric variable u at the time instant $t_{j+1} = (j+1)\Delta t = (j+1)/J$ and the node u_i can be written as

$$\begin{aligned} \left(\frac{\partial \boldsymbol{d}}{\partial u}\right)_i^{j+1} &= \frac{\boldsymbol{d}_{i+1}^{j+1} - \boldsymbol{d}_{i-1}^{j+1}}{2\Delta u} = 0.5I\left(\boldsymbol{d}_{i+1}^{j+1} - \boldsymbol{d}_{i-1}^{j+1}\right) \\ \left(\frac{\partial^2 \boldsymbol{d}}{\partial u^2}\right)_i^{j+1} &= \frac{\boldsymbol{d}_{i-1}^{j+1} - 2\boldsymbol{d}_i^{j+1} + \boldsymbol{d}_{i+1}^{j+1}}{\Delta u^2} \\ &= I^2\left(\boldsymbol{d}_{i-1}^{j+1} - 2\boldsymbol{d}_i^{j+1} + \boldsymbol{P}_{i+1}^{j+1}\right) \\ \left(\frac{\partial^4 \boldsymbol{d}}{\partial u^4}\right)_i^{j+1} &= \frac{6\boldsymbol{d}_i^{j+1} - 4(\boldsymbol{d}_{i-1}^{j+1} + \boldsymbol{d}_{i+1}^{j+1}) + \boldsymbol{d}_{i-2}^{j+1} + \boldsymbol{d}_{i+2}^{j+1}}{\Delta u^4} \\ &= I^4\left[6\boldsymbol{d}_i^{j+1} - 4(\boldsymbol{d}_{i-1}^{j+1} + \boldsymbol{d}_{i+1}^{j+1}) + \boldsymbol{d}_{i-2}^{j+1} + \boldsymbol{d}_{i+2}^{j+1}\right] \end{aligned} \qquad (15)$$

and the first and second derivatives with the time variable t at the time instant $t_j = j\Delta t = j/J$ and the node u_i can be written as

$$\begin{aligned} \left(\frac{\partial \boldsymbol{d}}{\partial t}\right)_i^{j} &= \frac{\boldsymbol{d}_i^{j+1} - \boldsymbol{d}_i^{j-1}}{2\Delta t} = 0.5J\left(\boldsymbol{d}_i^{j+1} - \boldsymbol{d}_i^{j-1}\right) \\ \left(\frac{\partial^2 \boldsymbol{d}}{\partial t^2}\right)_i^{j} &= \frac{\boldsymbol{d}_i^{j-1} - 2\boldsymbol{d}_i^{j} + \boldsymbol{d}_i^{j+1}}{\Delta t^2} \\ &= J^2\left(\boldsymbol{d}_i^{j-1} - 2\boldsymbol{d}_i^{j} + \boldsymbol{d}_i^{j+1}\right) \end{aligned} \qquad (16)$$

Substituting the last two of Eq. (15) and the last one of Eq. (16) into Eq. (13), we obtain the following finite difference equation of the dynamic equation (13)

$$\begin{aligned} & k_b I^4\left(\boldsymbol{d}_{i-2}^{j+1} + \boldsymbol{d}_{i+2}^{j+1}\right) - I^2(4k_b I^2 + k_s)\boldsymbol{d}_{i-1}^{j+1} + (6k_b I^4 + 2k_s I^2 \\ & + mJ^2)(\boldsymbol{d}_{i-1}^{j+1} + \boldsymbol{d}_{i+1}^{j+1}) + mJ^2(\boldsymbol{d}_i^{j-1} - 2\boldsymbol{d}_i^{j}) = \boldsymbol{f}_i^{j+1} \\ & (i = 1,2,3,\cdots I-1\,; j = 0,1,2,3,\cdots, J-1) \end{aligned} \qquad (17)$$

