

# Islamic Star Patterns in Absolute Geometry

CRAIG S. KAPLAN

University of Waterloo

and

DAVID H. SALESIN

University of Washington and Microsoft Corporation

---

We present *Najm*, a set of tools built on the axioms of absolute geometry for exploring the design space of Islamic star patterns. Our approach makes use of a novel family of tilings, called “inflation tilings,” which are particularly well suited as guides for creating star patterns. We describe a method for creating a parameterized set of motifs that can be used to fill the many regular polygons that comprise these tilings, as well as an algorithm to infer geometry for any irregular polygons that remain. Erasing the underlying tiling and joining together the inferred motifs produces the star patterns. By choice, *Najm* is built upon the subset of geometry that makes no assumption about the behavior of parallel lines. As a consequence, star patterns created by *Najm* can be designed equally well to fit the Euclidean plane, the hyperbolic plane, or the surface of a sphere.

Categories and Subject Descriptors: I.3.5 [**Computational Geometry and Object Modeling**]: Geometric Algorithms, Languages and Systems; I.3.8 [**Computer Graphics**]: Applications; J.5 [**Arts and Humanities**]: Fine arts

General Terms: Design, Algorithms

Additional Key Words and Phrases: Non-Euclidean geometry, symmetry, tessellations, tilings

---

## 1. INTRODUCTION

The rise and spread of Islamic culture from the seventh century onward has provided us with one of history’s great artistic and decorative traditions. In a broad swath of Islamic rule, at one time extending across Europe, Africa, and Asia, we find artistic treasures of unrivaled beauty. Islamic art encompasses great achievements in calligraphy, stylized floral designs, architecture, and abstract geometric patterns. In this work we focus on the latter category, specifically on *Islamic star patterns* such as the ones catalogued by Bourgoin [1973]. These patterns adorn buildings throughout the Islamic world. They are perhaps best known to Americans and Europeans through the Alhambra Palace in Granada, Spain, one of the jewels of Islamic art [Irving 1931; Stewart 1974].

How were Islamic star patterns originally devised? Unfortunately, very little information about historical techniques survives to the present day. These techniques were a closely-guarded trade secret, passed from master to apprentice and ultimately lost in history [Abas and Salman 1992]. The quest to design star patterns is therefore an intriguing puzzle. As a guide, we have an enigmatic set of examples

---

Authors’ addresses: C. S. Kaplan, School of Computer Science, University of Waterloo, 200 University Avenue West, Waterloo, Ont., N2L 3G1 Canada; email: csk@cgl.uwaterloo.ca; D. H. Salesin, Department of Computer Science and Engineering, University of Washington, Box 352350, Seattle, WA 98195; email: salesin@cs.washington.edu.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or direct commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 1515 Broadway, New York, NY 10036 USA, fax: +1 (212) 869-0481, or permissions@acm.org.  
© 2004 ACM 0730-0301/04/0400-0097 \$5.00

from the past thousand years, accessed via several published collections [Bourgoin 1973; Abas and Salman 1995; Castéra 1999].

One thing we do know is that star patterns are deeply mathematical in nature. The artisans who developed them were well versed in geometry; in their pursuit of mathematical knowledge, early Islamic scholars translated Euclid's *Elements* into Arabic.

We have developed a set of tools called *Najm* (Arabic for “star”) for exploring the design space of Islamic star patterns. In our approach, a tiling is used to guide the placement of motifs. When the tiling is removed and the motifs joined together, the result is a star pattern that can then be decorated or used to drive various computer-aided manufacturing processes. In this article, we present a parameterized family of tilings especially suitable for Islamic star patterns (Sect. 4.1), symmetric motifs to fill the regular polygons in the tiling (Sect. 4.2), and an algorithm to derive motifs for any irregular tiles (Sect. 4.3).

By choice, our construction technique is restricted to those facts of geometry that do not make any assumption about the behavior of parallel lines. We use the erstwhile term *absolute geometry* to refer to this form of geometry. By performing our construction in absolute geometry, we can move seamlessly between designs in the Euclidean plane, in the hyperbolic plane, and on the sphere. Thus we respond to an early suggestion by Lee [1987] that star patterns might be adapted to the hyperbolic plane.

## 1.1 Related Work

Within the field of computer graphics, many systems have been developed for visualizing symmetric designs. Alexander [1975] gave an early demonstration in the second SIGGRAPH conference. More recent examples include Kali [Amenta and Phillips 1996] and Tess [Pedagoguery Software Inc. 2000]. Gunn [1993] created a unified system that permits the visualization of symmetric designs in Euclidean, elliptic, and hyperbolic geometry.

Although Islamic star patterns have been studied by artists and historians for centuries, it is only recently, with the aid of modern algebra and geometry, that a rigorous mathematical treatment of them can be given. Accordingly, many twentieth-century scholars have discussed various analysis and synthesis methods for star patterns.

Grünbaum and Shephard [1992] provide a deep and thorough application of group theory to the study of periodic star patterns. They derive a powerful set of mathematical tools for analyzing patterns in terms of symmetry groups and predicting their properties when elaborated over the entire plane. Ostromoukhov [1998] extends Grünbaum and Shephard's analysis to the seventeen wallpaper groups, and provides a workflow for artists and designers to use in the creation of symmetric ornament. Abas and Salman [1995] carry out this group-theoretic analysis on a library of historical designs. In other work, Abas and Salman [1992], they trace a plausible development of certain patterns from the mathematical tools available to the artisans who created them. In all these cases, the research is largely analytical, with little suggestion of how new star patterns might be constructed.

Dewdney [1993] presents a complete method for constructing designs based on reflecting lines through a regular arrangement of circles. Although this technique could be used to construct some well-known designs, Dewdney admits that he requires many intuitive leaps to arrive at a finished design. Dispot's recent Arabesque software [Dispot 2002] allows the user to construct star patterns using an approach similar to Dewdney's.

In his book, Castéra [1999] presents a rich technique motivated by the practicalities of working with the clay tiles used in traditional architectural settings. He starts out with a hand-placed “skeleton” of 8-pointed stars and flattened hexagons called *safis*, and fills the remaining space with additional shapes. With carefully chosen skeletons, he creates designs of astonishing beauty and complexity. Castéra's designs tend to be centered around a single prominent motif and are not intended to be repeated across the entire plane.

Dunham has a long history of creating ornamental designs in the hyperbolic plane [Dunham 1986b; 1999]. Recently, he has adapted several well-known Islamic geometric designs to the hyperbolic plane [Dunham 2001], though each design is developed by hand and no star patterns are included.

We build star patterns by first specifying a tiling and then filling tiles with motifs. Evidence of such a tiling-based (or at least tiling-aware) construction can be found in the centuries-old Topkapı scroll [Necipoğlu 1995]. Hankin [1925] wrote of his discovery of a Turkish bath where the star patterns on the walls were accompanied by a lightly-drawn polygonal tiling. Wade [1976] elaborates on this construction, presenting what he calls the “point-joining technique.” He specifies that a design should be developed from a tiling by drawing line segments that cross the midpoints of the tiling’s edges. Referring to Hankin, Lee [1987] mentions the “polygons-in-contact technique,” stating that new star patterns might be constructed by searching for polygonal tessellations. Bonner [2000] has built a massive collection of star patterns from tilings, and is the creator of Geodazzlers [Bonner 1997], a set of foldable paper polyhedra decorated with star patterns. Building on the work of Hankin and Lee, Kaplan [2000a] presents a software tool that carries out a tiling-based construction on a small set of hard-coded Euclidean tilings.

We elaborate and improve upon the work of Kaplan [2000a] in several ways. First, we provide a generalized system for symmetric motifs and more useful parameterizations for stars and rosettes. Second, whereas the tilings described in this earlier work were hand-coded and limited in scope, we introduce a novel parameterized collection of tilings suitable for star pattern construction. Third, we present a practical algorithm that derives motifs for hole regions in the tiling. Finally, and most significantly, we eliminate the assumption of Euclidean geometry, allowing us to create designs on the sphere and in the hyperbolic plane.

## 2. MATHEMATICAL BACKGROUND

In this section, we provide a high-level introduction to the mathematical concepts used in this article. We attempt to present only enough background to make the sequel comprehensible to the general reader. More details can be found in some of the excellent texts on geometry, symmetry, and patterns [Coxeter and Moser 1980; Greenberg 1993; Grünbaum and Shephard 1987; Shubnikov and Koptsik 1974; Washburn and Crowe 1992].

### 2.1 Symmetry

A *symmetry* of a set  $S$  is defined as an isometry (a distance-preserving transformation) that maps  $S$  to itself. Isometries are also known as rigid motions, since they move objects around in space without distorting their shapes. Given any set  $S$  in a space equipped with a notion of distance, we may speak of  $G(S)$ , the set of symmetries of  $S$ . The set  $G(S)$  can easily be seen to have an associated group structure via composition of rigid motions. The symmetries associated with  $S$  are therefore called its *symmetry group*. If  $G(S)$  is finite, it must fall into one of two infinite categories: the *cyclic group*  $c_n$  of the  $n$ -armed swastika, or the *dihedral group*  $d_n$  of the regular  $n$ -gon.

Symmetry implies redundancy. A figure with non-trivial symmetries will necessarily contain information that could be copied from another part of the figure. We can factor out all the symmetries of a figure by reducing it to a minimal set of non-redundant information. We call this set a *fundamental region* of a symmetry group. A figure can be reconstructed from a single fundamental region by passing it through every member of its symmetry group.

