Abstract
In this paper we describe Persian floral patterns and explore techniques for animating them. We present several approaches for this dynamic recreation: visualizing pattern symmetries, illustrating their design process, and simulating plant growth. For creating a pleasant illusion of a never-ending movie, we also explore an infinitely cycling effect for self-similar patterns. The construction of animating patterns is started by interactive modeling of plant elements using NURBS. We then use procedural techniques to control the animation.

1. Introduction
Persian floral patterns have been used by professional artists for more than two thousand years to decorate and illuminate poems, religious books, goldware, carpets, and tiles (Figures 1 and 2). The patterns are characterized by curved shapes that are abstractions of plant elements such as flowers, stems, buds and leaves. They have both translational symmetry, repeating the elements using rotation, translation, reflection, and dilational symmetry, using scaled repetition (Figure 3). As in other classes of floral patterns [WZS98], they also use analogy, a more subtle form of repetition in which similar elements are used to induce a pleasing unity of form (Figure 2). Persian floral elements are highly detailed, delicate, and include many variations (Figures 2 and 4). The frequent use of spirals and Eslimi (a decorative leaf) stems (Figures 4a) are the other distinctive features. The patterns are often richly illuminated with a traditional set of colors. Gold and sky blue are frequently used. Therefore, the illuminated patterns are often called Tazhib, which is a derivation of the word gold in Arabic [Tak06], [Mac00].

Many efforts have been made to reproduce Islamic symmetric patterns and characterize their symmetry and geometry ([KS04], [AS92], [AS95]). Our goal is to describe and animate Persian floral patterns, the structure of which is different from purely geometric Islamic patterns. We thus consider the question of how to dynamically create these patterns. Animating patterns may create a mesmerizing illusion of “being alive”. For example, the opening sequence in the Casino Royale movie is based on the use of dynamic patterns [Cas06]. Furthermore, animation may help to understand the complex structure and symmetries of patterns. Finally, animation is an effective way for visualizing their skilful design process.

Problem Specification. There are several strategies to animate patterns. One possibility is to use animation to illustrate the design process used by traditional artists. The other possibility is to visualize the symmetries by dynamically extending a simple motif into a final complex pattern. A yet another way is inspired by the biological process of plant growth. In this case, the floral elements are gradually increased in size, while new details are added. By combining growth and panning, infinitely cycling dynamic patterns can be constructed, creating a pleasant illusion of a never-ending movie. In addition to aesthetic objectives, these dynamic productions may lead to new possibilities for learning and understanding the puzzling geometry of Persian floral patterns.

We begin the construction of animating patterns by mod-
2. Related work

The problem of creating symmetrical patterns has already been investigated in computer graphics. For example, Alexander introduced a Fortran program for generating the 17 symmetry patterns in the plane [Ale75]. Gunn [Gun93] presented a software for interactive visualization of the discrete groups. Grünbaum and Shephard use a more sophisticated computer program based on group theory to generate periodic tilings and patterns in their book [GS86].

Many attempts have also been made for analyzing, designing and reproducing symmetric Islamic patterns and deriving their geometric description ( [AS92], [AS95], [Kap00], [KS04]). These patterns are purely geometric, often dividing the plane into very regular star-shaped regions and polygons. However, Persian floral patterns include curved elements abstracted from real plants. Furthermore, they have dilational symmetry that creates a more dynamic repetition.

Wong et al. examined algorithmic methods for creating floral patterns to fit a particularly shaped region of the plane [WZS98]. They describe some of the principles of traditional floral ornamental design.
The closest work to our dynamic recreation of a traditional art is presented by Xu et al. [XXK06] who introduced animation of Chinese paintings by extracting brush strokes automatically from the image of a given painting. Once the brush strokes have been extracted, animation can be generated by user manipulation at the stroke level. Due to the automatic extraction of brush strokes, the proposed technique has limitations when the painting involves overlapping strokes, dense strokes and variation of texture in the strokes. In our work, plant elements could also be automatically extracted from the scanned images of existing patterns using B-spline tracing techniques [SMA00]. However, to have more precise models and more control of the details of their elements, we interactively model the curves.