When using Eq. (17) to write the finite difference equation for the nodes i=1, and i=I-1, the nodes -1, 0, I, and I+1 will be involved. When using Eq. (17) to write the finite difference equation for the nodes i=2, and i=I-2, the nodes 0, and I will be involved. Therefore, before solving Eq. (17), the deformations at the nodes i=-1, i=0, i=I, and i=I+1 must be determined first. Since $\boldsymbol{d}_0(u)$ and $\boldsymbol{d}_1(u)$ are known, we can determine $\partial \boldsymbol{d}_0/\partial u$ and $\partial \boldsymbol{d}_1/\partial u$. Then we use the first of Eq. (16) to obtain the following deformations of the nodes -1 and I+1 at the initial position j=0 (t=0) and the final position j=J (t=1)

$$\begin{aligned} \boldsymbol{d}_{-1}^0 &= \boldsymbol{d}_1^0 - \frac{2}{I}\left[\frac{\partial \boldsymbol{d}_{0(u)}}{\partial u}\right]_{u=0} \\ \boldsymbol{d}_{I+1}^0 &= \boldsymbol{d}_{I-1}^0 + \frac{2}{I}\left[\frac{\partial \boldsymbol{d}_{0(u)}}{\partial u}\right]_{u=1} \\ \boldsymbol{d}_{-1}^J &= \boldsymbol{d}_1^J - \frac{2}{I}\left[\frac{\partial \boldsymbol{d}_{1(u)}}{\partial u}\right]_{u=0} \\ \boldsymbol{d}_{I+1}^J &= \boldsymbol{d}_{I-1}^J + \frac{2}{I}\left[\frac{\partial \boldsymbol{d}_{1(u)}}{\partial u}\right]_{u=1} \end{aligned} \qquad (18)$$

The deformations at the nodes 0, 1, I-1, and I at any time instant variable t can be determined by linearly interpolating the nodes at the time instants $t=0$ and $t=1$ which leads to the following equation

$$\begin{aligned} \boldsymbol{d}_k^j &= \boldsymbol{d}_k^0 + (\boldsymbol{d}_k^J - \boldsymbol{d}_k^0)t \\ (k = -1,0,I,I+1\,;\;\; & 0 \le t \le 1) \end{aligned} \qquad (19)$$

After knowing the deformations at the nodes i=-1, i=0, i=I, and i=I+1, Eq. (17) can be solved with the following method. First, we set j=0 and $J=1$ and change Eq. (17) into

$$\begin{aligned} & k_b I^4(\boldsymbol{d}_{i-2}^1 + \boldsymbol{d}_{i+2}^1) - I^2(4k_b I^2 + k_s)\boldsymbol{d}_{i-1}^1 + (6k_b I^4 + \\ & 2k_s I^2 + m)(\boldsymbol{d}_{i-1}^1 + \boldsymbol{d}_{i+1}^1) + m(\boldsymbol{d}_i^{-1} - 2\boldsymbol{d}_i^0) = \boldsymbol{f}_i^1 \\ & (i = 1,2,3,\cdots I-1) \end{aligned} \qquad (20)$$

According to the first of Eq. (14), when $=0$, $\boldsymbol{d}_i^j = \boldsymbol{d}_i^0 = \boldsymbol{d}(u_{\,i}, 0) = \boldsymbol{d}_0(u_i)$ is known, but $\boldsymbol{d}_i^{j-1} = \boldsymbol{d}_i^{-1}$ is unknown. From the second of Eq. (14) and the first of Eq. (16), we have $[\partial \boldsymbol{d}(u,0)/\partial t]_i^0 = 0.5J(\boldsymbol{d}_i^1 - \boldsymbol{d}_i^{-1}) = 0$ which gives $\boldsymbol{d}_i^{-1} = \boldsymbol{d}_i^1$. From the third of Equation (14), $\boldsymbol{d}_i^1 = \mathbf{d}(u_i, 1) = \mathbf{d}_1(u_i)$ ($i = 0,1,2,3,\cdots I-1, I$) are also known. We also have $\boldsymbol{d}_0^1 = \mathbf{d}_1(u=0)$, $\boldsymbol{d}_I^1(u=1)$, $\boldsymbol{d}_{-1}^1 = \boldsymbol{d}_{-1}^J$, and $\boldsymbol{d}_{I+1}^1 = \boldsymbol{d}_{I+1}^J$. Therefore, we can us equation (18) to obtain the forces $\boldsymbol{f}_i^1$.