### 2.2 Euclidean and Non-Euclidean Geometry

In his *Elements*, Euclid formalized geometry by giving a set of five postulates that, when taken as true, yield the rest of geometry through accepted methods of logical deduction. Since then, mathematicians

have proposed many alternate formulations of Euclid's postulates; the intervening centuries have brought mathematical insights that provide a deeper, more solid bedrock upon which to construct geometry.

Historically, the most controversial of Euclid's postulates was always the *parallel postulate*. Euclid himself held off as long as possible before introducing the postulate, finally using it to prove his twenty-ninth proposition. The parallel postulate can be recast in several logically equivalent forms, one of which is known as *Playfair's postulate*: through a point  $P$  not on a line  $l$ , there exists exactly one line  $m$  parallel to  $l$  (where two lines are said to be *parallel* if they have no point in common).

Mathematicians have long sought to eliminate Euclid's parallel postulate from geometry by showing it to be a consequence of the other four. Today we know that consistent non-Euclidean geometries exist in which the parallel postulate does not hold. The original version of the postulate is simply one of three possibilities<sup>1</sup>:

- (1) No line through  $P$  is parallel to  $l$  (*spherical geometry*).
- (2) Exactly one line through  $P$  is parallel to  $l$  (*Euclidean geometry*).
- (3) At least two lines through  $P$  are parallel to  $l$  (*hyperbolic geometry*).

Spherical geometry is the geometry of points on the two-dimensional surface of a three-dimensional ball. Lines are defined as great circles.

Hyperbolic geometry (perhaps the most surprising of the three) can be visualized in several ways [Greenberg 1993]. In this work, we rely most heavily on the *Poincaré model*, in which the points are the interior of a unit disk in the Euclidean plane and lines are arcs of circles that cut the disk at right angles. The Poincaré model is *conformal*: the angle between two arcs correctly reflects the angle between the hyperbolic lines they represent. We feel that this model therefore comes closest to preserving the “shape” of a hyperbolic ornament.

### 2.3 Absolute Geometry

Instead of accepting one of the three parallel postulates given above, what happens if we choose *none* of them? In other words, let us decide to leave the behavior of parallel lines undefined, and develop that part of geometry that follows only from the remaining postulates. This “parallel-agnostic” subset of geometry is occasionally known as *absolute geometry* [Martin 1975]. It contains those logical statements that are simultaneously true in Euclidean, spherical, and hyperbolic geometry, or, equivalently, those statements of Euclidean geometry that do not rely on parallelism. Within this framework, we can still speak of points, lines, distances, angles, and other familiar aspects of geometry. There is even an *absolute trigonometry*, a set of formulae that relate the angles and side lengths of absolute triangles (see Appendix A for details).

What are the points and lines of absolute geometry? The simple answer is that they are undefined abstractions. To define them (in the sense that Euclidean points are defined as ordered pairs of real numbers) is to demand a *model* of absolute geometry. A model of a geometry is a concrete interpretation of its points and lines in which the postulates are theorems. It follows immediately that any model of Euclidean, spherical, or hyperbolic geometry is also a model of the less-constrained absolute geometry. It is from this fact that our approach derives its power. By phrasing our construction technique purely in terms of absolute geometry, we are effectively parameterizing it over the choice of model. As a result, the

<sup>1</sup>Some care must be taken when attempting to graft the three versions of the parallel postulate onto Euclid's geometry. Although the Euclidean and hyperbolic cases work immediately, the spherical version of the postulate can be shown to lead to an inconsistent formal system. However, there exist modern formulations of geometry in which a core set of postulates can be extended by any of the three parallel postulates. Kay [1969] provides one such development, based on the *ruler and protractor postulates* of Birkhoff.

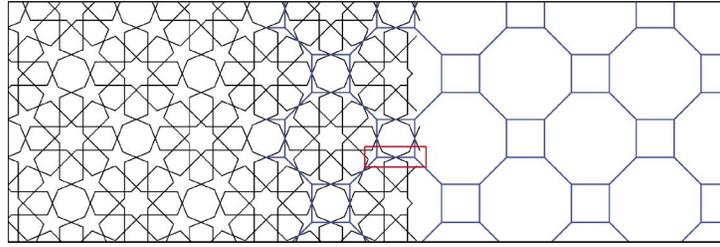


Fig. 1. An example of how a template tiling may be extracted from an existing pattern (the pattern can be found in Bourgoïn [1973, Plate 48]). The eightfold rosettes are recognized as regions of locally high symmetry and bounded by regular octagons. The red rectangle is shown in close-up in Figure 2.

same construction can then be applied seamlessly across the Euclidean plane, the surface of a sphere, and the Poincaré model. As discussed in Section 6, our source code makes this parameterization explicit, making it possible to express geometric constructions without regard to which version of the parallel postulate is eventually adopted.

#### 2.4 Regular Tilings

For any  $p \geq 3$  and  $q \geq 3$ , there is a *regular tiling*  $\{p, q\}$ , consisting of regular  $p$ -gons meeting  $q$  around every vertex [Coxeter and Moser 1980]. This tiling is spherical, Euclidean, or hyperbolic, depending on whether  $1/p + 1/q$  is respectively greater than, equal to, or less than  $1/2$ .

Every  $\{p, q\}$  has an associated symmetry group  $[p, q]$ , generated by reflections in the sides of a right-angled triangle with interior angles  $\pi/p$  and  $\pi/q$ . The tiling  $\{p, q\}$  can then be seen to be composed of copies of this generating triangle by drawing, for every  $p$ -gon, line segments connecting the  $p$ -gon's center to each of its vertices and edge midpoints. For convenience, we will always refer to the generating triangle oriented and labeled as shown in Figure 5.

### 3. APPROACH

Artistic and architectural renderings of Islamic star patterns are richly decorated, often made up of colored regions bounded by interlaced strands. To understand them mathematically, we must find an abstraction of this rendering, one to which various decoration techniques can later be applied. We therefore discard all color and interlacing information, arriving at what Grünbaum and Shephard [1992] call the pattern's *design*: a diagram consisting of straight line segments. A design may be represented by a *planar map* (also known as a *planar subdivision*), a planar embedding of a graph in which every vertex is given a position in the plane [de Berg et al. 2000].

An analysis of star patterns using symmetry groups can yield useful insights. However, as Grünbaum [1984] and Lee [1987] point out, these analyses operate on the whole plane, and can miss higher-order symmetries lurking in localized regions of an overall design.

We adopt a tiling-based approach, in which local regions of high-order  $d_n$  symmetry are bounded by regular  $n$ -gons. The remaining parts of the plane not covered by regular polygons are filled with additional, possibly irregular polygonal tiles. The result is a tiling of the plane that captures the structure of a star pattern at a finer granularity than would be possible using symmetry groups. Figure 1 shows such a tiling being derived from a design.

This tiling-based approach suggests a two-phase construction process. First, a tiling, called the *template*, is chosen to fix the design's overall layout. The template should contain many tiles that are regular polygons in order to express the regions of high local symmetry found in historical designs. We will present a method for specifying symmetric tilings containing many regular polygons. Once the

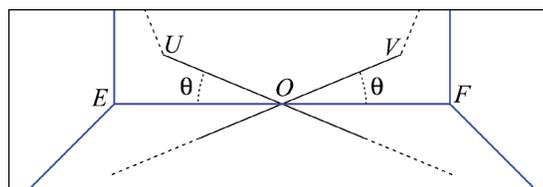


Fig. 2. A close-up of the design and template shown in Figure 1, with labels indicating the important features of the geometry surrounding the *contact position*  $O$ . Point  $O$  lies at the midpoint of the tile edge  $\overline{EF}$ . The two *contact edges*  $\overline{OU}$  and  $\overline{OV}$  form the identical *contact angle*  $\theta$  with  $\overline{OE}$  and  $\overline{OF}$ , respectively. In order to create a perfect crossing at  $O$ , the motifs that meet there must have the same contact angle.

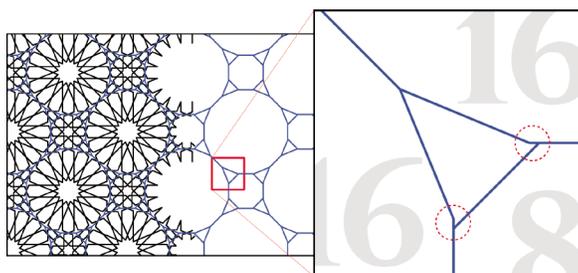


Fig. 3. An example from Bourgoïn [1973, Plate 129] of a pattern for which the derived tiling has unaligned edges. The inset shows that what appears to be a triangular hole is actually an irregular pentagon with two very short edges (shown circled). A tiling-based construction technique needs to handle such tilings gracefully.

tiling is specified, *motifs* are selected or derived to fill the template's tiles. Each motif is a small planar map. Where a motif contacts an enclosing tile, it should do so at the midpoints of the tile's edges. We provide a family of parameterized symmetric motifs for the regular polygonal tiles and an inference algorithm that derives motifs to fill the irregular tiles. The tiling is then elaborated over a given region and populated with motifs. The motifs are joined together to form the final design.

