Surface-based Growth Simulation for Opening Flowers

Takashi Ijiri[1], Mihoshi Yokoo[1], Saneyuki Kawabata[1], Takeo Igarashi[1,2]

[1]The University of Tokyo, [2]JST ERATO

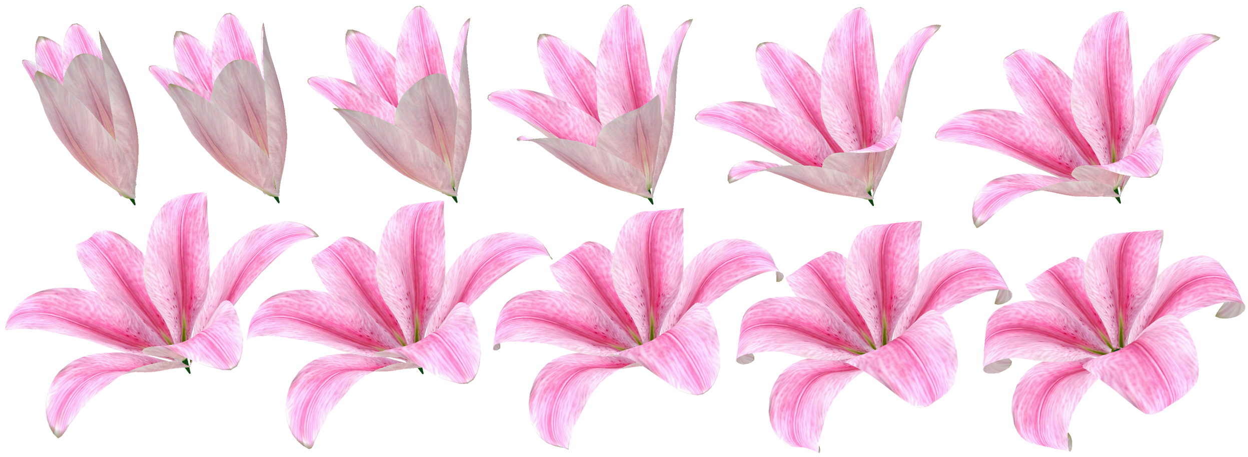

Figure 1. Representative keyframes of the unfolding animation created using our system for an Asiatic lily. The figure shows 1st, 10th, 20th, 30th, 40th, 50th, 60th, 70th, 80th, 90th, and 100th keyframe from top left to bottom right.

Abstract

We propose a biologically motivated method for creating animations of opening flowers. We simulate the development of petals based on the observation that flower opening is mainly caused by cell expansion. We use an elastic triangular mesh to represent a petal and emulate its growth by developing each triangular region. Our simulation process consists of two steps. The system first grows each triangle independently according to user-specified parameters and derives target rest edge lengths and dihedral angles. The system then updates the global shape to satisfy the rest lengths and dihedral angles as much as possible by means of energy minimization. We repeat these two processes to obtain keyframes of the flower opening animation. Our system can generate an animation in about 11.5 minutes. Applications include the creation of graphics animations, designing 3D plant models, and simulation for aiding biological study. In contrast to existing systems that simulate the development of flattened 2D petals, our system simulates the growth of petals as 3D surfaces. We show the feasibility of our method by creating animations of Asiatic lily and Eustoma grandiflorum.

Keywords: Flower opening animation, plant growth simulation.

Index Terms: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling; I.6.8 [Simulation and Modeling]: Types of simulation-Animation.

1 Introduction

Time-lapse photographs of flowers reveal impressive motions of petal unfolding: warping backward, popping open, and opening outward from a spiral form. Due to their visual appeal, animations of opening flowers are also used in computer graphics. However, it is still difficult to create convincing animations because designers must assign numerous keyframes based on expert knowledge of flowers as well as 3D modeling and animation. It is also necessary to deal with many collisions between petals.

Flower opening is also interesting from the physiological viewpoint, and there have been many reports in the biology literature regarding these phenomena [10]. Flower opening is generally induced by cell expansion that is characterized by its high rate and specific regulation. Petal motion during opening is caused by differences in growth rate between the inner and outer sides. For example, in the case of tulips, the optimum temperature for elongation of the outer side is about 10°C lower than that of the inner side, resulting in opening in the morning and closing in the evening [10]. In some species, movement of a midrib (petal axis) is important. For example, in the case of the Asiatic lily [3], the midrib undergoes strong warping and pulls the lamina of the petal in the early stage of opening. This triggers unfolding of the lily.

In this paper, we present a method to create animations of flower opening (Figure 1) by biologically motivated growth simulation; the opening of an actual flower is caused by differences in the cell expansion rate between the inner and outer sides or regional differences in the cell expansion rate. Our system represents a petal as an elastic triangular mesh and simulates its developing motion by repeating the following two processes. We first grow each triangle according to user-specified growth parameters and obtain the desired rest edge lengths and dihedral angles. The growth parameters consist of elongation growth, width growth, and growth direction for each side of a surface, and are specified by textures, which we call *growth maps* (Figure 2). We then optimize the global shape so that it satisfies the rest lengths and dihedral angles as much as possible. This process is formulated as a nonlinear energy minimization problem. Figure 1 shows an example of an Asiatic lily opening animation using the growth maps shown in Figure 2. It took about 11.5 minutes to generate 100 keyframes of this animation. We achieved this

Sci. bldg. 7, 302, 7-3-1, Hongo, Bunkyo-ku, Tokyo, Japan.
Ijiripui.is.s.u-tokyo.ac.jp

performance using implicit Euler methods introduced in cloth simulation [2].

Although simulation of petal surface development by expanding local regions has already been studied experimentally in biology [7, 28], these experiments were limited to flattened 2D shapes. Some computer graphics researchers introduced modeling systems for organic surfaces in 3D by using physically based simulation [6, 17]. They expand and bend user-specified local area of an elastic surface and relax accumulated energies. However, these methods deal with only a single surface and have not been applied for flower opening.

Our system simulates the opening of a flower based on biological observations. The use of growth maps is important for the biological perspective, surface with many folds could potentially be modeled more easily, and the physically-based approach provides a potentially better framework for handling collisions. We demonstrate the effectiveness of our technique with animations of the opening of Asiatic lily and Eustoma grandiflorum. We design the growth parameters for the lily opening animation based on a previous study [3] and on practical observations for Eustoma grandiflorum opening. Applications of our system include simulation for aiding biological research as well as the creation of computer graphics animations. Resulting animations are also useful for 3D modeling of plants. We introduce a simple plant modeling system which allows the user to easily design plants with flowers in different matured stages.

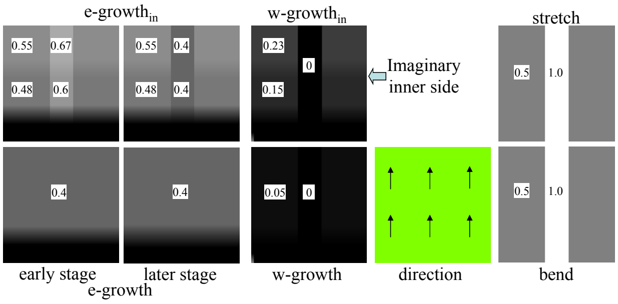

Figure 2. Growth maps for lily opening.