By definition, floral patterns are abstractions of real plants. Previous work on animating real plants [PL96] can be a source of inspiration for animating floral patterns.

3. Methodology

To animate Persian floral patterns for different purposes such as growth simulation or demonstrating the design process, we may need to model them differently. In some cases, although the pattern is 2D, the animation may be in 3D. Therefore, our target space for modeling, rendering, and animation is 3D.

A large variety of plant elements can be found in Persian floral patterns. For example, in [Kho00] there is a collection of more than three thousand different drawings of a category of Persian flowers. In our work, we have modeled and animated only a subset of Persian floral patterns.

3.1. Modeling

Persian floral patterns are highly geometric but not always symmetric. Arc and spirals are used extensively, although these are not the only types of curves found in the patterns. The basic elements should be designed using an interactive curve modeler; we have modeled them using NURBS. The control points and their weights provide enough flexibility for interactively modifying the shapes into their final form.

We begin by tracing basic elements of a drawing with a mouse or pen, which generates a sequence of points \( P_i \). If these points are sparse and accurately entered, they are used directly to define an interpolating NURBS curve (a stroke) \( N(a) \), where \( a \) is in \([0, 1]\). The control points \( C_j \) of \( N(u) \) are determined such that \( N(u) \) interpolates \( P_i \) (see [PT95]). If the points \( P_i \) are very dense and noisy, a least square curve approximation (see [PT95]) or multiresolution curve representation [SB99] can be used for detecting \( C_j \). The resulting curves can be repetitively used to form the final composition. As an alternative method, automatic scan conversion techniques can be used to convert scanned drawings to NURBS. In [SMA00], a combination of B-spline curves, corner points, arcs and line segments was automatically extracted from the scanned image of a drawing. This automatic conversion was performed by minimizing the distance of the model from the data and the energy (fairness) of the curves. However, we found that the interactive approach gives more flexibility as well as better possibilities for enforcing geometric and symmetric constraints.

To control repetition and symmetry in the patterns, we use procedural modeling. In the simplest case, this controlling procedure creates several instances of the element curves (NURBS) using basic transformations (Figure 5). In addition, a procedure may be created for different animation strategies, as discussed in Section 3.3.

3.2. Rendering

Persian floral patterns are richly illuminated by a specific collection of colors (e.g. gold and sky-blue). Although the exact production of a gold color is not an easy task, for our purposes we found that the standard Phong reflection model worked well enough. We thus take the advantage of 3D space and the standard graphics API for drawing curves and filling polygons.

3.3. Animation

We consider the following approaches to dynamically create the patterns: visualizing the symmetries, illustrating the design process, and animating growth. In addition to these approaches, for the sake of beauty, we have also created cyclic animation for some of the patterns. For these approaches, we employ separate techniques that are discussed in this section.

3.3.1. Visualizing the symmetry

Conceptualizing the process from the motif to the final composition using conventional media is a challenging task (Figure 6), so its animation is helpful (Figure 7). The approach described below is one of the simplest animation strategies.

Consider a transformation \( A \) that acts on \( M \) composed of one or more motifs. The transformation \( A \) is a combination of \( T_v \) (translation by vector \( v \)), \( R_\theta \) (rotation by \( \theta \)), and reflection (a 3D rotation about the reflection’s axis by \( \theta = \pi \)). In addition to the angle, a pivot point for the 2D rotation and
a pivot axis for the reflection must be determined. For animating this process, we use a smooth transition between $M$ and $A(M)$. This can be done by incrementing the parameters of the transformation (e.g. the rotation angle $\theta$) when the time is increased. To do this, the initial and the final values of the parameters are set in the key-frames. The Figure 7 demonstrates this kind of animation for a sample pattern using snapshots of its animation.

