Artificial Pruning-Aware Procedural Modeling of Shrub Roses

Wataru Umezawa
Graduate School of Systems Design
Tokyo Metropolitan University
Tokyo, Japan
umezawa-wataru@ed.tmu.ac.jp

Tomohiko Mukai
Graduate School of Systems Design
Tokyo Metropolitan University
Tokyo, Japan
tmki@acm.org

***Abstract*—Roses fascinate people in many graphics contents such as movies and games. Although the beauty of roses is brought by the underlying botanical structure and skillful pruning, the conventional modeling method does not explicitly consider the tree shape change with artificial care. We propose a procedural method for modeling a branching structure of well-maintained shrub roses. The branch generation rules are derived to reproduce the characteristics of the ideal tree shape in the blooming season, which comes from both the species-specific growth model and the artificial pruning applied appropriately at various times throughout the year. Our system enables intuitive control to change the tree shape by tweaking several parameters. Those manual parameters are designed to represent the differences among rose varieties and mimic manual pruning that reflects the gardener's intentions. We will demonstrate the usability of our method through several experiments.**

***Index Terms*—procedural modeling, shrub rose, artificial pruning**

I. Introduction

Many plant models appear in computer graphics content such as movies and games. Recent animation productions frequently use procedural techniques, such as SpeedTree[1] and GrowFX[2], for generating a wide variety of plants semi-automatically. Several plant modeling methods represent a botanical structure as a set of mathematical expressions or structural rules [1]. Lindenmayer system (L-system) is the most famous mathematical model to produce the branching structure of wild plants [2]. The other procedural techniques, such as the space colonization method [3] and the self-organization tree model [4], generate life-like tree structures with simple editable rules. These conventional methods succeeded in generating diverse shapes of untended wild plants seen in grassland and forest.

We usually see well-tended plants in our daily life. For example, roses have fascinated people since ancient times. The beauty of rose comes from such as the colors, the fragrance, the shape of blooms, and the silhouette of trees. The shape appearance of roses varies depending on the varieties and can be designed with artificial care, as rose gardens are called gardeners' artwork. The pruning is crucial to designing the appealing silhouette and spread of the tree and leaves. There are typical procedures to maintain the rose [5] including the followings.

- Pruning trunks and stems for designing the tree silhouette before the high season of flower blooming
- Eliminating damaged stems to ensure healthy plants throughout the year
- Trimming redundant buds to promote the growth of large blooms during the blooming season
- Removing wilted blooms to stimulate sprouting and maintain a fresh appearance

Many gardeners employ the typical procedure incorporating unique ideas to grow their own roses. However, the conventional procedural techniques do not fully consider artificially maintained plants.

This paper proposes a procedural system for designing 3D models of well-maintained roses, especially shrub roses. Our insight is that artificial pruning must be considered in addition to plant species-specific growth rules for creating a convincing 3D shape model of decorative plants such as roses. Our method generates a tree structure of shrub rose and determines where leaves and blooms occur, as shown in Figure 1, where the bloom and leave shapes are manually created.

Fig. 1: Procedural generation of the branching structure of shrub rose. Our system generates the branching structure of shrub rose according to the species-specific structural rules and typical procedure of artificial pruning. Note that bloom and leaf shapes must be created by a designer.

[1]https://speedtree.com/
[2]https://exlevel.com/

Our approach is to derive growth rules that best approximate the ideal tree shape in the blooming season, which results from the underlying branching structure of shrub roses and the typical procedure of artificial pruning applied before the blooming season. For instance, the curved shape of each stem is computed according to the variety-specific growth model. The tree trunks stop growing when they reach an ellipsoidal boundary surface, which approximates the artificial pruning in the season before the flower blooming. The branching position and orientation of the stems, leaves, and blooms are determined based on several branching rules to enhance the plant's appearance. As shape differences between varieties can also be reproduced by tweaking several parameters, users can easily design various convincing rose models reflecting the actual cultivation process.

A drawback of our method is that it can generate only a skeletal structure of well-cultivated roses; untended wild roses and shapes of blooms, leaves, and thorns are not generated by our system. Moreover, our purely geometric algorithm does not guarantee a physically-valid result. Our system often produces interpenetration between stems and leaves and hardly reproduces the effect of gravity and wind. Despite these limitations, we believe that our artificial pruning-aware approach can be applied to model various decorative plants.

II. Related Work

L-system [2] is the most famous method to model the plant development process. The recursive nature of the L-system is represented by a formal grammar; the branching structure of tree-like plants is built by recursively applying the growth rules. Sketch-based methods [6] provides an intuitive user interface for controlling the L-system. An inverse procedural modeling technique [7] estimates an underlying L-system grammar of an input tree model or image.

Space colonization algorithm is the other well-known technique for procedurally generating a plant structure [3]. The plant silhouette and the distribution of tree nodes are specified by seed points representing the space available for growth. The tree skeleton is generated in an iterative manner using the seed points. In each iteration, short segments extend the skeleton toward nearby seed points. The users can easily control the plant shape by locating the seed points arbitrarily. This algorithm is later used in the self-organizing tree model [4] that well simulates the competition of buds and stems for available space and light resources while incorporating an underlying structural rule of plants. These conventional methods produce convincing geometrical tree structures of untended wild trees. Artificial pruning is only allowed after the synthesis using off-the-shelf shape editing tools, which requires designers to be knowledgeable about the gardening process.