2 Background and Related Work

Plant modeling poses important challenges in computer graphics because of the intricate structure and unique free-form components, such as leaves, petals, and stems. The L-System, the best known tool for plant modeling [26], was first formulated by Lindenmayer [16] and introduced to the computer graphics community by Prusinkiewicz and Lindenmayer [23]. The L-System represents plant development by initial structure and rewriting rules. Prusinkiewicz et al. [27] also used the positional information to control parameters along a plant's axis. Boudon et al. proposed a *bonsai* tree modeling system [4] that provides intuitive control of the L-System using a decomposition graph. Deussen and Lintermann developed the Xfrog System [8, 9], which combines the power of a rule-based approach and intuitive user interfaces using graphical representation. Users can design a graph representing the branching structures of a plant with 11 node types. Furthermore, some researchers have developed sketch-based plant modeling systems [1, 14, 15, 20] to make plant design more accessible.

Simulation-based animation of plant development is also an interesting topic. Prusinkiewicz et al. [24] introduced dL-System by combining differential equations with the L-System. While the standard L-System describes plant growth discretely, dL-System provides a model that simulates plant growth in continuous time. The L-System has also been extended to simulate responses to a wide variety of interactions between plants and their environments [18, 25]. Hart et al. used a physical model to balance the weights of tree branches [11]. Streit et al. [30] used a feedback control system to emulate plant responses to stimuli and created naturally varied skeletal-based shapes by introducing imperfections in sensing stimuli. These studies, however, mainly focused on branching structure or skeletal shape, and did not deal with the growth of surface tissues of plants, such as petals or leaves.

Recently, some studies published methods dealing with plant surfaces. Runions et al. presented an algorithm for generating leaf venation patterns based on the canalization hypothesis [29]. They simulated modification of hormone source distribution and vein development toward the hormone during growth. Wang et al. simulated the growth of a *Xanthium* leaf in 2D [31]. They considered leaf tissues to be viscous and simulated growth by the expansion of incompressible fluids. Some research highly relevant to our work was reported in biology [7, 28] and computer graphics [5, 6, 17]. Coen et al. used a mass-spring model to represent flattened 2D petals of *Antirrhinum* and simulated their growth by changing the rest length of the springs [7, 28]. Combaz and Neyret introduced semi-interactive 3D modeling systems for wrinkles of fabrics [5] or organic shapes such as leaves [6]. Their system deforms an elastic surface by locally expanding the user-specified area and relaxing accumulated energies. Matthews applied a similar method for simulating the growth of plant surface tissues [17].

2.1 Representation of leaf and petal growth

The growth of petals can be described by growth characteristics of local regions [7, 13, 28, 29]. Coen et al. specified the growth of infinitesimal regions by three parameters: growth rate, anisotropy, and direction [7, 28] (Figure 3). (i) Growth rate is the rate at which a region changes in size. (ii) Anisotropy is the degree to which growth occurs preferentially in a certain direction. This can be represented by growth rate along the orthogonal axis. (3) Direction is the angle in which the principal direction of growth is oriented. These parameters may vary depending on both space (region) and time. Here, we used this biological growth parameter representation to control as many growth features as possible.

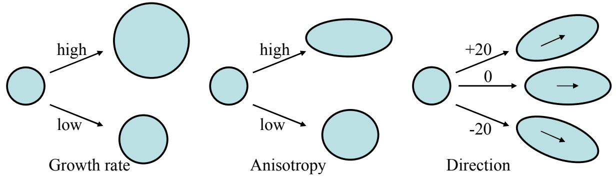

Figure 3. Three types of regional growth characteristics [28].

2.2 Physical model of petal tissue in simulation

The individual growth of local regions may conflict. In such cases, local regions must adjust their shapes with some compromises. A way to simulate this effect is to assign an elastic characteristic to the components of structure. Two physical representations have been proposed to simulate the tissue of petals or leaves: a mass-springs model presented in biology [7, 23, 28] and a fluids model presented in comuter graphics [31]. We chose the mass-spring model used by biologists because plant cells are normally prevented from sliding past each other by cell walls during growth, and the mass-spring representation maintains adjacent nodes. In contrast, the fluids representation does not, and it ignores the effect of cell walls.

It may be possible to introduce more physically accurate models to simulate a petal tissue. However, it is difficult to compute large models and large deformations, such as flower petals, in practical time using physically accurate models with currently available CPUs. In this paper, we used the mass-spring model to obtain animations within a reasonably short time.

Figure 4 shows a simple example of growth simulation using a spring model in 2D. (a) is the initial shape in which all springs have their rest lengths. The growth of regions in small time steps is emulated by changing their rest lengths as shown in (b). The system then modifies the global shape to relax the stretching forces of springs as shown in (c).

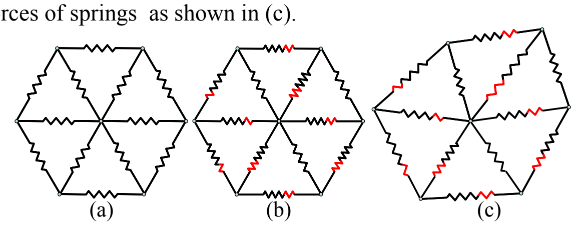

Figure 4. Growth simulation using a spring model [7].

3 Simulation Framework

We assume that the opening of a petal is driven by the growth of local regions. The local regions change their shapes due to the regulated growth rates, and the global shape is obtained as the accumulation of these local changes. We use an elastic triangular surface to simulate this behavior. To simulate 3D deformation, we take dihedral angles between neighboring faces into account (Figure 5(a)) [6].

The inputs to our system are a 3D bud model and its growth parameters. The user first loads isotropic triangular meshes designed using commercial software (Figure 5(b)). Each mesh has uv-coordinates for mapping the growth maps. The user then specifies all necessary parameters for each petal and arranges them onto the base of the flower, which we call the floral receptacle (Figure 5(c)). Our system uses a floral diagram editor [14] to support this arrangement process.

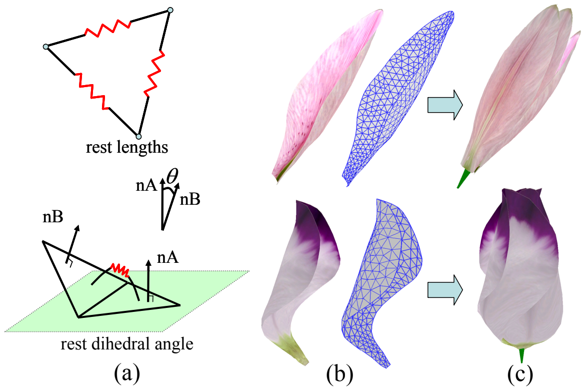

Figure 5. Rest length and angle, and initial bud models.