Setting J to a new value larger than 1 and introducing the obtained $\boldsymbol{f}_i^1$ and the deformations (19) at the nodes -1, 0, I, and I+1 into Equation (17), the deformations at the nodes $(i = 1,2,3,\cdots I-1)$ at $t_j = j\Delta t$ can be determined by solving Equation (17).

If the obtained deformations $\boldsymbol{d}_i^J$ at j=J (t=1) are not very close to the real deformations $\mathbf{d}_1(u_i)$, the forces $\boldsymbol{f}_i^1$ are replaced by the modified forces below

$$\boldsymbol{f}_i^n = \frac{\mathbf{d}_1(u_i)}{\boldsymbol{d}_i^J}\boldsymbol{f}_i^1 \tag{21}$$

The modified forces are introduced Eq. (17) to obtain a closer approximation to the real deformations. This process is repeated until the simulated deformations at the final position are very close to the real deformations. That is to say, the errors between the simulated deformations at the final position and the real deformations are less than a specified value.

VI. Experimental results

In this section, we present some examples to demonstrate our proposed approaches. In Subsection A, the shapes of a lily flower at five different time instants ere reconstructed. In Subsection B, our developed PDE-driven and Data-based simulation method is used to simulate the blossom and decay of the lily flower.

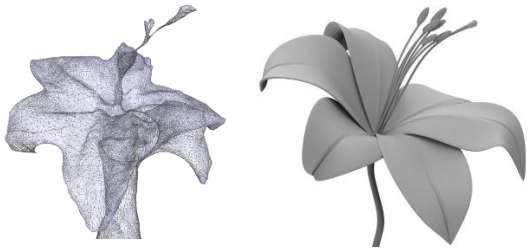
A. Modelling of lily flowers

The blossom process of two lily flowers (Lily flower A and Lily flower B) were captured. The method developed in Section III was used to reconstruct the 3D polygon models of the lily flowers at the five different time instants from the captured photos. The reconstructed lily flower models are transformed into the PDE surfaces with the method developed in Section IV. The transformed PDE surfaces of the lily flower are shown in Figure 8 where the top row is for Lily flower A and the bottom row is for Lily flower B.

a b c d e

Fig. 8. PDE surface representations of Lily flower blossom sequence

Next, we demonstrate the small data size of the proposed PDE surface representation with the flower models shown in Figure 9. The polygon model (Fig. 9a) of the Lily flower reconstructed by Autodesk ReCap has 302,123 vertices. In contrast, the PDE surface-represented model (Fig. 9b) only has 6,024 design variables.



a
Flower shape reconstructed by Autodesk ReCap
No of vertices: 302,123

b
Flower shape reconstructed by PDE
No of design variables: 6,024

Fig. 9. Data size comparison

The above discussions indicate that the proposed PDE surface-based modelling method generates realistic lily flower shapes with a small data size.

We also timed the process of transforming the reconstructed polygon models into PDE surfaces. The total CPU time used to obtain the PDE surfaces of the lily flowers shown in Figure 8 is 0.032 ms. It indicates that our proposed PDE surface-based modelling is highly efficient.

B. Simulation of lily flowers

For simulation of lily flower blossom and decay, the initial time instant and the last time instant were taken as the initial position and the final position. The PDE surfaces of the lily flowers at the two positions are used to obtain the deformations $\mathbf{d}_0(u)$ and $\mathbf{d}_1(u)$ required in Eq. (14). Here we use lily flower decay to demonstrate the simulation result.