Artificial pruning is considered in the synthetic topiary method [8]. This method generates an arbitrary plant silhouette by specifying the boundary shape of dense leaves. The plant silhouette can also be controlled by bounding the extent of stems [9]. A hierarchical, graph-based design interface was proposed for a bonsai model [10]. The dynamic response of the tree shape to the environmental change can be incorporated into a hand-craft shape [11]. Those methods aim to change the plants' silhouette like 3D sculpting. In contrast, our method allows specifying a boundary shape as a guideline to mimic an actual gardening process while considering the underlying tree structure.

III. Preliminary

This section summarizes terminology about the rose structure and a typical pruning procedure in actual cultivation.

A. Terminology of Shrub Rose

There are two types of rose: shrub roses and climbing roses. A shrub rose has multiple self-sustained trunks, and its silhouette is like a bush and tree. In contrast, a climbing rose is secured to a trellis or other structure because the stems are not thick and hard enough to support their weight. We focus only on the shrub roses in this paper, and the procedural modeling of climbing roses will remain for our future research.

The main stem growing from the bud union is called the trunk. We further classify the stems into two types reflecting the typical pruning procedure detailed in the following subsection. Hence the stems are classified into the following three types.

- **Trunk** grows upward from the bud union and has multiple child stems. It has no blooms or leaves.
- **Terminal stem** grows from a middle point on a parent stem and has child leaves on them, a few blooms at its tip and no child stems. A gardener does not prune terminal stems during the blooming season until the blooms wither.
- **Intermediate stem** grows from a parent stem. It has multiple child stems and no blooms or leaves. Its tip is artificially made by manual cut.

Trunks and intermediate stems work as a foundation of the tree shape. Blooms and leaves appear only from the terminal stems. Note that a leaf model consists of multiple leaflets along the petiole since the rose has compounded leaf.

B. Typical Pruning Procedure

This section briefly summarizes the typical procedure of artificial pruning in the actual cultivation of roses [5]. Several manual operations are necessary to stimulate plants with many flowers during the high-blooming season, each of which should be applied at different times in the year.

- **Pinching out trunks** The size and spread of the shrub rose is designed by pinching out the trunks before the first blooming season. This operation determines the base size of the plant because the pinched trunk stops growing. The pinch is also applied to control the number of stems since the plant forms new child stems around the pinched location.
- **Deadheading** Deadheading is a pruning operation to remove faded and spent flowers before they set seed, as shown in Figure 2. Deadheading aims to maintain

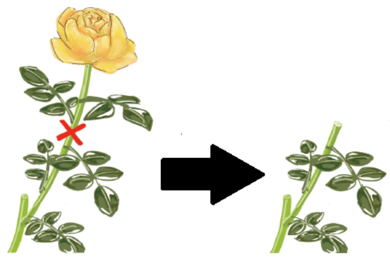


Fig. 2: Deadheading removes a bloom before they set seeds.

Fig. 3: New shoots grow around the point of deadheading

the plant's health and its fresh appearance. The other purpose is to stimulate the growth of new stems near the cut location, as shown in Figure 3. The deadheading is performed at any time over the blooming season.

- **Pruning of stems** Intermediate stems are pruned to maintain the plant shape. For example, gardeners cut out stems close to the surrounding stems. Damaged stems are pruned to maintain the health of the plant. Artificial pruning is also applied to control the location to appear a new shoot and bloom. When the tip of a parent stem is cut, the rose makes a new stem slightly below the cut location. The stem pruning is performed before the blooming season.
- **Shoot picking** Shoot picking is performed after pruning so that only selected shoots will grow. Crowded and downward-facing shoots are eliminated to maintain the plant's health and appearance.

Note that our method does not consider minor differences in manual pruning depending on region, climate, and individuals.

IV. Algorithm

Our method generates the branching structure of stems by determining the length and curvature of each stem, branching location, and branching angle between the parent and child stems. Our procedural system computes these geometrical structures to reflect the underlying botanical structure while allowing intuitive control with manual parameters representing the pruning policy and differences among rose varieties.

This section explains how to build the stem geometry (§IV-A, §IV-B), how to determine where the child branch grow (§IV-C, §IV-D, §IV-E), and how to compute where the leaves and blooms occur (§IV-F, §IV-G), according to the manual parameters. Note that our method determines the attachment points of blooms and leaves, assuming the user gives the shape model of blooms and leaves.