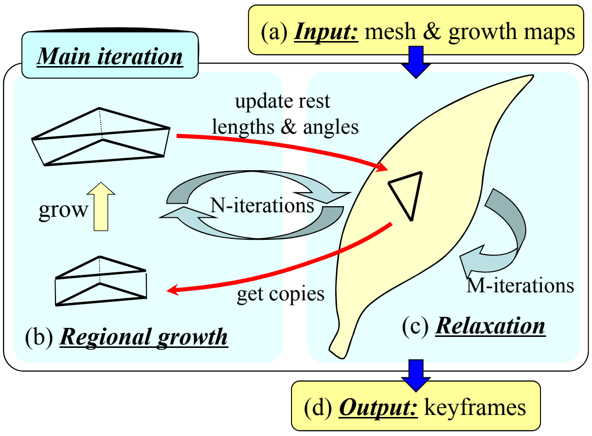

Figure 6. Overview of the simulation process.

Figure 6 shows the iteration process of the simulation. The edge lengths and dihedral angles of the initial petal mesh are used as the initial rest lengths and rest angles (a). The system emulates regional growth in a small time step by growing each triangle individually and modifying the rest lengths and angles (b). Then, the system modifies the global shape to make the resulting edge lengths and angles as close to the given rest lengths and angles as possible (c). We use static rather than dynamic simulation for the animation because flower opening is a very slow process. That is, we ignore velocity and compute each keyframe as a static shape that minimizes static energy. This minimization is formulated as a nonlinear problem and requires iterative computation (about M = 100 iterations in our case). Finally, the system outputs the resulting shape as a keyframe of animation (d). We then return to stage (b) to grow the petal again. We repeat this iteration of stages (b) and (c) about N = 100 times and generate a sequential animation by linearly interpolating the keyframes. Depending on the species, an increase in the angle between the petal and the flower axis is also important for opening (Figure 7). Our system supports this process by allowing the user to specify the increase in the angle of the joint at each iteration step.

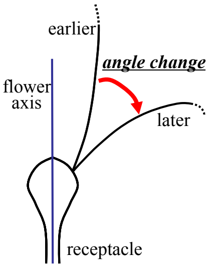

Figure 7. Changes in joint angle.

3.1 Growth parameters

Our system specifies regional growth by the following five parameters: direction, elongation growth (e-growth), width growth (w-growth), e-growth of inner surface (e-growth$_{in}$), and w-growth of inner surface (w-growth$_{in}$) similar to [28]. Instead of using the single anisotropy value approach, we use e-growth and w-growth values to represent anisotropic behavior. E-growth represents growth rate in the direction of growth, while w-growth represents the growth rate in the orthogonal direction (Figure 8(a)). E-growth$_{in}$ and w-growth$_{in}$ are used to control bending of the surface. Figure 8(b) shows the side view of a petal's cross-section. If the growth rate of the inner surface is higher than that of the base surface, it causes outward bending. Note that the inner surface is only used in regional growth computation for calculating target dihedral angles and not in global shape computation. We assume that petal thickness is approximately constant during opening.

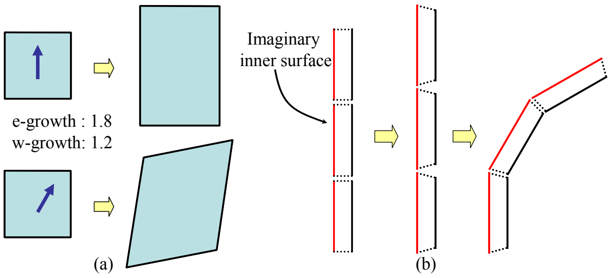

Figure 8. Growth representation for local regions.

Growth parameters vary depending on space and time. We specify regionally varying parameters by textures, which we call growth maps (Figure 2). Growth maps for e-growth and w-growth define the rate by brightness. The growth map for direction specifies the direction in a uv-coordinate system using red and green channels (red is u and green is v). The system also supports the use of multiple growth maps varying in time. The user defines a set of key growth maps at different points in time and the system interpolates them when computing regional growth at each iteration step.

4 Regional Growth Simulation

The goal of this process is to update the target rest edge length and target rest dihedral angle that guides the deformation of the global mesh geometry. At each growth simulation step, the system temporarily copies a triangle element of the mesh and grows the copy (reference triangle, $\boldsymbol{T}$) according to the growth parameters (e-growth and w-growth). We use the length of each elongated edge to update the rest length of the corresponding target edge. The system also creates an inner triangle ($\boldsymbol{T}_{in}$) on top of the reference triangle $\boldsymbol{T}$ to account for the current rest angles, and grows it according to the growth parameters for the inner surface (e-growth$_{in}$ and w-growth$_{in}$). The slanting angle caused by the difference between the reference triangle and the inner reference triangle is used as a new rest angle. Only the updated rest edge length and rest dihedral angle are returned to the main simulation cycle, while the reference triangle and inner reference triangle are discarded after each regional growth.

4.1 Computing growth parameters for each element

We refer to the vertex positions of $\boldsymbol{T}$ as $\boldsymbol{x}_0$, $\boldsymbol{x}_1$, and $\boldsymbol{x}_2$, and the uv-coordinates as $\boldsymbol{uv}_0$, $\boldsymbol{uv}_1$, and $\boldsymbol{uv}_2$. The growth parameters are given by the user-specified growth maps and a scalar parameter. The system defines the growth map value for $\boldsymbol{T}$ as a weighted average of growth map values at $\boldsymbol{uv}_0$, $\boldsymbol{uv}_1$, $\boldsymbol{uv}_2$ and $\boldsymbol{uv}_c$ = ($\boldsymbol{uv}_0$ + $\boldsymbol{uv}_1$ + $\boldsymbol{uv}_2$)/3;

$$\mathrm{map}(\boldsymbol{T}) = \frac{\mathrm{map}(\boldsymbol{uv}_0) + \mathrm{map}(\boldsymbol{uv}_1) + \mathrm{map}(\boldsymbol{uv}_2) + 3\,\mathrm{map}(\boldsymbol{uv}_c)}{6}, \quad (1)$$

where map($\boldsymbol{uv}$) is a color of a growth map at $\boldsymbol{uv}$.

The growth parameters for $\boldsymbol{T}$ are defined by four growth maps (e-growth, w-growth, e-growth$_{in}$, and w-growth$_{in}$) and one scalar value, dg, which defines the maximum growth rate. We define the growth rate of $\boldsymbol{T}$ by scaling the texture value (brightness) by dg:

$$\textit{growth rate} = 1 + dg \times \mathrm{map}(\boldsymbol{T}). \quad (2)$$

The direction map specifies the growth direction $\boldsymbol{d}$ in the uv-coordinate system in which elongation growth is oriented. The red and green channels define values in the u-direction and the v-direction, respectively. We convert the 2D vector $\boldsymbol{d}$ to a 3D vector on the plane of triangle $\boldsymbol{T}$, obtaining growth direction in the world coordinate $\boldsymbol{D}$.