In a template tiling, we refer to the edge of a tile as “aligned” if its midpoint is coincident with the midpoint of an edge of some other tile. The tiling-based construction techniques mentioned in Section 1.1 assume that every edge of every tile is aligned, but it is possible to derive tilings from historical examples that do not have this property. For example, in Figure 3, the irregular pentagonal hole tile has two very short unaligned edges; these arise naturally from the need to place two regular 16-gons and one regular octagon in mutual contact. This tile's motif should not contact the unaligned edges, for there will not be motifs across the edges to link up with those contacts. To prevent such a situation, we mandate that no motif should ever have a vertex incident on an unaligned tile edge.

In a finished design, an aligned tile edge will have four line segments emanating from its midpoint. Although it is not strictly required in creating attractive Islamic ornament, we further require that these four segments be arranged as in Figure 2, so that they can be interpreted as two longer segments that intersect at the edge midpoint. We may then speak of the unique *contact point* of a motif on the edge of its tile, and the *contact angle*  $\theta$  between the motif edge and the tile edge. We obtain a *perfect crossing* if the motifs that meet at a contact point have the same contact angle.

#### 4. CONSTRUCTION METHOD

Using the facts about symmetry and geometry presented in Section 2, and motivated by the discussion of Section 3, we now explain our method in detail.

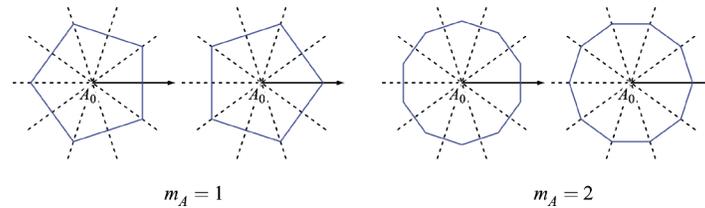


Fig. 4. Examples of valid orientations for on-axis polygons around a fivefold rotational axis. The first and third examples have edge midpoints lying on the designated ray (marked by an arrow). The second and fourth have vertices on the ray. We use the notation  $o_A = e$  and  $o_A = v$  respectively to refer to these two cases.

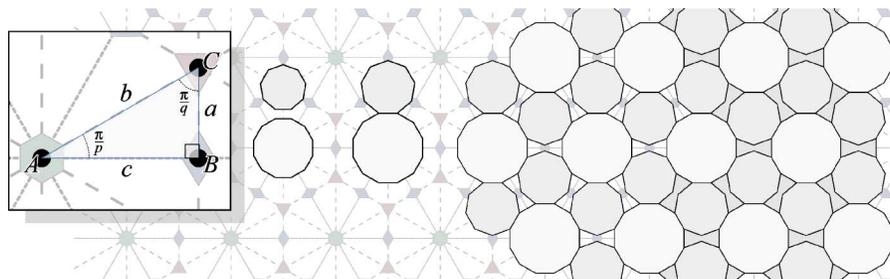


Fig. 5. An example showing step-by-step how the tiling  $\{[6, 3]; 2e, 0, 3e; AC\}$  is constructed. The inset on the left shows the labels on a single fundamental region. Regular dodecagons and enneagons are placed at vertices  $A$  and  $C$ , respectively. They are both oriented so that they present edge midpoints on their designated rays. The polygons are then inflated until they meet and have the same edge length. These polygons can be copied to all other fundamental regions, leaving behind a set of bowtie-shaped holes that are filled with additional tiles.

### 4.1 The Tilings

We present here a novel parameterized space of tilings called *inflation tilings*, which are particularly suitable to the construction of Islamic star patterns.

The regular tiling  $\{p, q\}$ , presented in Section 2.4, has centers of  $p$ -fold, 2-fold, and  $q$ -fold rotation at its face centers, edge midpoints, and vertices, respectively (in Figure 5, these centers of rotation are represented in order by green hexagons, blue diamonds, and red triangles). When the tiling’s symmetry group  $[p, q]$  is visualized through copies of its generating triangle, the rotational axes correspond to triangle vertices  $A$ ,  $B$ , and  $C$ , respectively (as labeled in Figure 5). Inspired by the work of Kaplan and Hart [2001], we use these rotational axes to guide the placement of “on-axis” regular polygons, yielding a new tiling with  $[p, q]$  symmetry.

Consider a single  $p$ -fold rotational axis  $A_0$ . For a regular  $n$ -gon to be compatible with the local symmetry at  $A_0$ ,  $n$  must be a multiple of  $p$ . Furthermore, there are only two orientations of the  $n$ -gon that make it compatible with the  $p$  lines of reflection that pass through  $A_0$ . Given a distinguished ray starting at  $A_0$  and lying on a line of reflection, the  $n$ -gon can intersect the ray either at a vertex or an edge midpoint, as shown in Figure 4. We are therefore left with the following free parameters in defining the on-axis polygon at  $A$ : the multiplier  $m_A = n/p$ , the choice of vertex- or edge-orientation  $o_A$  relative to some ray, and the radius  $r_A$  of the circle in which to inscribe the polygon. The same parameters are available at the  $q$ -fold and 2-fold axes, with the exception that we do not permit  $m_B = 1$ , which would result in a degenerate two-sided tile.

To record orientations unambiguously, we use the designated rays  $\vec{AB}$ ,  $\vec{BC}$ , and  $\vec{CA}$  respectively at vertices  $A$ ,  $B$ , and  $C$ . The symbols  $v$  and  $e$  can then be used to determine whether the polygon should

present a vertex or an edge midpoint on its designated ray. We represent a given set of multipliers and orientations using the notation  $([p, q]; m_A o_A, m_B o_B, m_C o_C)$ , where  $[p, q]$  is the desired symmetry group. We allow any of the multipliers to be zero (indicating that polygons should not be placed at that set of rotational axes), in which case the orientation is irrelevant and can be omitted from the notation. This symbol tells us that regular polygons with  $pm_A$ ,  $2m_B$ , and  $qm_C$  sides should be centered on vertices  $A$ ,  $B$ , and  $C$ , respectively, oriented according to  $o_A$ ,  $o_B$ , and  $o_C$ .

We are left with the choice of how to record the radii  $r_A$ ,  $r_B$ , and  $r_C$ . Ultimately, we will aim to link together motifs inscribed in the on-axis polygons. Therefore, we will usually want to choose values for the radii that force the polygons to come into contact with one another. Although it would be possible to supply explicit radii that achieve these contacts, the scaling operations are fundamental enough that we make them an integral part of the notation. In essence, we introduce notation for declaring constraints that must be met by the radii, and later solve for values of  $r_A$ ,  $r_B$ , and  $r_C$  that satisfy those constraints.

We refer to the scaling process applied to the regular polygons as “inflation.” When it is an on-axis polygon’s turn to be inflated, we center it at the appropriate vertex of the generating triangle, orient it relative to its designated ray, and scale it until it is as large as possible without overlapping any other inflated polygons. We also do not permit the inflating polygon to cross the triangle edge opposite its center; if it did, it would then overlap its own symmetric copy erected on the neighbouring triangle.

We determine the three radii by adjoining to the above notation an “inflation symbol,” describing how and in what order the on-axis polygons should be inflated. The symbol mentions every polygon with a nonzero multiplier exactly once. An optional first part of the symbol, fixing the radii of one or more of the polygons, takes one of the following seven forms. In each case the letters  $A$ ,  $B$ , and  $C$  refer to the polygons centered at those vertices of the generating triangle.

- $A = r, B = r, C = r$  ( $r \in \mathbb{R}^+$ ): Set the radius of the corresponding polygon to  $r$ .
- $AB, BC, AC$ : Inflate the two polygons simultaneously until they meet one another, subject to the constraint that their edge lengths are the same.
- $ABC$ : Inflate all three polygons simultaneously until each one contacts the other two.

Once the radii of one or more polygons are known, any remaining polygons can be inflated. The order in which to inflate them is specified by naming the polygons in a comma-separated list, again using the vertex names of the generating triangle.

The equations required to carry out all these inflations rely on formulae from absolute geometry. Radii for the three on-axis polygons can be solved for in closed form or computed numerically. Algorithms for numerical solutions are given in Appendix B.

To summarize, a tiling is described using the notation  $([p, q]; m_A o_A, m_B o_B, m_C o_C; S)$ , where  $[p, q]$  gives the symmetry group,  $m_A o_A$ ,  $m_B o_B$ , and  $m_C o_C$  give the multipliers and orientations for the on-axis polygons, and  $S$  is the inflation symbol describing how the polygons should be scaled. Using this notation, the Euclidean tiling in Figure 5 is given by  $([6, 3]; 2e, 0, 3e; AC)$ . The multipliers  $m_A$  and  $m_C$ , together with the rotational orders 6 and 3 from the symmetry group, indicate that the tiling will contain regular dodecagons and enneagons. The tiling in Figure 3 is  $([4, 4]; 4e, 4e, 4e; ABC)$ . Additional examples of the many tilings expressible with this notation are given in Figure 6.