A. Stem Model

Each stem is composed of short segments that are connected with thicker nodes $\{\mathbf{p}_n | n \in \{1, \cdots, N_b\}\}$ where n denotes the node index, and $\mathbf{p}_n$ is the node position in the global coordinate system. The number of b-th stem's segments N_b

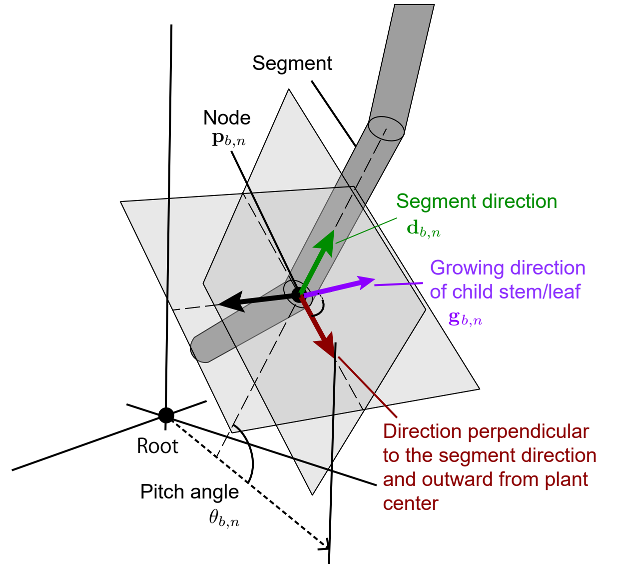


Fig. 4: Stem model composed of short segments connected by thicker nodes.

are determined according to the manual control (§IV-B). The segment length l is manually determined for each stem type. We denote the segment length of trunk, intermediate stem, and terminal stem by l_{trunk}, l_{inter}, and l_{term}, respectively.

The n-th node position $\mathbf{p}_{b,n}$ is recursively determined from the stem base to the tip by the following procedure.

1) The first node $\mathbf{p}_{b,1}$ locates at the stem base.
2) The $(n+1)$-th node position $\mathbf{p}_{b,n+1}$ is computed by $\mathbf{p}_{b,n} = l\mathbf{d}_{b,n} + \mathbf{p}_{b,n}$ where $\mathbf{d}_{b,n}$ is a unit vector representing segment direction.
3) These process is iterated as long as the number of segments is less than the upper limit and until the node position is within the boundary region.

The segment direction $\mathbf{d}_{b,n}$ is computed based on the underlying growth system of the shrub rose. Each stem aims upward as it approaches the tip. To emulate the growing behavior, our method increases the pitch angle $\theta_{b,n} = \arctan\left(d_{b,n,y}/\|\mathbf{d}^{\perp}_{b,n}\|\right)$ with respect to the horizontal plane as the segment index increases, where $\mathbf{d}^{\perp}$ represent a horizontal projection of a directional vector $\mathbf{d}$ as $\mathbf{d}^{\perp} = [d_x,\ 0,\ d_z]$. The segment direction $\mathbf{d}_{b,n+1}$ is computed based on $\mathbf{d}_{b,n}$ to gradually increase the pitch angle to the manually-specified target angle $\hat{\theta}$. The ideal direction $\mathbf{d}^*$ is determined so that the pitch angle becomes $\hat{\theta}$ while preserving its horizontal projection $\mathbf{d}^{\perp}_{b,n+1}$ parallel to that of the preceding segment $\mathbf{d}^{\perp}_{b,n}$ as

$$\mathbf{d}^* = \begin{pmatrix} d^*_x \\ d^*_y \\ d^*_z \end{pmatrix} = \begin{pmatrix} \frac{d_{b,n,x}\cos(\hat{\theta})}{\|\mathbf{d}_{b,n}\|} \\ \sin(\hat{\theta}) \\ \frac{d_{b,n,z}\cos(\hat{\theta})}{\|\mathbf{d}_{b,n}\|} \end{pmatrix}, \tag{1}$$

where $\|\mathbf{d}^*\| = 1$ holds. The direction of the subsequent

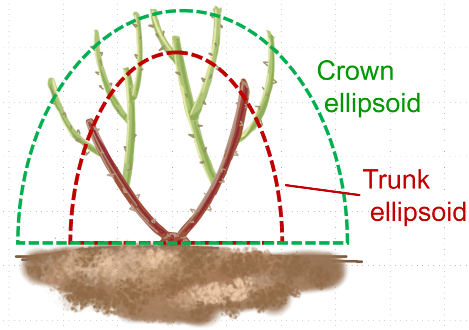


Fig. 5: Crown and trunk ellipsoids for bounding tree height and spread and mimicking pinch out of trunks, respectively.

segment $\mathbf{d}_{b,n+1}$ is computed by blending $\mathbf{d}_{b,n}$ and $\mathbf{d}^*$ to gradually blend the stem as

$$\mathbf{d}_{b,i+1} = \frac{\kappa \mathbf{d}_{b,i} + n^\sigma \mathbf{d}^*}{\kappa + n^\sigma}, \tag{2}$$

where κ denotes the linearity of the stem, σ is the bending coefficient. These parameters are each defined for intermediate stem κ_{inter}, σ_{inter} and terminal stem κ_{term}, σ_{term}, respectively. The bending term n^σ works to rapidly increase the pitch angle closer to the target value $\hat{\theta}$ as the number of segments increases. Once the pitch angle reaches $\hat{\theta}$, the subsequent nodes $\{\mathbf{d}_{b,i} | i > n\}$ keeps same direction $\mathbf{d}_{b,n}$ to grow the tip straightly.

The direction of the first segment $\mathbf{d}_1$ is determined depending on the stem type. Regarding a trunk, $\mathbf{d}_1$ is randomly determined so that its pitch angle is positive to grow upward from the bud union. The direction of the intermediate stem and terminal stem's first segment is determined to grow upward while satisfying the underlying botanical structure, as detailed in the following subsection.