4.2 Updating the rest length and rest angle

Given the e-growth rate, w-growth rate, those of the inner surface, and direction $\boldsymbol{D}$ for each reference triangle $\boldsymbol{T}$, the system expands $\boldsymbol{T}$, updating the rest lengths and angles. The system first scales $\boldsymbol{T}$ by the e-growth rate in direction $\boldsymbol{D}$ and by the w-growth rate in the orthogonal direction to obtain new vertices {$\boldsymbol{x'}_0$, $\boldsymbol{x'}_1$, $\boldsymbol{x'}_2$} (Figure 9). The barycenter of $\boldsymbol{T}$ is fixed. The system updates the rest length of the corresponding edge by the length of each grown edge as:

$$\textit{new\_rest\_length} = \| \boldsymbol{x'}_i - \boldsymbol{x'}_j \| \qquad (i, j = 0, 1, 2;\ i \neq j). \quad (3)$$

If an edge is not on the boundary of a petal surface, its two adjacent triangles give new individual rest lengths. We simply use the mean of the two.

The change in rest dihedral angles is caused by the difference in growth rate between the inner and outer surfaces. The system simulates this by using an imaginary inner reference triangle ($\boldsymbol{T}_{in}$). We begin by setting up $\boldsymbol{T}_{in}$ considering the current rest angle (θ_0) and *thickness* of the petal (Figure 10(a)). We perpendicularly translate each edge of $\boldsymbol{T}$ by offset s, which is obtained using the current rest angle; s = *thickness* * tan(θ_0/2). We calculate the intersections of the translated edges and lift them upward by *thickness* to obtain $\boldsymbol{T}_{in}$ ($\boldsymbol{x}^{\mathrm{in}}_0$, $\boldsymbol{x}^{\mathrm{in}}_1$, $\boldsymbol{x}^{\mathrm{in}}_2$). We then grow $\boldsymbol{T}_{in}$ using $\boldsymbol{D}$**,** e-growth$_{in}$, and w-growth$_{in}$ to obtain an expanded inner triangle $\boldsymbol{T'}_{in}$ ($\boldsymbol{x}^{\mathrm{in}}_0$', $\boldsymbol{x}^{\mathrm{in}}_1$', $\boldsymbol{x}^{\mathrm{in}}_2$') as in the expansion of $\boldsymbol{T}$. Finally, the system obtains the rest angle as:

$$\textit{new\_rest\_angle} = \tan^{-1}(s'/\textit{thickness}), \quad (4)$$

where s' is the distance between edge ($\boldsymbol{x'}_i$, $\boldsymbol{x'}_j$) and edge ($\boldsymbol{x}^{\mathrm{in}}_i$', $\boldsymbol{x}^{\mathrm{in}}_j$') projected onto the base surface (Figure 10(b)). These two edges may not be parallel, so we approximate s by the distance between the edge ($\boldsymbol{x'}_i$, $\boldsymbol{x'}_j$) and the midpoint ($\boldsymbol{x}^{\mathrm{in}}_i$'+ $\boldsymbol{x}^{\mathrm{in}}_j$')/2. Note that two adjacent triangles of each edge give new individual rest angles. We add the two and use the result as the new rest dihedral angle of the edge.

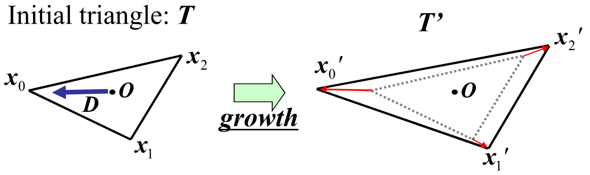

Figure 9. Expansion of a local triangle.

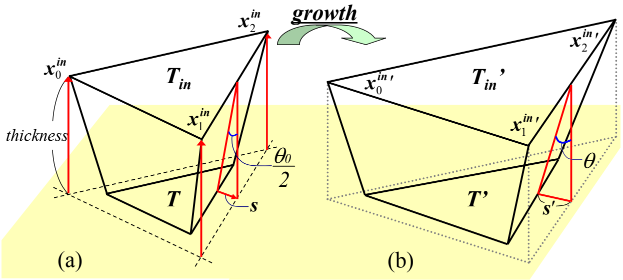

Figure 10. Change of a rest angle.

5 Updating Global Shape *via* Energy Minimization

The objective of this second process is to obtain a global petal pose that satisfies the rest lengths and rest angles resulting from the regional growth simulation as much as possible. We design an energy function that measures the deviation of edge lengths and dihedral angles and then seek to minimize it. Note that our goal here is a static pose of the petal surface, as opposed to dynamic animations. We implemented an implicit Euler method to ensure stability and achieve rapid convergence [2]. In this process, we also applied a special treatment to fix the bottom of the petal onto the receptacle and avoid intersections between petals. We use the following notation in this section. Given a triangular mesh of n particles, we use $\boldsymbol{x} \in \boldsymbol{R}^{3n}$ to refer to the positions of all particles, and $\boldsymbol{x}_i \in \boldsymbol{R}^3$ to the position of the ith particle.

5.1 Definition of the energy function

We define the gross energy function $\boldsymbol{E}(\boldsymbol{x})$ as a summation of local energies as follows:

$$\boldsymbol{E}(x) = \boldsymbol{E}_{stretch} + \boldsymbol{E}_{bend} \quad (5)$$

$$\boldsymbol{E}_{stretch}(x) = \sum_{(\boldsymbol{x}_i, \boldsymbol{x}_j) \in edges} 0.5 k_{sij} (\| \boldsymbol{x}_i - \boldsymbol{x}_j \| - L_{ij}^{0})^2 \quad (6)$$

$$\boldsymbol{E}_{bend}(x) = \sum_{(\boldsymbol{x}_i, \boldsymbol{x}_j) \in edges} 0.5 k_{bij} (\theta_{ij} - \theta_{ij}^{0})^2, \quad (7)$$

where L^0_{ij} is the rest edge length, θ is the dihedral angle, θ^0_{ij} is the rest dihedral angle of the edge ($\boldsymbol{x}_i$, $\boldsymbol{x}_j$), and k_s and k_b are weight constants. This error function is similar to the particle representation for cloth [2, 12].

The physical characteristics of petal tissue vary depending on the region. For example, in the Asiatic lily, the center of each petal is relatively hard because of a midrib. In Eustoma grandiflorum, the bottom regions are harder than the top regions. Our system allows assigning such regionally varying weights, k_s and k_b, by using stretch and bend textures (Figure 2). We define $k_{sij} = a*K_s*c_{uv}$, where a is the area of the adjacent triangle, K_s is the user-specified scalar weighting value, and c_{uv} is the averaged brightness of the stretch texture at $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$; k_b is defined similarly. It is currently difficult to measure these parameters for real flowers, so we estimate these parameters from observations.

The following iteration process requires the first and second derivatives of $\boldsymbol{E}(\boldsymbol{x})$, $\frac{\partial \boldsymbol{E}(\boldsymbol{x})}{\partial \boldsymbol{x}}$ and $\frac{\partial}{\partial \boldsymbol{x}}\frac{\partial \boldsymbol{E}(\boldsymbol{x})}{\partial \boldsymbol{x}}$, respectively. The first derivative $\frac{\partial \boldsymbol{E}(\boldsymbol{x})}{\partial \boldsymbol{x}}$ forms a vector in $\boldsymbol{R}^{3n}$ and the second derivative $\frac{\partial}{\partial \boldsymbol{x}}\frac{\partial \boldsymbol{E}(\boldsymbol{x})}{\partial \boldsymbol{x}}$ forms a sparse matrix in $\boldsymbol{R}^{3n\times 3n}$. These calculations are described in detail by Pritchard [22].