## 4.2 The Design Elements

The design elements represent our opportunity to interpret most directly the features of traditional designs. A design element is a “clipping” from history, a fragment of a pattern that has been abstracted from its surroundings and endowed with some number of degrees of freedom. In practical terms a design

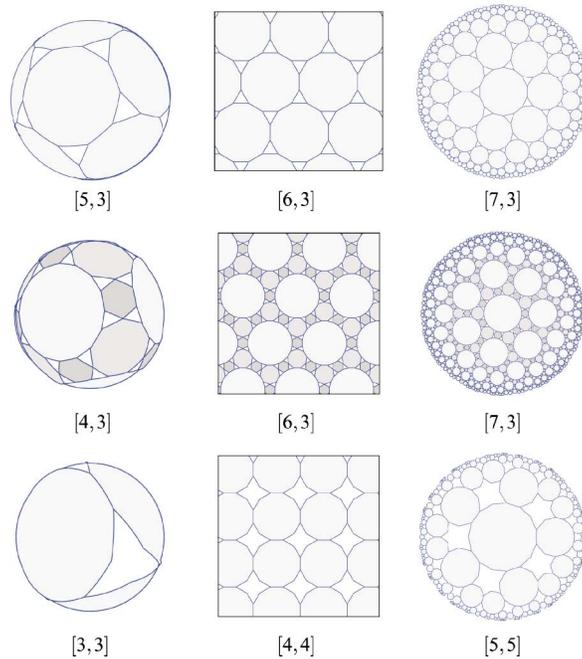


Fig. 6. Examples of tilings that can be constructed using the procedure and notation of Section 4.1. The tilings are of the form  $([p, q]; 2e, 0, 1e; AC)$  in the first row,  $([p, q]; 4e, 3v, 3e; ABC)$  in the second row, and  $([p, q]; 3e, 0, 0; A)$  in the third. The symmetry group is given under each tiling.

element is a function that, when given a regular  $n$ -gon with some radius  $r$ , together with some set of additional parameters, produces a planar map that can be used as a motif for the  $n$ -gon. By capturing the feel of commonly occurring motifs in historical designs, we stand a good chance of creating new designs with a similar spirit.

Let  $\mathcal{P}$  be a regular  $n$ -sided polygon with center  $O$ , inscribed in a circle of radius  $r$ . We represent a design element as a piecewise-linear path that starts at  $M$ , the midpoint of one of the edges of  $\mathcal{P}$ , and wanders around inside  $\mathcal{P}$ . We can obtain a  $d_n$ -symmetric motif by combining all images of the path under the symmetries of the surrounding polygon (Figure 7). During this duplication process, the original path will intersect rotated copies of itself. The intersections occur on successive lines of reflection of  $\mathcal{P}$ . As shown in the figure, we use an integer parameter  $0 < s \leq n/2$  to control how many of these subpaths to keep. The parameter  $s$ , which we have generalized from its standard use in describing star polygons [Lee 1987], allows us to turn a single path into a family of related design elements.

Using this path-based description of design elements, we can now define a family of higher-level procedural models that generate motifs common to star patterns. We implement three important models: *stars*, *rosettes*, and *extended design elements*. Because the rest of *Najm* interacts with design elements as planar maps, and looks only at the geometry incident on the contact points, it is easy to extend this core with new models.

**4.2.1 Stars.** At the heart of Islamic star patterns we find the *star polygon* [Grünbaum and Shephard 1987, Sect. 2.5]. Islamic art features stars with as many as ninety-six points [Castéra 1999]. In our system, a star is constructed from a path consisting of a single line segment that effectively acts as a ray. The segment begins at  $M$  and extends inward in a direction determined by an explicit contact

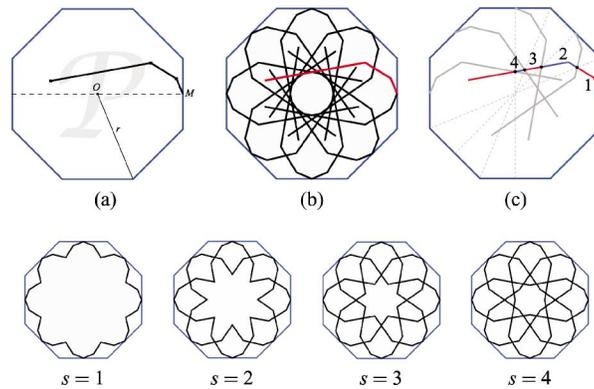


Fig. 7. Our path-based construction applied to a  $d_n$ -symmetric motif inscribed in a regular  $n$ -gon  $\mathcal{P}$ . The initial path is shown in (a). That path is combined with all its  $d_n$ -symmetric copies in (b). In (c), the original path is divided into subpaths by intersections with its copies. The bottom row shows how the parameter  $s$  can be used to control how many of the subpaths to keep.

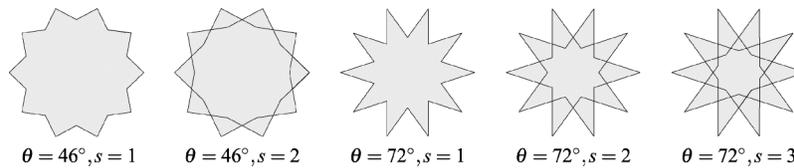


Fig. 8. A demonstration of the effect of the  $\theta$  and  $s$  parameters in the construction of 10-pointed stars.

angle  $\theta$ . It is truncated using the parameter  $s$ , as described above. Figure 8 shows 10-pointed stars under different choices of  $\theta$  and  $s$ .

4.2.2 *Rosettes*. The rosette is one of the most characteristic motifs in Islamic art. A rosette may be viewed as a star to which hexagons have been attached in the concavities between adjacent points (one such hexagon is shown shaded in Figure 9). Each hexagon straddles a line of reflection of the star, and thus has bilateral symmetry.

A rosette can be represented as a two-segment path. The first segment (labeled  $MG$  in Figure 9) becomes part of the outer edge of a hexagon. The path bends at what Lee [1987] calls the “shoulder” (labeled  $G$ ), and continues in a second segment that becomes the hexagon’s flank  $GC$  and the inner star. The path has inherently three degrees of freedom, which can be thought of as the position of the shoulder (two degrees), along with the direction of the flank (one more degree). The problem then is to encode these three degrees of freedom in a way that makes it easy to express rosettes with meaningful, intuitive properties. For convenience, we choose the contact angle as a first parameter. To derive two more parameters, we first attempt to understand what an “ideal” rosette might look like, and then provide as parameters deviations from this ideal.

Lee [1987] provides an ideal construction, demonstrated in Figure 9. Given the surrounding polygon, point  $C$  is found as the point on  $OA$  with  $AC = AM$ . We then construct the line through  $C$  parallel to  $OM$ . Point  $G$  can be found by intersecting this line with the bisector of  $\angle CAM$ .

Unfortunately, the existence of the line through  $C$  parallel to  $OM$  is a direct consequence of Euclid’s parallel postulate, and is therefore not valid in absolute geometry. To achieve generality, we consider instead the constraint that  $G$  should lie on  $MM$ . This new constraint allows us to find  $G$  as the intersection

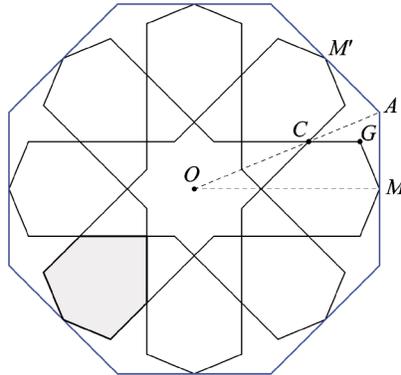


Fig. 9. A demonstration of Lee’s construction of an ideal rosette [Lee 1987]. A rosette is a star to which hexagons have been attached (one such hexagon is shown shaded).

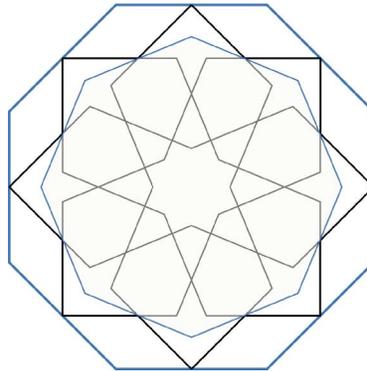


Fig. 10. An extended rosette. The contact edges of the inner element are extended until they meet to become the contacts of the outer element.

of  $\overline{MM'}$  and the bisector of  $\angle CAM$ . The location of  $C$  can be determined as above. We can further adapt this construction to any contact angle  $\theta$  by intersecting the bisector with  $\overline{MA'}$ , where  $A'$  is obtained by rotating  $A$  by an angle of  $\theta$  around  $M$ . The value of  $\theta$  that yields the ideal rosette is then simply  $|\angle AMM'|$ , which depends only on  $n$  and  $r$ .

To encode the remaining two degrees of freedom, we choose real-valued parameters  $h$  and  $\phi$ . The multiplicative factor  $h$  moves  $G$  to a new position  $G'$  on  $\overline{MG}$  such that  $MG' = hMG$ . It is important that  $h$  be a scaling factor and not an absolute length, so that it produces similar designs as the polygon’s radius is varied. The additive term  $\phi$  moves  $C$  to  $C'$  so that  $|\angle C'G'M| = |\angle CG'M| + \phi$ . This term allows the rosette hexagons to be “tapered”, as is seen in some historical patterns. These two parameters are chosen so that their obvious defaults ( $h = 1$  and  $\phi = 0$ ) generate the ideal rosette. Figure 11 shows how points  $G'$  and  $C'$  may be found given  $\theta$ ,  $h$ , and  $\phi$ .

**4.2.3 Extended Design Elements.** When the contact angle of a design element is sufficiently small, it is possible to connect contact edges from adjacent contacts until they meet outside the tile as in Figure 10, forming a larger motif with  $d_n$  symmetry.