B. Boundary Ellipsoids

The height and spread of the tree are controlled using two ellipsoids similar to [9]. The one ellipsoid, called *trunk ellipsoid*, represents the growth limit of the trunks that mimics pinching-out trunks. The other ellipsoid is used to bound the growth of stems, which mimics the pruning of stems. This ellipsoid is called *crown ellipsoid* as this type of pruning is to design the bounding shape (crown) of the tree.

The shape of each ellipsoid is determined with two manual parameters: the height from the ground h and the radius r of the horizontal cross-section. The crown ellipsoid is larger than the trunk ellipsoid in a typical gardening scenario as $h_{\text{crown}} \geq h_{\text{trunk}}$ and $r_{\text{crown}} \geq r_{\text{trunk}}$. Each trunk stops growing when its end segment reaches the boundary surface. The trunk ellipsoid indirectly controls the number of trunk segments. A stem grows through the trunk ellipsoid not to exceed the crown ellipsoid. The stems do not always reach the crown ellipsoid as there is another constraint to limit the number of stem segments N_b less than N_D. The maximum number N_D is manually specified to mimic the deadheading before the blooming season. Using a smaller crown ellipsoid than a trunk ellipsoid, both trunks and stems are bounded by the crown ellipsoid, while the trunk ellipsoid does not affect the tree shape.

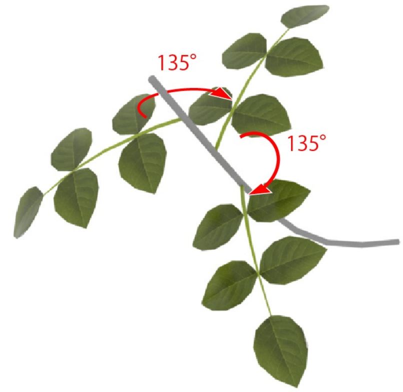


Fig. 6: $\frac{3}{8}$ phyllotaxis. Child stems and leaves execute a spiral with an angle of $3/8$ of a full rotation along with a parent stem.

C. Phyllotaxis

We here introduce *phyllotaxis* before explaining the branching rule in the next subsection. Child stems and leaves are arranged around a parent stem like a spiral staircase. Phyllotaxis is a general rule of the regular arrangement of child stems along with a parent stem [3]. Roses are known to follow the $3/8$ phyllotaxis, where the child stems execute a spiral with an angle of $3/8$ of a full rotation along with a parent stem, as shown in Figure 6. Our method employs the phyllotaxis to determine which direction child stems and leaves grow.

Let $\mathbf{d}_n$ and $\mathbf{p}_n$ be a direction, and a base position of n-th segment of a parent stem, and $\mathbf{g}_n$ be the first segment direction of a child stem at n-th parent segment. Our method imposes two constraints on the child direction $\mathbf{g}_n$. The first constraint enforces a child stem or leaf to grow vertically from the parent segment as $\mathbf{d}_n \cdot \mathbf{g}_n = 0$ where $\cdot$ denotes the inner product. The second constraint imposes an angle between the adjacent children to be $3\pi/4$ radian around the parent stem axis as $\forall n, \angle_{\mathbf{d}_n}(\mathbf{g}_n, \mathbf{g}_{n+1}) = 3\pi/4$ where $\angle_{\mathbf{d}_n}$ evaluates an angular difference between two vectors around the axis $\mathbf{d}_n$.

We also constrain the child's direction at the parent tip $\mathbf{g}_{N,1}$ to improve the plant shape appearance. Ideally, leaves should be arranged to face outward and easily seen by the viewer. Child stems should also be grown outward to make the plant silhouette large without crowding stems. Our method computes an ideal child direction $\mathbf{g}_N^*$ to ensure that ${\mathbf{g}_N^*}^\perp$ and $(\mathbf{p}_N - \mathbf{p}_{\text{root}})^\perp$ are parallel. This equation means that a child stem of the parent tip ideally grows away from the central vertical axis of the plant. The pitch angle of $\mathbf{g}_N^*$ is determined to make the angle between $\mathbf{d}_N$ and $\mathbf{g}_N^*$ a constant angle $\phi > 0$.

[3] https://mathworld.wolfram.com/Phyllotaxis.html

We use $\phi_{\text{stem}} = \pi/3$ for child stems and $\phi_{\text{leaf}} = \pi/2$ for child leaves in all experiments.

The ideal direction $\mathbf{g}_N^*$ is used as a baseline to calculate the actual growth direction as detailed in the following subsection. The child direction of the other nodes $\{\mathbf{g}_n | 1\{\leq n < N_b\}$ are determined so that the $3/8$ phyllotaxis and $\mathbf{g}_n \cdot \mathbf{d}_{b,n} = 0$ are satisfied.