5.2 Implicit iteration

We minimize $\boldsymbol{E}(\boldsymbol{x})$ using the steepest descent, starting from $\boldsymbol{x}^i$, which is the solution of the previous iteration step:

$$\boldsymbol{x}^{i+1} = \boldsymbol{x}^{i} - h\frac{\partial \boldsymbol{E}(\boldsymbol{x}^{i})}{\partial \boldsymbol{x}}. \qquad (8)$$

To simplify the notation, we define $\Delta\boldsymbol{x} = \boldsymbol{x}^{i+1} - \boldsymbol{x}^{i}$.

$$\Delta\boldsymbol{x} = -h\frac{\partial \boldsymbol{E}(\boldsymbol{x}^{i})}{\partial \boldsymbol{x}}. \qquad (9)$$

Step size h must be very small to ensure stability when solving this explicit Euler method. Thus, we use the implicit Euler method to ensure stability with a relatively large step size:

$$\Delta\boldsymbol{x} = -h\frac{\partial \boldsymbol{E}(\boldsymbol{x}^{i}+\Delta\boldsymbol{x})}{\partial \boldsymbol{x}}. \qquad (10)$$

We make a first-order approximation by applying Taylor series expansion to $\partial \boldsymbol{E}(\boldsymbol{x}^{i}+\Delta\boldsymbol{x})/\partial\boldsymbol{x}$:

$$\frac{\partial \boldsymbol{E}(\boldsymbol{x}^{i}+\Delta\boldsymbol{x})}{\partial \boldsymbol{x}} = \frac{\partial \boldsymbol{E}(\boldsymbol{x}^{i})}{\partial \boldsymbol{x}} + \frac{\partial}{\partial \boldsymbol{x}}\frac{\partial \boldsymbol{E}(\boldsymbol{x}^{i})}{\partial \boldsymbol{x}}\Delta\boldsymbol{x}. \qquad (11)$$

We obtain a linear system by substituting this equation into equation (10):

$$\left(\frac{1}{h}\boldsymbol{I} + \frac{\partial}{\partial \boldsymbol{x}}\frac{\partial \boldsymbol{E}(\boldsymbol{x}^{i})}{\partial \boldsymbol{x}}\right)\Delta\boldsymbol{x} = -\frac{\partial \boldsymbol{E}(\boldsymbol{x}^{i})}{\partial \boldsymbol{x}}, \qquad (12)$$

where $\boldsymbol{I} \in \boldsymbol{R}^{3n\times 3n}$ is an identity matrix. We solve the equation using the conjugate gradient method and stop the iteration process when $\|\boldsymbol{E}(\boldsymbol{x}^{i+1}) - \boldsymbol{E}(\boldsymbol{x}^{i})\| < \delta$ or i > max iteration number. These thresholds are user-specified parameters; we used $\delta = 1.0*10^{-4}$ and max iteration number = 100, and $h = 0.1$.

5.3 Positional constraints and collision detection

Each petal is attached to the receptacle with the joint angle, and a change in this angle can be specified by the user. One way to keep the attachment is simply by fixing some particles on the petal surface during simulation. However, we found that such constraints cause instability and slow convergence. Therefore, we let the petal deform freely in the air without these positional constraints and then apply rigid transformation to the result to pull it back to the appropriate position on the receptacle. We use a shape-matching technique [19] to compute the desired rigid transformation (Figure 11). Beforehand, we define particles in the lower region of a petal as base particles (Figure 11(a) and (b)) and remember the initial petal shape. After each Euler iteration step, the system rotates the initial petal shape with the rotation angle specified by the user and computes a rigid transformation $\boldsymbol{R}$ that maps all base particles in the deformed petal to those of the initial petal (Figure 11(d)) as closely as possible. We then apply the rigid transformation $\boldsymbol{R}$ to all particles of the deformed petal (Figure 11(e)).

Flowers have highly specialized arrangements depending on the species [14]. In the case of the lily, petals are arranged inside of sepals (Figure 12(a)). In the case of Eustoma grandiflorum, the right side of each petal is beneath the adjacent petal (Figure 12(b)). We use this positional configuration of petals to handle collisions efficiently. To avoid exhaustive collision testing for all particles, we keep track of the nearest particle on the adjacent petals (*tracking point*) for each particle of the petal. After the rigid transformation process described above, we update the tracking points of the petal's particles. Given the tracking point, detecting a collision is straightforward. If a particle is on the wrong side of the adjacent petals, we push the particle back to the desired side with an offset (the current offset is 0.01) (Figure 12(c)).

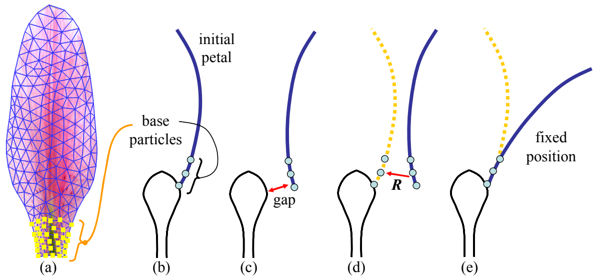

Figure 11. Fitting the bottom of a petal to the receptacle.

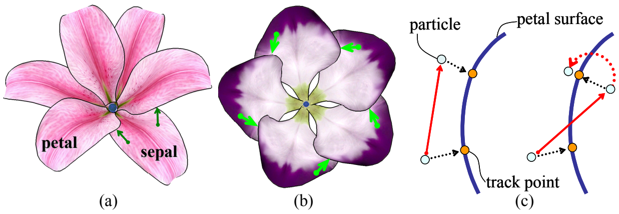

Figure 12. Handling collisions of petals.

6 Results and Discussion

We demonstrate the capability of our system by showing the opening animations of the Asiatic lily and Eustoma grandiflorum. We designed growth maps and other parameters by consulting the literature [3] and our own observations of real flowers. Also, we introduce a 3D plants modeling system which uses the resulting animations.

6.1 Asiatic lily

According to Bieleski et al. [3], the growth of petals or sepals of the Asiatic lily is controlled by the midribs. In the early stage of opening, the midrib reflexes strongly and pulls the lamina behind. As the flower matures, the midrib gradually stops warping. The lamina also expands during opening. There are differences in the rate between regions and this characterizes growth of the lamina, which is laterally flat and longitudinally warping. The joint angle increases by about 23° in the early stage, and does not change later. The opening behavior of a sepal is almost the same as that of a petal.