We refer to this process as *extension*. Our procedural model for extension takes as input any other procedural model that includes the contact angle  $\theta$  as a parameter and constructs directly an extended

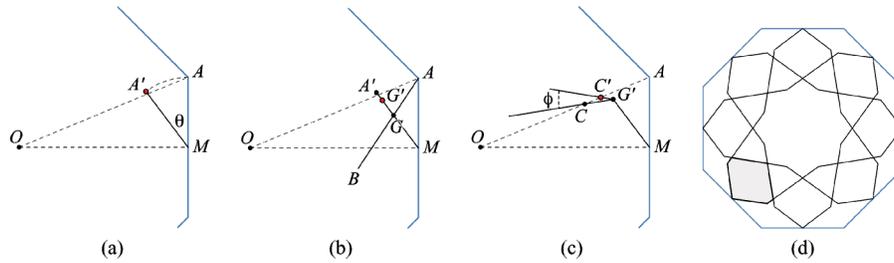


Fig. 11. An illustration of the steps in the construction of a generalized rosette, as explained in Section 4.2.2. In (a), point  $A'$  is obtained by rotating  $A$  about  $M$  by angle  $\theta$ . In (b), the intersection of  $\overline{A'M}$  with  $\overline{AB}$  (the bisector of  $\angle OAM$ ) yields point  $G$ ; we can then find  $G'$  so that  $MG' = hMG$ . In (c), we rotate line  $\overline{CG'}$  by angle  $\phi$  about  $G'$  and intersect with  $\overline{OA}$  to get point  $C$ . Points  $M$ ,  $G'$  and  $C$  determine a two-segment path that, when truncated, yields the final rosette in (d).

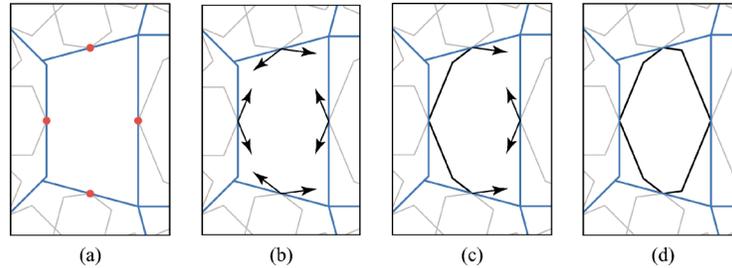


Fig. 12. A sequence of steps in the basic inference algorithm. The initial state of the system is shown in (a), with aligned edge midpoints highlighted in red. These edge midpoints serve as start points of the rays shown in (b). Rays are paired off, and the pairings used to construct the motif. In (c), an intermediate step is shown where two pairings of rays have been consumed. The final inferred motif is given in (d).

version of that model's elements inside a given polygon. Given  $n$ ,  $r$ ,  $s$ , and  $\theta$ , it is possible to compute the necessary  $r'$  and  $\theta'$  for the child element so that when extended, the resulting motif fits perfectly in the outer  $n$ -gon; see Appendix C. The child model is passed  $n$ ,  $r'$ ,  $s - 1$ , and  $\theta'$ , along with unchanged values for any remaining parameters. The resulting motif must be rotated by  $\pi/n$  about its center.

### 4.3 The Inference Algorithm

Once a tiling is specified and design elements are chosen for the regular polygon tiles, the problem remains of finding plausible motifs for the “holes,” the irregular polygonal regions that make up the rest of the tiling. Here, we cannot simply build a library of well-known hole fillers from historical examples; holes come in a wide variety of unpredictable shapes. We need an *inference algorithm*, a generic way of deriving hole geometry from information about the rest of the design. The inference algorithm is where we reap the benefits of the earlier restrictions on contact positions and angles in design elements—they allow us to give a simple algorithm that yields satisfactory results.

As usual, our guiding principle is to create perfect crossings at every contact position. In a nutshell, we can accomplish this goal by extending the contact edges of motifs adjacent to the hole, and cutting each edge off when it meets another extended edge.

The inference algorithm is handed a partially completed design in the form of a template tiling with motifs chosen for some of the tile shapes. An example of such a partially completed design appears in Figure 12(a). For any given non-regular tile  $Q$ , the algorithm should produce a planar map that can be used as a motif to fill copies of the tile.

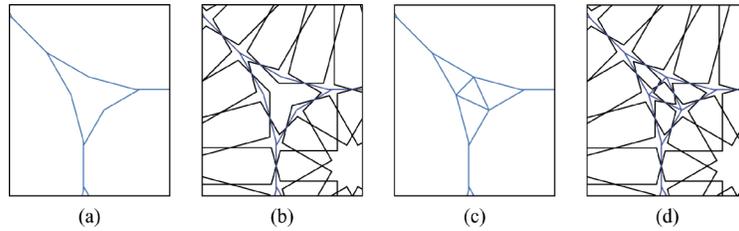


Fig. 13. Subdividing a hole region can improve inference. Inference over the hole shape (a) in  $([4, 4]; 6e, 6e, 6e; ABC)$  does a poor job (b). By subdividing the hole shape (c), a superior inference is obtained (d).

We use a greedy algorithm. First, a list of contact positions is found on  $Q$ 's boundary. The contact positions are the midpoints of the aligned edges of  $Q$ , as shown in Figure 12(a). Each contact is then turned into a pair of rays pointing from the contact point into the tile, as shown in Figure 12(b). The contact angle for a pair of rays at one contact position is determined in one of two ways. If a motif is already present in the neighboring tile, the contact angle is copied over to guarantee a perfect crossing. If not, we use a global default contact angle, chosen as the minimum of the contact angles of the design elements already specified.

Next, for every pair of rays  $\overrightarrow{AB}$  and  $\overrightarrow{CD}$ , we check whether  $\overrightarrow{AB}$  and  $\overrightarrow{CD}$  intersect at some point  $P$ , and if so whether  $P$  is inside the tile. Every pairing of rays that meets these conditions is assigned a weight  $w = AP + CP$  and stored in a list, sorted by weight.

Finally, we traverse the list, considering each pairing  $(\overrightarrow{AB}, \overrightarrow{CD})$  in turn. If neither ray in the pairing has been used yet, we add the line segments  $\overline{AP}$  and  $\overline{CP}$  (where  $P$  is the intersection point of the rays) to the motif being constructed and mark both rays as used. Figure 12(c) shows a step in this traversal.

Although the algorithm above is not guaranteed to always find pairings for every ray, it almost always performs very well in practice. (In fact, nearly all of the results shown in this article used just this simple algorithm. The one exception is the last row of Figure 15, in which the inferred motif was modified by hand.)

Occasionally, though, as Figure 13, shows complex holes can benefit from an intermediate subdivision step. Subdividing a nonconvex polygonal tile by inserting a central polygon will tend to yield a motif that better fills the tile. We find such a polygon using a heuristic that attempts to identify concave pockets in the original tile. The centers of the pockets are then joined into a new polygon.

For the purposes of subdivision, we refer to a polygon vertex as “eligible” if the interior angle of the polygon at that vertex is greater than  $\pi$  and both of the two adjacent polygon edges are aligned. This property will break the polygon's boundary into runs of eligible vertices separated by ineligible vertices. We form a new polygon by linking together the middle of every run, consisting of the central vertex or the central two vertices if there are respectively an odd or even number of vertices in the run. This central polygon will induce a subdivision of the original hole. The algorithm then infers geometry for each of the polygons in the subdivision and assembles the inferred motifs into a single solution for the hole tile.

## 5. DECORATION

We can use the high degree of symmetry of our template tilings to simplify decoration and rendering. We compute the restriction of the overall design to a single generating triangle. This restricted map, which we call the “fundamental map,” contains all geometric information necessary to render any amount of the final design. Furthermore, we can create a decorated design by decorating only one or two copies of the fundamental map. Figure 14 shows examples of decorated fundamental maps.

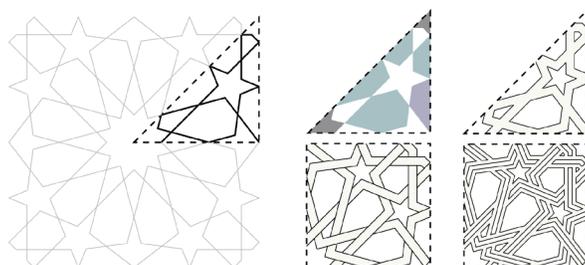


Fig. 14. Examples of decoration styles. The undecorated fundamental map is shown on the left, followed by the filled, outline, interlaced, and outlined-interlace decoration styles. To make the interlaces consistent in the latter two cases, decoration must be carried out on two adjacent fundamental maps.

*Filling.* A simple and effective decoration style is to color the faces of the fundamental map, including faces bordered by the generating triangle. This style emulates the many real-world examples executed using colored clay tiles. Because the designs produced by our construction technique have only 2-valent and 4-valent vertices, the map can be 2-colored. Our system automatically 2-colors the fundamental map as a basis for user selection of face colors. Following tradition, one set of faces in the 2-coloring will typically be left white and the other set will receive a range of colors.

*Outlining.* We can choose instead to simulate the “grout” of a real-world tiling by thickening the edges of the fundamental map. In the Euclidean plane, this operation is straightforward; to endow a line  $l$  with thickness  $w$ , we construct the two parallels at distance  $w/2$  to  $l$ . Unfortunately, these parallels are not well defined in absolute geometry. On the other hand, parallelism is not the defining quality of a thickened line, merely a convenient Euclidean equivalence. What we are really after are the loci of points of constant perpendicular distance  $w/2$  from  $l$ . These are called *equidistant curves*, and they are always uniquely defined. In the Euclidean plane, they are the usual parallels. On the sphere, an equidistant curve is just a small (i.e., non-great) circle; in the Poincaré model, it is a circle that does not intersect the unit disc at right angles [Greenberg 1993, Chap. 10]. Where two thickened line segments meet, we must perform a mitered join. A formula for mitered join in absolute geometry is given in Appendix D.