The real flower shapes at the initial and final positions represented with PDE surfaces are shown in the first and fifth figures of Figure 10. The deformations $\mathbf{d}_0(u)$ and $\mathbf{d}_1(u)$ required in Eq. (14) were obtained from the first and fifth figures of Figure 10. The PDE-driven and Data-based simulation method was used to generate the shapes at the second, third, and fourth time instants shown in the second, third, and fourth figures of Figure 10. The simulated shapes shown in Figure 10b, 10c, and 10d indicate the proposed PDE-driven and data-driven simulation method can create realistic shapes.

a b c d e

Fig. 10. Flower decay sequence (In-between flower shapes 2,3 and 4 generated by our method)

VII. Conclusions

In this paper, we have developed a new PDE surface-based modelling and PDE-driven and Data-based simulation technique. The developed PDE-surface based modelling reduces a two-dimensional modelling problem to a one-dimensional modelling problem to greatly simplify the mathematical

complexity. The developed PDE-driven and data-based simulation combines physics-driven simulation with data-based methods to achieve both good realism and high computational efficiency.

The experiment results demonstrate the effectiveness and advantages of the proposed PDE surface-based modelling and PDE-driven and data-based simulation. Apart from generating realistic 3D lily flower models with a small data size by the developed PDE surface-based modelling, the developed PDE-driven and data-based simulation can create realistic animation of lily flower blossom and decay efficiently.

ACKNOWLEDGMENT

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).

REFERENCES

[1] Olsen, L., Samavati, F.F., Sousa, M.C. and Jorge, J.A., 2008, April. A Taxonomy of Modeling Techniques using Sketch-Based Interfaces. In Eurographics (pp. 39-57).

[2] Tagliasacchi, A., Zhang, H., & Cohen-Or, D. (2009, July). Curve skeleton extraction from incomplete point cloud. In ACM Transactions on Graphics (TOG) (Vol. 28, No. 3, p. 71). ACM.

[3] Zhang, C., Ye, M., Fu, B. and Yang, R., 2014. Data-driven flower petal modeling with botany priors. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 636-643).

[4] Ijiri, T., Igarashi, T., Takahashi, S. and Shibayama, E., 2004, August. Sketch interface for 3d modeling of flowers. In ACM SIGGRAPH 2004 Sketches (p. 6).

[5] Ijiri, T., Owada, S. and Igarashi, T., 2006, September. Seamless integration of initial sketching and subsequent detail editing in flower modeling. In Computer Graphics Forum (Vol. 25, No. 3, pp. 617-624).

[6] Ding, Z., Xu, S.C., Ye, X.Z., Zhang, Y. and Zhang, S.Y., 2008. Flower solid modeling based on sketches. Journal of Zhejiang University-SCIENCE A, 9(4), pp.481-488.

[7] Ijiri, T., Owada, S., Okabe, M. and Igarashi, T., 2005, July. Floral diagrams and inflorescences: interactive flower modeling using botanical structural constraints. In ACM Transactions on Graphics (TOG) (Vol. 24, No. 3, pp. 720-726). ACM.

[8] Petrenko, O., Terraz, O., Sbert, M. and Ghazanfarpour, D., 2011. Interactive flower modeling with 3gmap l-systems. In Proceedings of the 21st International Conference on Computer Graphics and Vision (pp. 20-24).

[9] Paulus, S., Schumann, H., Kuhlmann, H. and Léon, J., 2014. High-precision laser scanning system for capturing 3D plant architecture and analysing growth of cereal plants. Biosystems Engineering, 121, pp.1-11.

[10] Yan F., Gong M., Cohen-Or D., Deussen O., Chen B.: Flower Reconstruction From A Single Photo. Computer Graphics forum 3 3, 2 (2014).

[11] Lu, L., Wang, L. and Yang, X.D., 2008, December. A flower growth simulation based on deformation. In 2008 International Symposium on Information Science and Engineering (Vol. 2, pp. 216-218). IEEE.

[12] Qin, P. and Chen, C., 2006. Simulation model of flower using the integration of L-systems with Bezier surfaces. International Journal of Computer Science and Network Security, 6(2), pp.65-68.