We designed growth maps (Figure 2) and other parameters (Table 1) using these observations. Direction is specified to fit the venation of the petal. We increase the joint angle by 23° from the 1st to the 60th iteration step (*i.e.*, by 23/60° for each iteration step). To specify expansion parameters, we divide the petal into three regions by the regional identity: midrib region, lamina region, and bottom region. The e-growth elongates the midrib and lamina regions equally. In the earlier stage (1st–40th iterations), e-growth$_{in}$ is larger than e-growth in the midrib region and slightly larger in the lamina region. This causes marked warping of the midrib and slight warping of the lamina. In the later stages (iterations 50–100), e-growth$_{in}$ in the midrib region is the same as e-growth, and e-growth$_{in}$ in the lamina region is slightly larger than e-growth. As a result, the midrib ceases warping. These maps are interpolated in the middle stage (iterations 40–50). There are slight increases in w-growth and w-growth$_{in}$ in the lamina region.

Some representative keyframes are shown in Figure 1. The warping motion of petals is captured with a high degree of realism. Figure 13 shows the 75th keyframe of a petal, indicating that our system can successfully mimic a slightly swollen petal shape caused by midrib warping. We think this swollen shape is caused by the large difference between e-growth and e-growth$_{in}$ at the midrib region.

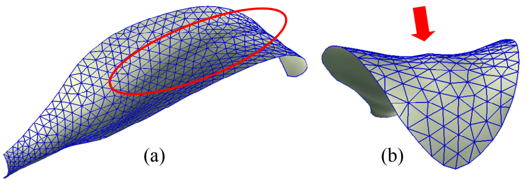

Figure 13. Swelled petal shape caused by midrib. The 75th keyframe is viewed from side (a) and top (b).

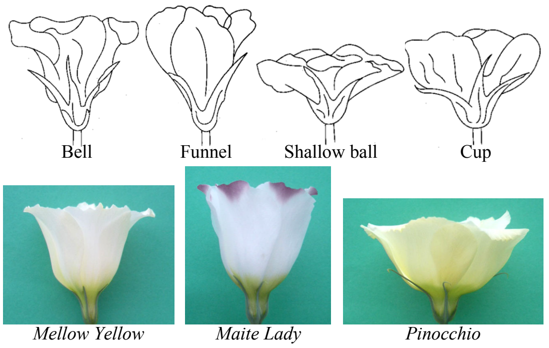

Figure 14. Illustrations of four shapes of *Eustoma* [21] and pictures of *Mellow Yellow*, *Maite Lady,* and *Pinocchio*.

Model	*dg*	*thickness*	*K_b*	*K_s*	*Average of edge length*
Lily	*0.003*	*0.2*	*20*	*20*	*1.3 (petal), 1.4 (sepal)*
Eustoma	*0.003*	*0.1*	*10*	*10*	*2.0*

Table 1. Parameters used for the simulation.

6.2 Eustoma grandiflorum

There are numerous cultivars of Eustoma grandiflorum, and they can be divided roughly into four groups with respect to their shape: bell, funnel, shallow ball, and cup [21] (Figure 14 top). We picked up three cultivars from different groups, *Mellow Yellow* from the bell group, *Maite Lady* from the funnel group, and *Pinocchio* from the shallow ball group (Figure 14 bottom), and observed their growth characteristics. We defined three stages of opening. At stage 1, the bud relaxes and the petals change color completely; at stage 2, buds open and the pistil matures; at stage 3, the flower matures. We divided the petal into three regions, the lower (proximal), middle, and upper (distal) regions, and observed cell lengths and widths in each stage, each region, and each side of the petal. The detailed data were reported previously [32]. The data on cells contained substantial noise and were difficult to use as it is. Therefore, it was used as a guide to manually design the growth parameters. Important features of the data are as follows:

(1) *Mellow Yellow*. In the later stage, the expansion rate of the inner/upper region was significantly larger than those of the outer/lower and outer/middle regions. This large growth rate of the inner/upper region is probably responsible for the reflexing petal shape. The growth rate was greater in the later stage than in the early stage. (2) *Maite Lady*. There were no significant differences in growth rate between regions. This relatively uniform growth rate is probably responsible for the flat petal shape. The growth rate was greater in the early stage than in the later stage. (3) *Pinocchio*. There were no significant differences in growth rate between regions, and this was probably responsible for the flat petal shape. Although the difference was not significant, the inner/lower and inner/middle regions showed greater elongation than the outer/lower and outer/middle regions.

We designed growth maps (Figure 16) and parameters (Table 1) for each cultivar taking these observations into consideration. The direction is specified to fit the venation; we observed a difference in venation patterns between *Maite Lady* and the other two (Figure 16). We increased the joint angles by 15° from the 1st to 50th iteration (*i.e.*, 15/50° for each iteration step). Simulations using different parameters were applied to the same bud model. Figure 17 shows the resulting animations. Starting from the same shape, each flower grew differently and adopted a final form similar to the shape of each corresponding cultivar.

6.3 Performance

It took about 11.5 minutes on a 2.4 GHz Intel Core 2 Duo CPU to obtain 100 keyframes for animation of the lily with three petals and three sepals. Each petal contained 388 particles, 685 triangles, and 1082 edges, while each sepal contained 323 particles, 567 triangles, and 889 edges. It also took about 8 minutes for the Eustoma grandiflorum with five petals. Each petal contained 261 particles, 458 triangles, and 718 edges. In the actual design process, we first created a growth map and applied it to a single petal. It took about 2.5 minutes for a single petal, which is fast enough for experiments with various parameters. After obtaining an acceptable result for a single petal, we computed the simulation of the whole flower opening.

6.4 Application for Plants Modeling

Once obtaining flower opening animations, they are useful for 3D plants modeling. For example, one can introduce opening flowers in a sketch-based flower modeling system based on a copy-and-paste interface introduced in [15]. In the system, the user can place a flower by drawing a stroke which specifies the flower axis. After the creation, the user can modify the age of a flower by a slider bar. The system then deforms the flower shape by using the specified age and the animation keyframes.. Figure 15 shows lily and Eustoma grandiflorum model designed in our system. These models contain flowers in different matured stages. We believe that our simulation system is applicable and useful for other existing plant modeling systems [9, 14, 23, 27].

7 Conclusions and Future Work

We have presented a method to create flower opening animations using biology-based growth simulations. We represent a petal as an elastic triangular mesh and emulate the regional growth from the expansion of each triangle. The rest lengths and dihedral angles of the mesh change gradually, driving the global deformation. We use static energy minimization to compute global geometry. Repetition of these processes yields keyframes of opening animation. The results indicated that our system can create highly realistic animations. The resulting animations are also useful for the 3D modeling of plants.The applications of our system include simulation for aiding biological research as well as systems for computer graphics purposes. The development of petals or leaves [10] is an important topic, and biologists have examined growth rates and the interplay between gene action and shape. Although the simulation model used in the current implementation is still experimental, the approach introduced here will be a powerful tool for building and examining various biological hypotheses in the near future.

One limitation of our work is that the user must specify many parameters (growth maps) manually to obtain reasonable simulation results. These growth maps are based on biological intuition, and we believe that such detailed control is required to allow biologists to run various types of experiment. However, easier control may be desirable for graphics applications. Therefore, we plan to provide a more compact, user-friendly set of parameters in the near future. Another future work is to develop a more accurate physical representation for the growth of plant tissues such as a viscoelastic model [17]. Although this paper focuses only on the growths of flowers, we also hope to create a growth animation of entire plant by combining our method with previous systems that can model entire plant structures such as L-System.