*Interlacing.* The designs produced by our technique can be interpreted as a collection of intersecting *strands*, some infinite and some closing back on themselves in loops. Grünbaum and Shephard [1992] show that the strands can be drawn with a consistent interlacing, where a given strand passes alternately over and under the strands it intersects. The interlaced decoration style can be derived from the outline style by drawing additional curves at every crossing to suggest the over-and-under relationship. As shown in Figure 14, the over-and-under relationship must be determined over two copies of the fundamental map, a map together with a copy reflected along one of the edges of the generating triangle. This larger map covers a fundamental region of  $[p, q]^+$ , the *orientation-preserving* subgroup of  $[p, q]$  [Coxeter and Moser 1980, Sect. 4.4].

*Combining Styles.* In practice, designs are rendered using some combination of the styles above. The most common combinations are the superposition of an outlined or interlaced rendering over a filled rendering. In some cases, we may also think of composing various styles. Consider that an interlaced rendering can itself be considered a kind of planar map (it is not a planar map because some vertices are connected by equidistant curves, not straight lines). This new map can now be outlined. The result is a composed outline-interlace style, shown in Figure 14.

## 6. IMPLEMENTATION

*Najm* is written in C++. The system is divided into two layers. The lower layer is a library that provides an abstract interface to absolute geometry. The main application suite is then written in a geometry-independent way on top of it. By hiding all specific knowledge of the Euclidean, spherical, and hyperbolic planes behind the abstraction of absolute geometry, we need only write the application layer once. This factoring has helped to clarify the nature of star pattern design by shielding the top-level code from unnecessary detail and repetition.

We implement the absolute geometry library in a typesafe and efficient manner by using explicit specialization of templated classes in C++ [Lippman and Lajoie 1998, Sect. 16.9]. For example, a generic `point<T>` class is declared but not defined. The generic declaration is then overridden by defining three specialized classes `point<Spherical>`, `point<Euclidean>`, and `point<Hyperbolic>`. A client can write generic code that manipulates objects of type `point<T>`, and at compile time, the code will be instantiated with one of the concrete implementations. This architecture is the compile-time analogue of a small class hierarchy, but without the run-time overhead of indirection. See Appendix E for further details on replicating the tilings in the different geometries.

Our implementation highlights the deep distinction between a geometry and its models, a distinction that most people are not generally used to making. Typically, we see the formal Euclidean plane (the geometry) as being indistinguishable from its representation in Cartesian coordinates (the model).<sup>2</sup> While formally the absolute plane does not have a coordinate system, it is still meaningful to speak of, say, the distance between two absolute points in a conceptual way—even if you cannot compute it—precisely because the axioms of absolute geometry assert the existence of such a distance metric. Points and distances are both made concrete as part of the model. A client of the library can therefore request the distance between two points, knowing that the appropriate metric will be used when the client code is instantiated with one of the three models of absolute geometry.

Although the full implementation of *Najm* is not publicly available, many of its techniques have found their way back into Taprats [Kaplan 2000b], the earlier implementation by Kaplan. Taprats is available for experimentation as a Java applet and as a downloadable application. It provides a demonstration of the operation of *Najm* when restricted to the Euclidean plane.

## 7. RESULTS

Figures 15 and 16 show a variety of examples of Islamic star patterns generated using *Najm*. Each column features a design rendered in spherical, Euclidean, and hyperbolic geometries. Note that the structure of star patterns reflects the curvature of the space in which it is embedded. For instance, the patterns in the second column of Figure 15 consist of 10-pointed stars on the sphere, 12-pointed stars in the Euclidean plane, and 14-pointed stars in the hyperbolic plane. Intuitively, this fact makes sense: the curvature is a measurement of the “amount of space” around each point. As curvature decreases and we move from the sphere to the Euclidean plane to the hyperbolic plane, the same underlying pattern accommodates stars with ever larger numbers of points.

We have also experimented with using *Najm* to drive various computer-aided manufacturing systems. Several Euclidean and spherical examples are shown in Figure 17.

## 8. FUTURE WORK

In this article, we have presented a construction technique for creating a broad set of traditional Islamic star patterns, as well as interesting new designs. We have further shown how these patterns can be

<sup>2</sup>Modern conceptions of geometry seek to erase the distinction between a geometry and its models by ensuring that all models are isomorphic (in which case the model can truly be said to “be” the geometry). Such geometries are called *categorical*.

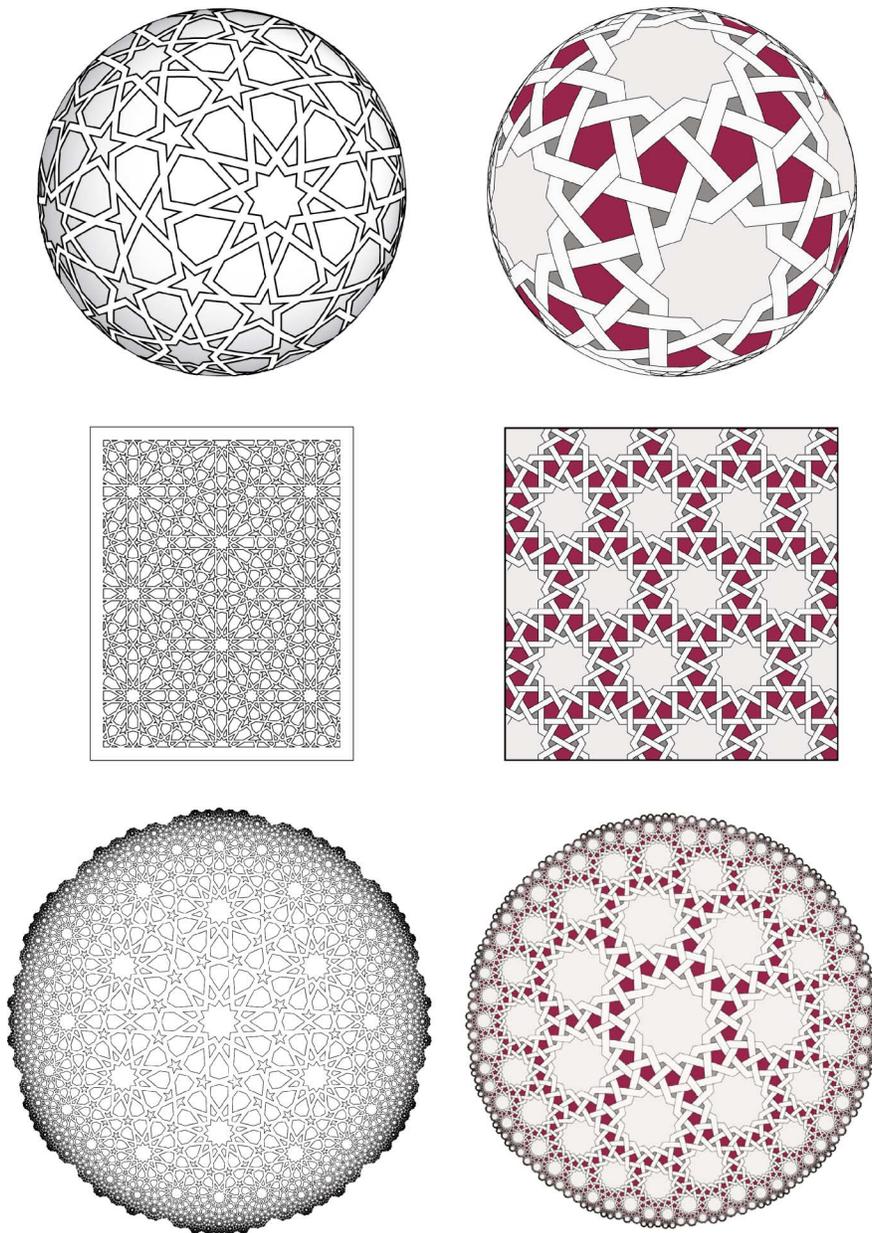


Fig. 15. Samples of Islamic star patterns that can be produced using *Najm*. To provide a basis for comparing patterns across geometries, each column presents a single conceptual design interpreted in each of the three different geometries. The notation for the underlying tilings is  $(\{p, q\}; 2e, 0, 3e; AC)$  in the first column and  $(\{p, q\}; 2e, 0, 0; A)$  in the second.

constructed in a way that is independent of Euclid's parallel postulate, allowing them to be adapted to the sphere or to the hyperbolic plane in addition to the Euclidean plane. However, there are still tremendous opportunities for future work on creating star patterns with computers. We sketch some of the most exciting such directions here.



Fig. 16. More star patterns created using *Najm*. The tilings are of the form  $([p, q]; 3v, 2v, 3e; ABC)$  in the first column, and  $([p, q]; 3v, 0, 4e; AC)$  in the second.