[13] Ijiri, T., Yokoo, M., Kawabata, S. and Igarashi, T., 2008, May. Surface-based growth simulation for opening flowers. In Proceedings of Graphics Interface 2008 (pp. 227-234).

[14] Li, J., Liu, M., Xu, W., Liang, H. and Liu, L., 2015. Boundary‐dominant flower blooming simulation. Computer Animation and Virtual Worlds, 26(3-4), pp.433-443.

[15] Zheng, Q., Fan, X., Gong, M., Sharf, A., Deussen, O. and Huang, H., 2016, August. 4D Reconstruction of Blooming Flowers. In Computer Graphics Forum.

[16] Quan, L., Tan, P., Zeng, G., Yuan, L., Wang, J. and Kang, S.B., 2006, July. Image-based plant modeling. In ACM Transactions on Graphics (TOG) (Vol. 25, No. 3, pp. 599-604). ACM.

[17] Tan, P., Zeng, G., Wang, J., Kang, S.B. and Quan, L., 2007, August. Image-based tree modeling. In ACM Transactions on Graphics (TOG) (Vol. 26, No. 3, p. 87). ACM.

[18] Li, Y., Fan, X., Mitra, N.J., Chamovitz, D., Cohen-Or, D. and Chen, B., 2013. Analyzing growing plants from 4D point cloud data. ACM Transactions on Graphics (TOG), 32(6), p.157.

[19] Livny, Y., Yan, F., Olson, M., Chen, B., Zhang, H. and El-Sana, J., 2010. Automatic reconstruction of tree skeletal structures from point clouds. ACM Transactions on Graphics (TOG), 29(6), p.151.

[20] Prusinkiewicz, P. and Lindenmayer, A., 1990. The algorithmic beauty of plants. New York: Springer Verlag.

[21] Yang, M., Huang, M.C. and Wu, E.H., 2011. Physically-based tree animation and leaf deformation using CUDA in real-time. In Transactions on edutainment VI (pp. 27-39).

[22] Zhao, Y. and Barbič, J., 2013. Interactive authoring of simulation-ready plants. ACM Transactions on Graphics (TOG), 32(4), p.84.

[23] Lu, S., Zhao, C. and Guo, X., 2009. Venation skeleton-based modeling plant leaf wilting. International Journal of Computer Games Technology, 2009, p.1.

[24] Kider, J.T., Raja, S. and Badler, N.I., 2011, April. Fruit senescence and decay simulation. In Computer Graphics Forum (Vol. 30, No. 2, pp. 257-266).

[25] Jeong, S., Park, S.H. and Kim, C.H., 2013, Simulation of morphology changes in drying leaves. In Computer Graphics Forum (Vol. 32, No. 1, pp. 204-215).

[26] You, L.H., Yang, X.S., Pachulski, M. and Zhang, J.J., 2007, September. Boundary constrained swept surfaces for modelling and animation. In Computer Graphics Forum (Vol. 26, No. 3, pp. 313-322).

[27] Chaudhry, E., Bian, S.J., Ugail, H., Jin, X., You, L.H., Zhang, J.J. 2015. Dynamic skin deformation using finite difference solutions for character animation. Computers & Graphics (Vol. 46, pp. 294–305).

[28] Botsch, M., Sorkine, O.: On linear variational surface deformation methods. IEEE transactions on visualization and computer graphics 14(1), 213-230 (2008).

[29] Terzopoulos, D., Platt, J., Barr, A., Fleischer, K.: Elastically deformable models. In: ACM Siggraph Computer Graphics, vol. 21, pp. 205-214. ACM (1987).

[30] Do Carmo, M.P., Fischer, G., Pinkall, U., Reckziegel, H.: Differential geometry. In: Mathematical Models, pp. 155-180. Springer (2017).

[31] Celniker, G., Gossard, D.: Deformable curve and surface finite-elements for free-form shape design. ACM SIGGRAPH computer graphics 25(4), 257-266 (1991).

[32] Welch, W., Witkin, A.: Variational surface modeling. In: ACM SIGGRAPH computer graphics, vol. 26, pp. 157-166. ACM (1992).