3.3.2. Illustrating the design process

The animation of a pattern can be designed to reveal the step by step progression of the pattern's creation. While this effect is aesthetically appealing (as shown in the Figure 15) illustrating the drawing sequence used by traditional artists can also create a useful learning tool.

Although it is hard to find some precise rules for the drawing Persian floral patterns, there are some basic guidelines that are usually discussed for teaching the traditional design of these patterns [Tak06, Mac00]. For example, the design usually starts from larger, more important elements to the small parts. When there is a branch, it is drawn after the main stem. Spirals are usually drawn from outside to inside. The drawing process progresses from a simple form to a complex shape level by level.

These guidelines can be included into a control procedure (see Section 3.1) for directing the animation. All strokes in motifs are animated first, then the level of detail is gradually increased.

**Drawing the strokes.** To animate all the strokes in motifs, we can use the basic guidelines of the traditional art to define an order. We sort all the strokes from larger elements to smaller ones and from main strokes to the branch strokes. To simulate the effect of live drawing, we gradually elongate the stroke. Figure 8 shows the effect of this technique.

**Levels-of-detail.** The design steps of Persian floral pattern are usually drawn from large scale (low detail) to low scale (high detail) elements. This level-of-detail ap-
proach has been clearly stated in the main traditional texts [Tak06, Mac00]. Figure 9 shows consecutive levels of detail while drawing an Eslimi leaf. Supporting the level-of-detail also helps to create view-dependent models whose details are generated based on the zoom/scale factor.

As evidenced in Figure 9, two kinds of modifications are introduced at different levels of the design process: adding new parts and revising the shape of strokes. For adding new parts, the proposed technique for drawing motif’s strokes can be used again. We can simply draw the new parts as the latest set of strokes by adding it into the end of the stroke list. Revising the shape of strokes can be done by changing control points to locally deform the stokes. For global modification of the strokes, we can use 2D space deformation [SP86]. For the features (e.g., the color, texture and width of strokes), we use feature based 2D morphing techniques [Par02].

3.3.3. Animating growth

Inspired by nature, traditional artists have long tried to capture the beauty of plants in their designs. In this effort, they have abstracted the patterns (including their growth) and changed their shapes to fit their applications and its constraints (base media, number of colors). These floral patterns have been fastidiously refined during their long history [Tak06]. Although in the standard books [Tak06, Mac00, Agh04] the growth aspect of these patterns have not been directly discussed or evidenced, the presence of the growth in these patterns are indirectly induced. Longer stems have more flowers and leaves than shorter stems, and the size of the elements become smaller along the stem ([Agh04]). Spirals are also useful for suggesting growth (more cycles in the spirals means more mature pattern).

In fact, these observations can directly lead us to animation techniques. We animate the growth of a stem by elongating its stroke in time. Small-scale flowers and leaves are placed in suitable positions on the stem. As the stem grows, the flowers and leaves are scaled up by changing their scale factors in key frames. Figure 10 shows several snapshots of the growth animation of a pattern. For more sophisticated growth, substantial changes in flowers and leaves (e.g., conversion of a bud to a flower) can be simulated by using feature based 2D morphing.

3.3.4. Infinite cycling

Since all animations have a limited length, their playback times are also limited. Repeating the animation in an infinite loop provides a potential for creating an illusion of a never-ending animation. However, the transition from the last frame to the first frame usually has a noticeable discontinuity. Is it possible to create never-ending animation without the discontinuity issue? We show that Persian floral patterns (and also other symmetric patterns) can be animated without this discontinuity.