D. Branching Rules

According to the underlying botanical structure, an actual plant creates child stems on a parent stem. Moreover, artificial pruning performed before the blooming season stimulates the emergence of child stems slightly below the cut point. The gardener then picks new shoots and leaves that face downward and limits the number of child stems per parent stem. We use different growth rules for the intermediate and terminal stem to approximate this cultivation process.

a) Terminal stem: Up to two terminal stems are generated on the intermediate stem pruned by the crown ellipsoid. Two child intermediate stems are created at $\mathbf{p}_{N_b}$ and $\mathbf{p}_{N_b-1}$. This branching rule mimics the pruning before the blooming season. All terminal stems have a manually-specified number of segments N_T because the terminal stem is not pruned during the blooming season.

b) Intermediate stem: An intermediate stem emerges from a trunk or another intermediate stem with the maximum number of segments N_D. In other words, stems whose number of constituent segments reaches the upper limit N_D generate two new child stems at the tip. Our system generates two intermediate stems from a parent stem at $\mathbf{p}_{N_b}$ and $\mathbf{p}_{N_b-1}$. This branching rule mimics the deadheading before the blooming season.

The direction of the child stem at the tip segment $\mathbf{g}_N$ is determined using the ideal direction $\mathbf{g}_N^*$. This direction is directly used as $\mathbf{g}_N$ without any modifications if the y-component $g_{N,y}$ is greater than zero, i.e., $\mathbf{g}_N^*$ aims upward. If the ideal direction aims downward, our method searches for the optimal $\mathbf{g}_N$ in which the y-component $g_{N,y}$ first becomes greater than zero by rotating $\mathbf{g}_N^*$ every 60 degrees around the segment axis $\mathbf{d}_N$.

Furthermore, these two types of stems are randomly generated from an unbranched segment of the intermediate stem and trunk, which approximates the growth of a few shoots during the blooming season. Such a stem is less likely to occur on older segments near the root, and some breeds rarely develop new shoots from the lower part of the plant. To reproduce the random nature, we use probabilistic selection and thresholding based on distance from the root.

Our algorithm first checks if the node position is distant from the root for each node of the intermediate stems and trunks. The lower boundary is defined by scaling the crown ellipsoid using shrinkage coefficient γ satisfying $0 \leq \gamma \leq 1$. The node $\mathbf{p}_n$ is confirmed to be outside the shrunken ellipsoid if satisfying $(p_{n,x}^2 + p_{n,z}^2)/{r_{\text{crown}}}^2 + p_{n,y}^2/{r_{\text{crown}}}^2 > \gamma^2$. The roulette selection algorithm is then used to determine whether the valid node has no branch, an intermediate stem, or a terminal stem. We use the occurrence probability of terminal stem ρ_{term} and intermediate stem ρ_{inter} for the roulette selection, where $0 \leq \rho_{\text{term}} + \rho_{\text{inter}} \leq 1$ holds.

The first segment direction of the randomly generated stems always satisfies the $3/8$ phyllotaxis and is perpendicular to the parent segment. If the direction aims downward, the new stem is removed, mimicking the artificial shoot picking. The generated intermediate stem also has child stems according to the branching rule.

E. Density and Appearance Control

Our stem model and branching rules do not take into account the maintenance of a certain separation distance among stems. In actual cultivation, dense branching causes damage to stems and blooms from collisions, deficiency of nutrients and sunlight, and poor visual appearance. Our method removes a new child stem after the generation if it is close to the other existing stem. A candidate stem b is removed if any one of its node locates within a certain distance from any other existing stem, expressed as

$$\exists n, \tilde{b}, \tilde{n}, \|\mathbf{p}_{b,n} - \mathbf{p}_{\tilde{b},\tilde{n}}\| < \delta_{\min} \ , \tag{3}$$

where $\delta_{\min}$ denotes the manually-specified distance threshold. This straightforward approach works well under the assumption that a skilled gardener can predictably remove new shoots that will eventually penetrate with other stems. The remaining problem is that our method does not check the collision with leaves and blooms. We could further improve the visual quality by using the other collision avoidance technique [12].

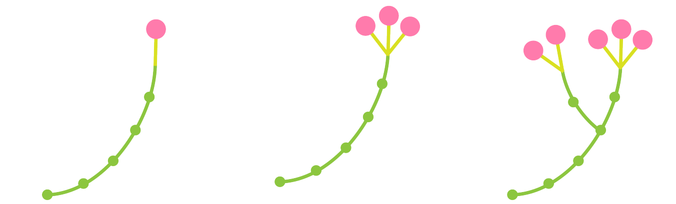
F. Leaves

Leaves grow only from the terminal stems, and the rose varieties determine the number of leaves per stem. Our method locates one attachment point of the petiole at each segment of the terminal stem according to the $3/8$ phyllotaxis. The ideal direction $\mathbf{g}_N^*$ is used as the direction of the petiole at the tip segment $\mathbf{g}_N$. Note that leaves are not removed even if they grow downward.

G. Peduncles and Blooms

Multiple bloom models are attached at the tip of each terminal stem. The end tip of the terminal stem branches into several peduncles, each of which supports a bloom. Our method generates at most three peduncles for each terminal stem to avoid a messy visual appearance.

The first peduncle is generated in the direction of the parent segment $\mathbf{d}_{N_T}$. The second and third peduncles are generated to make a $\pi/4$ angle with the first peduncle and to be perpendicular to the outward direction, as shown in Figure 7. The peduncle length is 1.5 times the terminal segment length l_{term}. The target pitch angle of each peduncles is defined by $\min(\hat{\theta}, \theta_{\text{droop}})$ where θ_{droop} denotes the drooping angle. This



(a) One bloom is attached via one peduncle.