*Better Decoration Tools.* The decoration tools provided by *Najm* are quite flexible, but still require manual intervention in many cases. Some of these cases might be automated by borrowing from traditional rules of star pattern design. For example, Castéra [1999] points out that for certain classes of star patterns, there is a “correct” choice of band width for the outlined and interlaced decoration

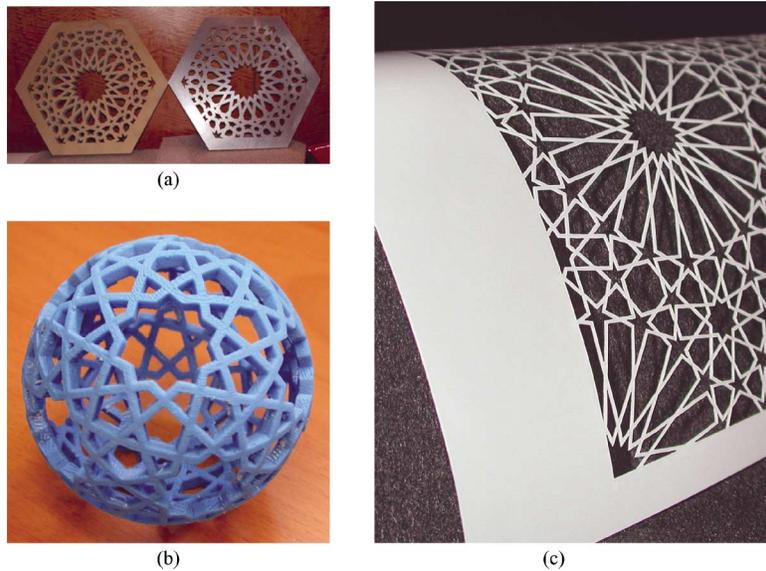


Fig. 17. Star patterns fabricated using various computer-aided manufacturing techniques: a waterjet cutter in (a), a fused-deposition rapid prototyping system in (b), and a CO<sub>2</sub> laser cutter in (c).

styles. In addition, there are conventions that govern the choice of colors and their distribution over the regions of the design. Some automation could be applied to make these sorts of decoration choices automatically.

*The Use of Optimization.* There are cases where the simple inference algorithm of Section 4.3 fails to discover what is historically the correct motif for a template tile. While layers of heuristics might be heaped upon the basic inference algorithm to account for these special cases, it is always more satisfying to discover general principles. To this end, the inadequacies of the inference algorithm may be surmountable through the use of optimization. An optimization procedure, with an aesthetic evaluation as its objective function, might be used to seek a configuration that achieves maximal visual balance.

*Strange Stars.* For what sets of integers can we construct attractive periodic star patterns in which there are  $k$ -pointed stars for every  $k$  in the set? Many simple combinations, such as the sets  $\{6, 9, 24\}$ ,  $\{8, 12\}$ , and  $\{9, 12\}$  follow immediately from the tiling notation of Section 4.1 or a review of historical examples. But we can accept a little flexibility by considering polygons that are nearly regular, in which we can inscribe motifs that are not-quite-perfect stars.

## APPENDIX

### A. ABSOLUTE TRIGONOMETRY

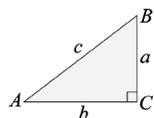
Trigonometry is what allows us to derive all angles and side lengths of a triangle from a subset of those values. Not all properties of Euclidean trigonometry apply in the non-Euclidean planes; for example, the fact that the interior angles of a triangle sum to  $\pi$  is equivalent to the parallel postulate. However, once a suitable standard for measuring distances and angles has been specified, it becomes possible to give hyperbolic and spherical versions of many common Euclidean identities [Kay 1969, Chap. 10; Greenberg 1993, Chap. 10].

It is also possible to provide generic trigonometric functions for absolute geometry. Such functions were originally proposed by Bolyai, and later expanded upon by De Tilly [Bonola 1955, P. 113]. As with the rest of our absolute geometry library, these functions will eventually require separate implementations for the spherical, Euclidean, and hyperbolic cases. The advantage is that a client of the library can solve for properties of triangles without knowing which specific geometry will ultimately be used.

From Bolyai and De Tilly, we define the absolute trigonometric functions  $\bigcirc(x)$  and  $E(x)$ . The function  $\bigcirc(x)$  gives the circumference of a circle of radius  $x$ . Given a line segment  $s$ ,  $E(x)$  is the ratio of the length of an equidistant curve erected at distance  $x$  from  $s$  to the length of  $s$  (this ratio can be seen to be independent of the choice of  $s$ ). They can be defined in cases thus:

	<i>spherical</i>	<i>Euclidean</i>	<i>hyperbolic</i>
$\bigcirc(x)$	$2\pi \sin x$	$2\pi x$	$2\pi \sinh x$
$E(x)$	$\cos x$	$1$	$\cosh x$

From  $\bigcirc(x)$  and  $E(x)$  it is easy to define their ratio  $T(x) = \bigcirc(x)/E(x)$  and the two inverses  $\bigcirc^{-1}(x)$  and  $T^{-1}(x)$  (the inverse of  $E(x)$  is undefined in the Euclidean case). These functions can also be used to define the following absolute trigonometric identities, for a triangle  $\triangle ABC$  with right angle at  $C$ :



$$\begin{aligned} \bigcirc(a) &= \bigcirc(c) \sin A & \cos A &= E(a) \sin B \\ \bigcirc(b) &= \bigcirc(c) \sin B & \cos B &= E(b) \sin A \\ E(c) &= E(a)E(b) \end{aligned}$$

Furthermore, the sine law for Euclidean triangles generalizes naturally to any absolute triangle:  $\bigcirc(a)/\sin A = \bigcirc(b)/\sin B = \bigcirc(c)/\sin C$ .

## B. INFLATION OPERATIONS

Below are the formulae needed to perform the inflation operations discussed in Section 4.1. Some definitions simplify the presentation to follow. As always, let  $\triangle ABC$  be the generating triangle of  $[p, q]$  with right angle at  $B$ , and let  $([p, q]; m_A o_A, m_B o_B, m_C o_C)$  be given. Let variables  $\alpha, \beta$ , and  $\gamma$  represent a permutation of the triangle vertices  $A, B$ , and  $C$ . If  $m_\alpha$  is nonzero, let  $\mathcal{P}_\alpha$  be the regular polygon centered at vertex  $\alpha$ .

The boundary of  $\mathcal{P}_\alpha$  intersects the two triangle edges  $\alpha\beta$  and  $\alpha\gamma$ ; define the *extent* of  $\mathcal{P}_\alpha$  on a triangle edge to be the length of the part of the edge that is contained inside  $\mathcal{P}_\alpha$ . For every ordered pair  $(\alpha, \beta)$  of triangle vertices, we can consider the extent of  $\mathcal{P}_\alpha$  on triangle edge  $\alpha\beta$ , which we denote by  $l_{\alpha\beta}$ . There are therefore six possible extents to consider:  $l_{\alpha\beta}, l_{\alpha\gamma}, l_{\beta\alpha}, l_{\beta\gamma}, l_{\gamma\alpha}$ , and  $l_{\gamma\beta}$ .

If  $m_\alpha$  is nonzero, then  $\mathcal{P}_\alpha$  is a regular  $n$ -gon centered on vertex  $\alpha$  with some radius  $r_\alpha$ . The extent  $l_{\alpha\beta}$  can take on one of two possible values. If  $\mathcal{P}_\alpha$  has a vertex lying on edge  $\alpha\beta$ , then  $l_{\alpha\beta} = r_\alpha$ . Otherwise,  $\mathcal{P}_\alpha$  has an edge midpoint lying on  $\alpha\beta$ , in which case the extent can be obtained from  $T(l_{\alpha\beta}) = T(r_\alpha) \cos \frac{\pi}{n}$ . Note that this calculation can be reversed as well; given one of a polygon's extents, we can determine the radius.

*Case 1: Inflating a Polygon to Another Polygon.* In this case, we have  $\mathcal{P}_\alpha$ , a regular  $n_\alpha$ -gon with fixed radius  $r_\alpha$  centered at vertex  $\alpha$ , and  $\mathcal{P}_\beta$ , a regular  $n_\beta$ -gon at vertex  $\beta$ . We wish to scale  $\mathcal{P}_\beta$  until it touches  $\mathcal{P}_\alpha$ . Let  $d$  be the length of triangle edge  $\alpha\beta$ . From the definitions above, this is a fairly simple relationship to solve algebraically. We can easily determine the value  $l_{\alpha\beta}$ , and then we solve for the value of  $r_\beta$  that gives  $l_{\beta\alpha} = d - l_{\alpha\beta}$ .

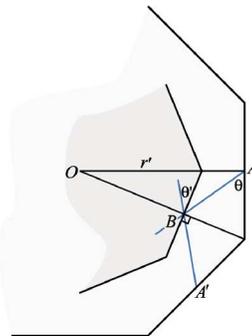
*Case 2: Inflating a Polygon to the Generating Triangle.* Here, the inflation of regular  $n_\alpha$ -gon  $\mathcal{P}_\alpha$  centered at  $\alpha$  is not constrained by any other regular polygon, and so we inflate it until it touches  $\beta\gamma$ , the edge of the generating triangle opposite  $\alpha$ . Let  $d$  be the perpendicular distance from  $\alpha$  to the opposite edge of the triangle. We assume that  $\alpha$  is  $A$ , or  $C$ , since we then have the simpler case that  $d$  is the length of one of the triangle edges. We omit the more complicated case  $\alpha = B$  because it is less useful in constructing practical tilings. Suppose  $\alpha = A$ . Then  $d = AB$  because of the right angle at  $B$  and we can simply set  $r_\alpha$  so that  $l_{\alpha\beta} = d$ . By the definition of extent, we will either have  $r_\alpha = d$  or  $T(r_\alpha) = T(d)/\cos\frac{\pi}{n}$ . A similar argument yields the solution for the case  $\alpha = C$ .