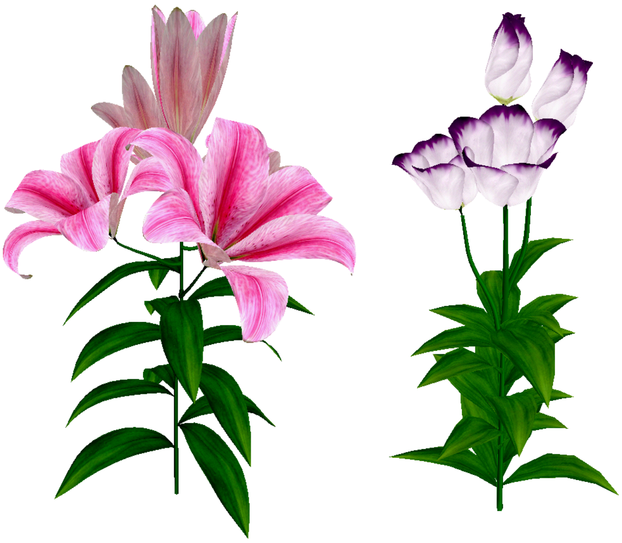

Figure 15. Lily and Eustoma grandiflorum model designed in our sketch-based system.

Acknowledgments

We thank the *Graphics Interface* reviewers for their insightful comments. This work was funded in part by Adobe Systems Inc. and JSPS Research Fellowship.

References

[1] Anastacio F., Sousa M. C., Samavati F., Jorge J.: Modeling plant structures using concept sketches. In *Proc. NPAR* 2006, 105-113.

[2] Baraff D., Witkin A.: Large steps in cloth simulation. In *Proc. ACM SIGGRAPH '98* (1998), 43-54.

[3] Bieleski R., Elgar J., Heyes J.: Mechanical aspects of rapid flower opening in Asiatic lily. *Annals of Botany*, 86 (2000), 1175-1183.

[4] Boudon F., Prusinkiewicz P., Federl P., Godin C., Karwowski R.: Interactive design of bonsai tree models. *Computer Graphics Forum*, 22, 3 (2003), 591-599. (*Proc. Eurographics '03*)

[5] Combaz J., and Neyret F.: Painting folds using expansion textures. In *Proc. Pacific Graphics*, 2002, 176–182.

[6] Combaz J., Neyret F.: Semi-interactive morphogenesis. In *Shape Modeling International*, 2006.

[7] Coen E., Rolland−Lagan A.-G., Matthews M., Bangham A., Prusinkiewicz P.: The genetics of geometry. In *Proc. the National Academy of Sciences*, 101, 14 (2004), 4728-4735.

[8] Deussen O., Lintermann B.: A modeling method and user interface for creating plants. In *Proc. Graphics Interface '97* (1997), 189-197.

[9] Deussen O., Lintermann B.: Interactive modeling of plants. *IEEE Computer Graphics and Applications*, 19, 1 (1999), 56-65.

[10] Doorn W. G., Meeteren U.: Flower opening and closure: a review. *Experimental Botany*, 54, 389 (Aug. 2003), 1801-1812.

[11] Hart J. C., Baker B., Michaelraj J.: Structural simulation of tree growth. *The Visual Computer*, 19(2-3) (2003), 151-163.

[12] House D. H., Breen D. E. Eds.: *Cloth Modeling and Animation*. A. K. Peters, 2000.

[13] Hejnowicz Z., Romberger J.: Growth tensor of plant organs. *Journal of Theoretical Biology*, 110 (1984), 93-114.

[14] Ijiri T., Owada S., Okabe M., Igarashi T.: Floral diagrams and inflorescences: interactive flower modeling using botanical structural constraints. *ACM Trans. Graph.,* 24, 3(2005), 720-726.

[15] Ijiri T., Owada S., Igarashi T.: Seamless integration of initial sketching and subsequent detail editing in flower modeling. *Computer Graphics Forum,* 25, 3 (2006), 617-624. (*Proc. Eurographics '06*)

[16] Lindenmayer A.: Mathematical models for cellular interactions in development, I & II. *Journal of Theoretical Biology* 18, 3 (1968), 280-315.

[17] Matthews M. J.: Physically based simulation of growing surfaces. *Masters Thesis, University of Calgary,* 2002.

[18] Měch R., Prusinkiewicz P.: Visual models of plants interacting with their environment. In *Proc. ACM SIGGRH '96* (1996), 397-410.

[19] Muller M., Heidelberger B., Teschner M., Gross M.: Meshless deformations based on shape matching. *ACM Trans. Graph.,* 24, 3 (2005), 471-478.

[20] Okabe M., Owada S., Igarashi T.: Interactive design of botanical trees using freehand sketches and example-based editing. *Computer Graphics Forum,* 24, 3 (2005), 487-496. (*Proc. Eurographics '05*)

[21] Okawa K.: *Torukogikyou (Eustoma grandiflorum).* Seibundo Shinkosha 2003 (In Japanese).

[22] Pritchard D.: Implementing Baraff & Witkin's Cloth Simulation. http://davidpritchard.org/freecloth/ docs/report.pdf, 2002.

[23] Prusinkiewicz P., Lindenmayer A.: *The Algorithmic Beauty of Plants.* Springer–Verlag, New York, 1990. With J. S. Hanan, F. D. Fracchia, D. R. Fowler, M. J. M. de Boer, and L. Mercer.

[24] Prusinkiewicz P., Hammel, M., Mjolsness E.: Animation of plant development. In *Proc. SIGGRAPH '93* (1993), 351-360.

[25] Prusinkiewicz P., James M., Měch R.: Synthetic topiary. In *Proc. ACM SIGGRAPH '94* (1994), 351-358.

[26] Prusinkiewicz P., Hammel M., Hanan J., Měch R.: L-Systems: from the theory to visual models of plants. In *Proc. the 2nd CSIRO Symposium on Computational Challenges in Life Sciences* (1996).

[27] Prusinkiewicz P., Mündermann L., Karwowski R., Lane B.: The use of positional information in the modeling of plants. In *Proc. ACM SIGGRAPH '01* (2001), 289-300.

[28] Rolland, A.-G., Bangham, J. A., Coen, E.: Growth dynamics underlying petal shape and asymmetry. *Nature*, 422 (2003), 161-163.

[29] Runions A., Fuhrer M., Lane B., Federl P., Rolland-Lagan A.-G., and Prusinkiewicz P.: Modeling and visualization of leaf venation patterns. *ACM Trans. Graph.*, 24, 3 (2005), 702-711.

[30] Streit L., Federl P., Sousa M. C.: Modelling plant variation through growth. *Computer Graphics Forum*, 24, 3 (2005), 487-596. (*Proc. Eurographics '05*)

[31] Wang I., Wan J., Baranoski, G.: Physically-based simulation of plant leaf growth. *Computer Animation and Virtual Worlds*, 15, 3-4 (2004), 237-244.