(b) Three blooms are attached to one terminal stem via branched peduncles.

(c) Five blooms are attached using one additional stem.

Fig. 7: Attachment of bloom shape models to terminal stem.

definition is for approximating the droop of blooms caused by their weight.

An additional stem is generated at the two-node back from the tip of the terminal stem for attaching more than three blooms. A second additional stem is generated from one more node back. These additional stems have the same number of segments as the number of nodes to the tip of the terminal stem. For example, an additional stem has three segments if generated at $(N_T - 3)$-th node of the terminal stem. Consequently, each terminal stem can have up to nine blooms located at about the same height. The user manually specifies the number of blooms per terminal stem N_B.

V. Experimental Results

We manually created a shape model of bloom according to [13]. The leaf model was also created with five leaflets along a single petiole. The segment lengths of trunk and intermediate stem are fixed to $l_{\text{trunk}} = 7.0$ and $l_{\text{inter}} = 5.0$ for all experiments.

A. Experiment on 2D

We first created a branching structure in two dimensions without the density control and the attachment of blooms and leaves, as shown in Figure 8. The lighter-blue and darker-blue circles represent the trunk ellipsoid and crown ellipsoid, respectively. The trunks and intermediate stems were cut slightly outside these boundary ellipsoids, shown by black lines. Several intermediate stems stopped growth in the crown ellipsoid, which well approximated the result of the deadheading before the blooming season. The terminal stems, shown by green lines, grew exceeding the crown ellipsoid, and each terminal stem had one broom according to the parameter setting. No stems grew downward owing to the pitch angle control.

B. Individuality of Trunk Direction

The specified number of trunks grow from the bud union in different directions. This randomness corresponds to individual differences of the same rose variety. Figure 9 shows some examples of the individuals, which were generated by fixing all parameters to the specific values shown in the Table I. Differences in trunk direction significantly affected the final tree shape and branching structure, as these results show.

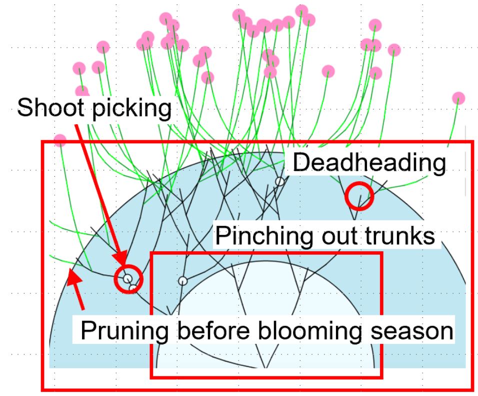


Fig. 8: Procedurally generated branching structure in 2D without density control and attachment of blooms and leaves.

TABLE I: Parameter setting for Figure 9 and Figure 12

Terminal stem

	N_B	N_T	θ_{droop}	l_{term}	κ_{term}	σ_{term}	γ
Fig 9	1	7	$\pi/2$	5.0	3.0	1.2	0.1
Fig 12(a)	1	7	$\pi/2$	5.0	5.0	1.2	0.5
Fig 12(b)	1	9	$\pi/2$	7.0	1.0	1.2	0
Fig 12(c)	3	7	$\pi/2$	5.0	5.0	1.2	0.1
Fig 12(d)	1	13	$\pi/3$	8.0	18.0	0.6	0.1

Intermediate stem

	N_D	h_{crown}	r_{crown}	κ_{inter}	σ_{inter}
Fig 9	6	70.0	70.0	10.0	1.0
Fig 12(a)	6	50.0	50.0	8.0	1.1
Fig 12(b)	7	70.0	60.0	1.0	1.2
Fig 12(c)	6	70.0	70.0	5.0	1.0
Fig 12(d)	6	30.0	30.0	11.0	1.0

Trunk and others

	N_S	h_{trunk}	r_{trunk}	$\hat{\theta}$	ρ_{inter}	ρ_{term}	$\delta_{\min}$
Fig 9	4	40.0	40.0	1.05	0.05	0.2	2.0
Fig 12(a)	2	40.0	40.0	1.1	0.05	0.1	2.0
Fig 12(b)	5	100.0	100.0	1.05	0.03	0.33	3.0
Fig 12(c)	5	55.0	55.0	1.05	0.15	0.25	2.0
Fig 12(d)	4	40.0	40.0	1.05	0.05	0.25	2.0

C. Control with Pitch Angle

Figures 10 shows the effects of pitch angle $\hat{\theta}$. Figures 10 (a), (b), and (c) were generated using $\hat{\theta} = 0.32$ rad, 1.05 rad, and $\pi/2$ rad, respectively, and the number of trunks was $N_S = 1$. The other parameter values were identical to those of Figure 9. The results show that the aspect ratio of the tree shape varied depending on $\hat{\theta}$ even when using the identical values for the other pruning parameters. The slight pitch angle resulted in the low and horizontally spreading tree shape because all stems grew in the lateral directions, as shown in Figure 10 (a). In contrast, a large pitch angle produced tall, narrow, and dense stem distribution, as shown in Figure 10(c). There were many L-shaped stems because every stem rapidly changed the pitch angle and saturated to the target angle.