The main idea is that the patterns at the starting time $t_s$ and the end time $t_e$ should be identical. Then the repeat of the animation can create the desired effect. If the pattern has a translational symmetry, we can make a never-ending effect in which the elements seem to be infinitely constructed and moved away from the camera’s viewport. Assume that we want to have a specific number of the generator pattern in the camera’s viewport. For example, in Figure 11 (a), this number is three. To make the animation, we need to add one more copy of the generator as illustrated in Figure 11 (a) (the camera’s viewport has been highlighted). At $t_e$, the viewport of the camera is set such that it includes the extra copy of the generator while one of its original generators is excluded as shown in Figure 11 (b). Although the camera’s viewports at $t_s$ and $t_e$ are different, their containing patterns are exactly the same. Therefore, the smooth transition between these two camera settings and repeating the resulting animation create the desired effect. During this transition, strokes of the
extra copy of the generator can be also dynamically drawn to make the impression of a never-ending construction of the pattern. Figure 11 (c) shows an in-between frame in which the extra copy of the generator is partially drawn.

For patterns with dilational symmetry (e.g. (d) in Figure 11) we can create a more sophisticated cycling that induces the impression of an infinite growth. In these patterns the generator $N_1$ is repeated and scaled down several times. More formally, the copies of the generator are defined as

$$N_i = T(N_{i-1})$$

where $T$ is a contractive affine transformation and for $i = 2, \ldots, k$. If $N_k$ is small enough to be almost invisible during the animation, we can create the infinite growth effect. For doing this, we start with the original pattern $P_t = \bigcup_{i=1}^{k} N_i$ at the time $t_t$ and use $P_c = T^{-1}\left(\bigcup_{i=1}^{k} N_i\right)$ at the time $t_c$. Notice that $T^{-1}(N_2)$ is exactly the same as $N_1$ and in general $T^{-1}(N_{i+1})$ is $N_i$ for $i = 1, \ldots, k-1$. Therefore, $P_c$ has only two differences with $P_t$: the absence of $N_k$ (the smallest copy) and the existence of $T^{-1}(N_1)$ (the largest copy). The absence of $N_k$ is not noticeable. However, we need to handle the issue of the big copy. By appropriate selection of the camera’s viewport, $T^{-1}(N_1)$ can be excluded in $P_c$.

4. Results and Implementation

We have used Maya and its scripting language for implementing this work. Floral elements are designed interactively in the Maya environment. Each animation has a control procedure which is written in the scripting language. For example, Figures 5, 7 and 8 are created from snapshots of a pattern representing a simple flower form. Once we have the control procedure, many variations of the animation can be generated. For examples, in Figure 12 the symmetric expansion of the pattern is animated from a corner and the center, respectively. It is also possible to expand and animate the pattern in a parallel fashion starting from multiple points.

Figure 10 was created using snapshots of a more complex patterns inspired by traditional art (the right pattern in Figure 2).

Figure 13 is based on the patterns that are usually used for decorative title of books. This kind of patterns is called Shamse, which is the Arabic word for the sun; they have rotational symmetry, decorations similar to sun flares, and white space in the centre (used for the title) [Tak06]. We have created animations for the symmetry visualization and design process of this kind of patterns as demonstrated in (Figure 15).

Figure 14 shows several frames of infinite cycling animation for a pattern with dilational symmetry.

5. Conclusion and future work

In this paper, we have explored animation of Persian floral patterns. Our methodology is based on interactive modeling for the strokes and procedural techniques for animations.

Using Maya as implementation environment allowed us to do the basic experiments, and design the necessary prototypes in this research. However, a specialized system for interactive modeling and animation of the patterns is more desirable.

We modify animations by changing the source code of their control procedures. However, a visual programming paradigm for the control procedure would be useful.
Figure 13: A sample of Shamse patterns that are usually used for the title of books.

Figure 14: Frames from the infinite cycling animation of a Persian floral pattern with dilational symmetry.

Acknowledgment

We thank Reza Kianzad, Lynn Mercer, Luke Olsen, Sheelagh Carpendale, and Brendan Lane for their helpful comments and discussions. The support of the Natural Sciences and Engineering Research Council of Canada is gratefully acknowledged.

References


© The Eurographics Association 2008.
Figure 15: Illustration of the design process and Levels-of-details.