[32] Yokoo M., Kawabata S.: Quantitative evaluation of Corolla shape in Eustoma grandiflorum. *Jour. Japan. Soc. Hort. Sci. 75 (S2), 341* 2006.

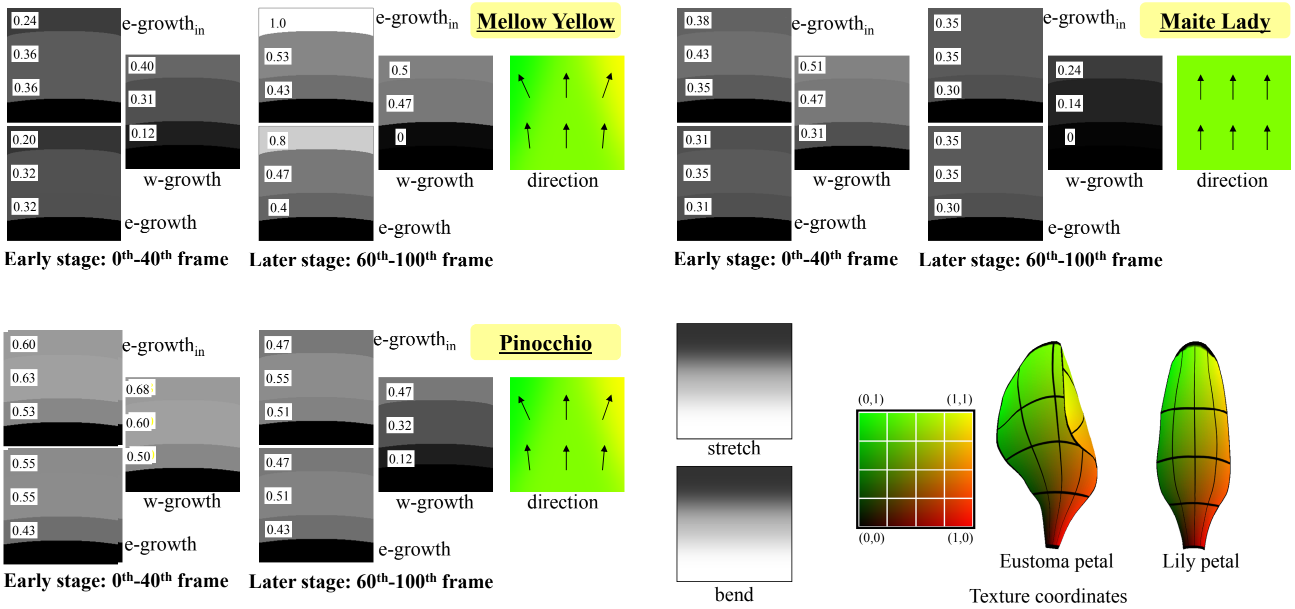

Figure 16. Growth maps for *Mellow Yellow* (top left), *Maite Lady* (top right), and *Pinocchio* (bottom left). We linearly interpolate two sets of growth maps for the early stage and later stage during the middle stage (iterations 40–60). We set the same stretch and bend texture for each cultivar. The figures on the bottom right indicate the uv-coordinates of Eustoma grandiflorum and Lily petals.

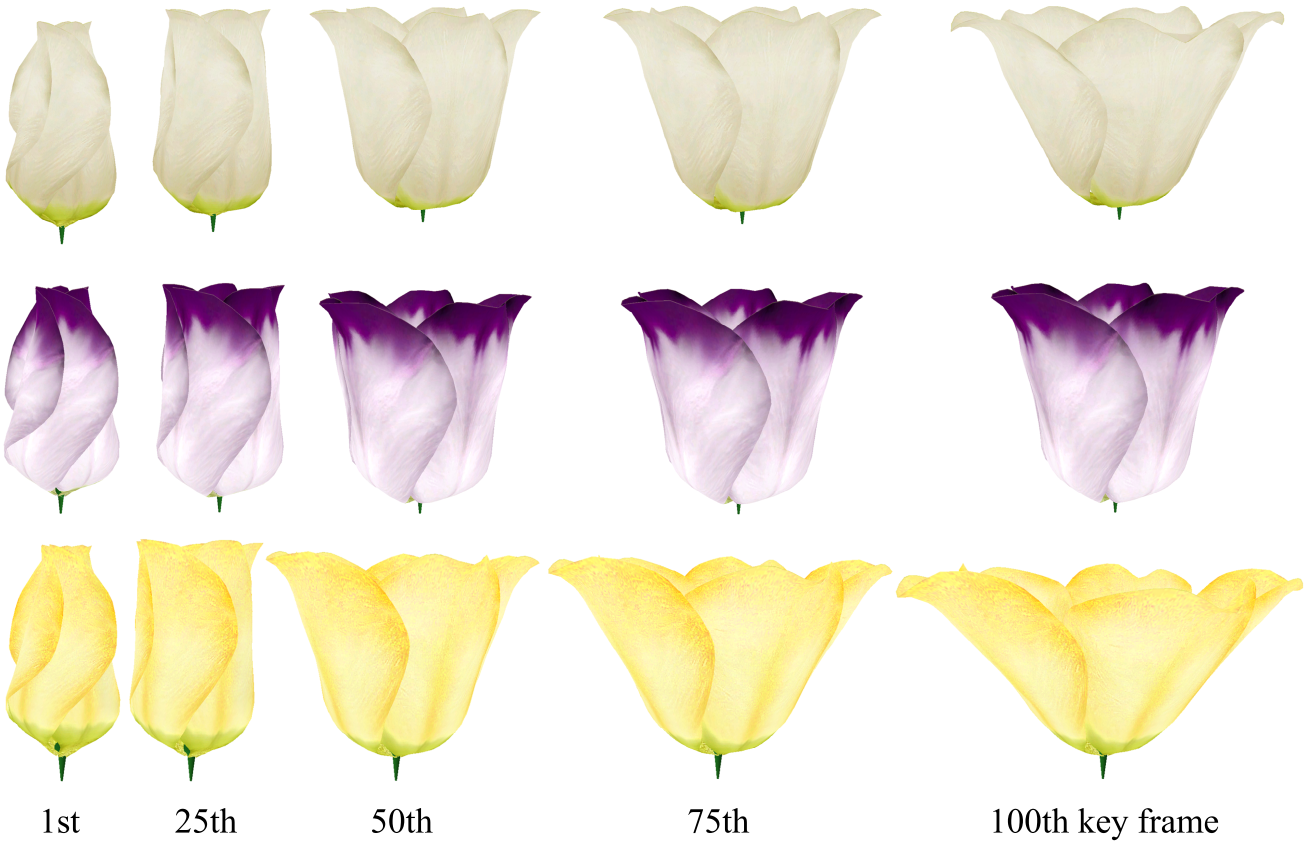

Figure 17. Representative keyframes of opening animations of Eustoma grandiflorum. Top row: *Mellow Yellow* (bell shape); middle row: *Maite Lady* (funnel); bottom row *Pinocchio* (shallow ball).