D. Control with Trunk Ellipsoids

Here we show the differences produced by the size of the trunk ellipsoid. The radius and height $(r_{\text{trunk}}, h_{\text{trunk}})$ of the trunk ellipsoid were set to $(100, 100)$, $(60, 60)$ and $(20, 20)$

Fig. 9: Individual difference of the same variety produced by trunk directions. These results were generated by fixing the manual parameters as shown in Table I.

Fig. 10: Different rose varieties produced by changing pitch parameter $\hat{\theta}$. (a) Small pitch angle resulted in the low and widely spreading tree. (b) An average value produced an medium height and spread. (c) Large pitch angle produced a narrow and dense branches.

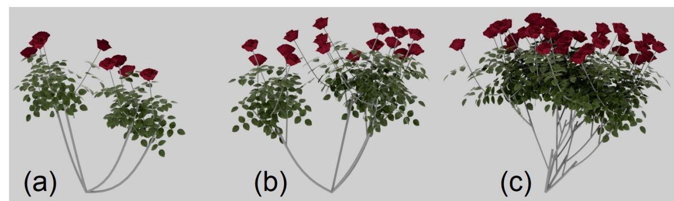


Fig. 11: Shape difference produced by the size of trunk ellipsoid. The radius and height of the ellipsoid decrease from (a) to (c).

TABLE II: Parameter settings for Figure 13

	δ_{min}	N_S	N_D	r_{trunk}	h_{trunk}	r_{crown}	h_{crown}
Left	4.5	4	6	20	80	30	150
Right	2.0	6	4	40	15	50	35

for creating Figure 11(a), (b), and (c), respectively. The randomized stem generation was disabled by setting $\rho_{\text{inter}} = 0$ and $\rho_{\text{term}} = 0$ for the sake of clarity of trunk structure. The other parameter values, including the size of the crown ellipsoid, were identical to that of Figure 9.

In Figure 11(a), the trunk was pruned by the crown ellipsoid that is smaller than the trunk ellipsoid. Terminal stems were therefore grown from the trunks without any intermediate stems. Figure 11(b) show the shortening of trunks with size reduction of the trunk ellipsoid. Moreover, the smallest trunk ellipsoid produced the shortest trunks and the largest number of stems and blooms as there was sufficient space available for many branches to occur between the trunk and crown ellipsoids, as shown in Figure 11(c).

E. Various Rose Varieties

Figure 12 shows four examples created using different parameter settings as shown in Table I. Different rose varieties are represented using different parameter settings. Figure 12(a) shows a small and sparse tree having a few blooms because this tree has fewer trunks and a smaller crown ellipsoid. In Figure 12(b), many terminal stems grew from the trunks due to the more significant occurrence probability ρ_{term} and the smaller crown ellipsoid than the trunk ellipsoid. The terminal stems have a long and straight shape owing to the larger N_T, l_{term}. Figure 12(c) shows dense tree produced by the larger N_B and N_S. Figure 12(d) shows the result of the small crown ellipsoid and the longest terminal stems. The blooms drooped significantly by the larger θ_{droop}.

F. Purpose-built Controls

Figure 13 shows two examples generated for different purposes, assuming that the two roses were of the same variety. The rose variety is expressed by setting $\hat{\theta} = \pi/2$, $\sigma_{\text{inter}} = 2$, $\kappa_{\text{term}} = 1$, $\sigma_{\text{inter}} = 2$, and $\rho_{\text{inter}} = 20$, and different pruning were applied depending on the different purpose. We made the left rose to have a tall and slim silhouette and fewer stems and leaves to maximize visual presence in tight spaces and a situation surrounded by other plants. In contrast, the right rose had a broader and shorter shape and more stems and leaves. This type of pruning keeps the plant small so as not to block other objects and enhance the garden's openness.

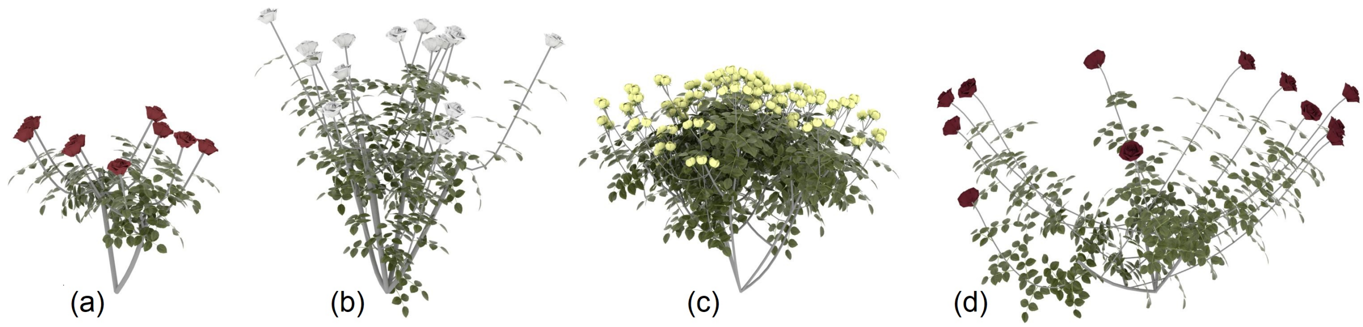


Fig. 12: A variety of tree shape were create by tweaking manual parameters.

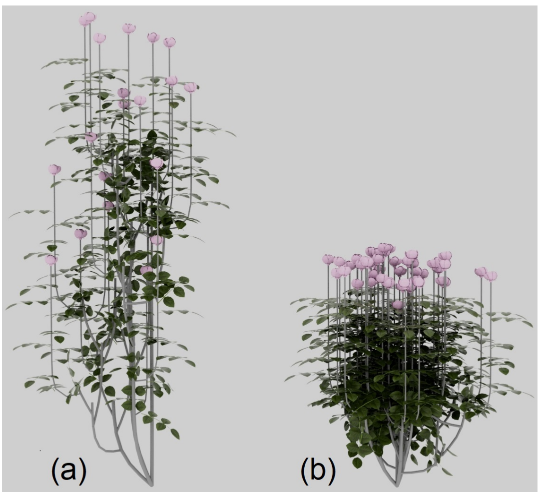


Fig. 13: Two roses are assumed to be of the same variety and made by the different pruning for different purpose.

VI. Discussion

This paper proposes a procedural method for modeling the skeletal structure of shrub roses. The proposed procedure takes into account both the species-specific structural rule and the typical process of artificial care. Our system can generate various rose models reflecting different rose varieties and pruning policies by tweaking several parameters. Although the manual parameters are designed concerning the actual cultivation process, we believe that novice users will be familiar with our system thanks to the direct correspondence between parameters and shape variation.

Our system can produce only well-maintained shrub roses. The growth rules represent a visually appearing rose in the high season with few mistakes in manual care. Therefore, our method cannot generate untended wild roses and tree shapes after the blooming season. Our geometrical algorithm cannot accurately represent the effect of external forces, such as gravity and wind.

Our future work includes procedural modeling of climbing rose. We think it is more challenging than a shrub rose because elaborate consideration is required for the effect of the supporting object and the complex curve shape of the vine. Moreover, we should introduce more types of artificial pruning to generate a wider variety of beautiful roses. The ability to select several types of care would make the method more usable since not all gardeners follow the standard procedure depending on purpose, region, and personalities.

Acknowledgments

This work was supported by PlatinumGames Inc.

References

[1] Sören Pirk, Bedrich Benes, Takashi Ijiri, Yangyan Li, Oliver Deussen, Baoquan Chen, and Radomir Měch. Modeling plant life in computer graphics. In *ACM SIGGRAPH 2016 Courses*, 2016.

[2] Przemyslaw Prusinkiewiczy, Przemyslaw Hananz, Mark Hammely, and Radomir Mech. L-systems: From the theory to visual models of plants. In *Plants to ecosystems: Advances in computational life sciences I*, pages 1–27. CSIRO Publishing, 1996.

[3] Adam Runions, Brendan Lane, and Przemyslaw Prusinkiewicz. Modeling trees with a space colonization algorithm. In *Proceedings of the Third Eurographics Conference on Natural Phenomena*, page 63–70, 2007.

[4] Wojciech Palubicki, Kipp Horel, Steven Longay, Adam Runions, Brendan Lane, Radomír Měch, and Przemyslaw Prusinkiewicz. Self-organizing tree models for image synthesis. *ACM Transactions on Graphics*, 28(3), 2009.

[5] Alan Titchmarsh. *How to Garden: Growing Roses*. BBC Physical Audio, 2011.

[6] Takashi Ijiri, Shigeru Owada, and Takeo Igarashi. Sketch L-system: Global control of tree modeling using free-form stroke. In *Proc of the 6th International Symposium of SmartGraphics*, pages 138–146, 2006.

[7] Jianwei Guo, Haiyong Jiang, Bedrich Benes, Oliver Deussen, Xiaopeng Zhang, Dani Lischinski, and Hui Huang. Inverse procedural modeling of branching structures by inferring l-systems. *ACM Trans. Graph.*, 39(5), 2020.

[8] Przemyslaw Prusinkiewicz, Mark James, and Radomír Měch. Synthetic topiary. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*, page 351–358, 1994.

[9] Ling Xu and David Mould. A procedural method for irregular tree models. *Computers & Graphics*, 36(8):1036–1047, 2012.

[10] Frédéric Boudon, Przemyslaw Prusinkiewicz, Pavol Federl, Christophe Godin, and Radoslaw Karwowski. Interactive design of bonsai tree models. *Computer Graphics Forum*, 22(3):591–599, 2003.

[11] Sören Pirk, Ondrej Stava, Julian Kratt, Michel Abdul Massih Said, Boris Neubert, Radomír Měch, Bedrich Benes, and Oliver Deussen. Plastic trees: Interactive self-adapting botanical tree models. *ACM Trans. Graph.*, 31(4), 2012.

[12] Yong-Joon Kim, Jong-Hwa Woo, Myung-Soo Kim, and Gershon Elber. Interactive tree modeling and deformation with collision detection and avoidance. *Computer Animation and Virtual Worlds*, 26(3–4), 2015.

[13] Lanling Zeng and Guozhao Wang. Modeling golden section in plants. *Progress in Natural Science*, 19(2):255–260, 2009.