*Case 3: Simultaneous Inflation of Two Polygons.* Here, we have regular polygons  $\mathcal{P}_\alpha$  and  $\mathcal{P}_\beta$  with  $n_\alpha$  and  $n_\beta$  vertices, and we wish to scale the two polygons until they touch, subject to the constraint that they have the same side length. Again, let  $d$  be the length of the shared triangle edge  $\alpha\beta$ . Once the two polygons are scaled, they will have the same side length; let this length be represented by  $x$ . Using some trigonometry, we can give formulae for  $l_{\alpha\beta}$  and  $l_{\beta\alpha}$  in terms of  $x$ . Specifically,  $\circ(l_{\alpha\beta}) = \circ(x)/\sin\frac{\pi}{n_\alpha}$  or  $\circ(l_{\alpha\beta}) = T(x)/\tan\frac{\pi}{n_\alpha}$  when  $\mathcal{P}_\alpha$  respectively has a vertex or an edge midpoint on  $\alpha\beta$ . One of two identical formulae determine  $l_{\beta\alpha}$  from  $x$  and  $n_\beta$ . Since  $n_\alpha$  and  $n_\beta$  are given, the equation  $l_{\alpha\beta} + l_{\beta\alpha} = d$  has  $x$  as its single unknown. A solution for  $x$  can be used to back out final values for  $r_\alpha$  and  $r_\beta$ . In the implementation, we observe that the expression  $l_{\alpha\beta} + l_{\beta\alpha} - d$  is monotonic in  $x$  and solve for  $x$  numerically using binary search.

*Case 4: Three-Way Pinning.* In this most complicated case, we have only the inflation symbol  $ABC$ , indicating that all three polygons should be inflated until each one touches the other two. Our goal is to calculate radii  $r_A$ ,  $r_B$ , and  $r_C$  for regular polygons  $\mathcal{P}_A$ ,  $\mathcal{P}_B$ , and  $\mathcal{P}_C$ . Although it is possible to solve this problem in closed form, the algebra involved is quite grueling. Instead, observe that we can build a numerical solution using the results of previous cases. Given some value for  $r_A$ , we can inflate both  $\mathcal{P}_B$  and  $\mathcal{P}_C$  until they meet  $\mathcal{P}_A$  as in case 1, yielding candidate values for  $r_B$  and  $r_C$ . We can then decide how close  $\mathcal{P}_B$  and  $\mathcal{P}_C$  come to touching each other by computing  $l_{BC} + l_{CB} - d$ , where  $d$  is the length of triangle edge  $BC$ . This expression is a monotonic function of  $r_A$ , and so we can search for a solution to  $l_{BC} + l_{CB} - d = 0$  numerically using binary search. The final value for  $r_A$  determines the values for  $r_B$  and  $r_C$ .

### C. EMBEDDING EXTENDED DESIGN ELEMENTS

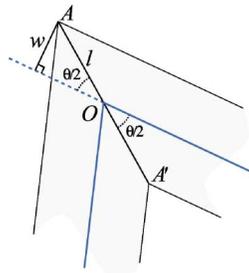
When working with an extended design element, we are given  $n$  and  $r$  describing a regular polygon, and the contact angle  $\theta$  of the motif to inscribe in that polygon. We wish to find  $r'$  and  $\theta'$  so that an inner design element with radius  $r'$  and contact angle  $\theta'$  can be extended as in Figure 10 to exactly fit the original  $n$ -gon.



In our implementation, we compute values for  $r'$  and  $\theta'$  numerically. Given  $n$ ,  $r$ , and  $\theta$ , we construct a regular  $n$ -gon of radius  $r$ . We then compute the location of  $B$  as the intersection of the two blue motif segments, as shown. We can use  $B$  to find  $\alpha = \angle ABA'$  and compute  $\theta'$  from the observation that  $4\theta' + 2\alpha = 2\pi$ . Furthermore, we can erect a perpendicular from  $B$  and find its intersection with the segment  $\overline{OA}$  to obtain the inner radius  $r'$ .

#### D. NON-EUCLIDEAN MITERED JOIN

Suppose that we wish to perform a mitered join of two line segments that meet at point  $O$  with an angle of  $\theta$ , as shown in the diagram on the left. We wish to find the two points  $A$  and  $A'$ . We know that these points lie on the bisector of the two line segments and are equidistant from  $O$ . It remains to determine the distance  $l = OA = OA'$ . The distance  $l$  can be found via a direct application of one of the identities of absolute trigonometry:  $\odot(l) = \odot(w) / \sin \frac{\theta}{2}$ .



#### E. REPLICATION

One important aspect of the library implementation that changes drastically from geometry to geometry is the algorithm that fills a region of the plane with copies of a symmetry group's fundamental unit. Each geometry has a specialized structure that calls for a tailored algorithm.

The sphere permits the simplest replication process. There are three regular spherical symmetry groups:  $[3, 3]$ ,  $[3, 4] = [4, 3]$  and  $[3, 5] = [5, 3]$  (for the purposes of creating star patterns, we disregard the so-called *prismatic groups*  $[2, q]$  and  $[p, 2]$ ). These three groups are finite, so we precompute rigid motions for all copies of the generating triangle and store them in tables. No region is specified; the sphere is simple enough that we always draw the entire pattern.

The Euclidean groups  $[3, 6] = [6, 3]$  and  $[4, 4]$  are infinite, so we need an algorithm that fills only a region. We assemble fundamental units into a *translational unit*, a region that can be repeated to fill the plane using translations alone. This translational unit consists of twelve triangles in a hexagon for  $[3, 6]$  and eight triangles in a square for  $[4, 4]$ . Copies of the translational unit can then be replicated to cover any rectangular region, using an algorithm presented by Kaplan and Salesin [2000, Sect. 6.4].

Replication in the hyperbolic groups presents the greatest challenge. Fortunately, efficient algorithms already exist, including remarkable table-driven systems based on the theory of Automatic Groups [Epstein et al. 1992; Levy 1993]. We base our code directly on the pseudocode presented by Dunham et al. [1981, 1986a]. The regions we fill are discs centered at the origin in the Poincaré model.

#### ACKNOWLEDGMENTS

This project originated with a course taught by Mamoun Sakka at the University of Washington. Tony Lee and Jay Bonner both provided feedback about the history and construction of star patterns. Doug

Dunham and John Hughes helped with details of Euclidean and non-Euclidean geometry. Craig Chambers and Andrei Alexandrescu gave feedback on the use of C++ templates in *Najm*. Victor Ostromoukhov helped improve the exposition in a number of places. Thanks also to Carlo Séquin, James McMurray, and especially to Nathan Myhrvold for their generosity with time and resources in manufacturing the various physical models of star patterns. Finally, thanks to the anonymous referees whose many valuable comments improved the article.

## REFERENCES

- ABAS, S. AND SALMAN, A. 1992. Geometric and group-theoretic methods for computer graphics studies of Islamic symmetric patterns. *Comput. Graph. For.* 11, 1, 43–53.
- ABAS, S. J. AND SALMAN, A. S. 1995. *Symmetries of Islamic Geometrical Patterns*. World Scientific.
- ALEXANDER, H. 1975. The computer/plotter and the 17 ornamental design types. In *Proceedings of SIGGRAPH'75*. ACM, New York, 160–167.
- AMENTA, N. AND PHILLIPS, M. 1996. Kali. <http://www.geom.umn.edu/java/Kali/>.
- BONNER, J. 1997. Geodazzlers. <http://www.dstoys.com/GD.html>.
- BONNER, J. F. 2000. *Islamic Geometric Patterns: Their Historical Development and Traditional Methods of Derivation*. Unpublished.
- BONOLA, R. 1955. *Non-Euclidean Geometry*. Dover Publications.
- BOURGOIN, J. 1973. *Arabic Geometrical Pattern and Design*. Dover Publications.
- CASTÉRA, J.-M. 1999. *Arabesques: Decorative Art in Morocco*. ACR Edition.
- COXETER, H. S. M. AND MOSER, W. O. J. 1980. *Generators and Relations for Discrete Groups*. Springer-Verlag, New York.
- DE BERG, M., VAN KREVELD, M., OVERMARS, M., AND SCHWARZKOPF, O. 2000. *Computational Geometry: Algorithms and Applications*, 2nd ed. Springer-Verlag, New York.
- DEWDNEY, A. 1993. *The Tinkertoy Computer and Other Machinations*. W. H. Freeman, 222–230.
- DISPOT, F. 2002. Arabesque home page. <http://www.wozzeck.net/arabesque/index.html>.
- DUNHAM, D. 1986a. Hyperbolic symmetry. *Comput. Math. Appl.* 12B, 1/2, 139–153.
- DUNHAM, D. 1986b. Creating hyperbolic escher patterns. In *M.C. Escher: Art and Science*, H. S. M. Coxeter, M. Emmer, R. Penrose, and M. L. Teuber, Eds. Elsevier Science Publishers B.V., 241–247.
- DUNHAM, D. 1999. Artistic patterns in hyperbolic geometry. In *Bridges 1999 Proceedings*, R. Sarhangi, Ed. 139–149.
- DUNHAM, D. 2001. Hyperbolic Islamic patterns—A beginning. In *Bridges 2001 Proceedings*, R. Sarhangi, Ed. 247–254.
- DUNHAM, D., LINDGREN, J., AND WITTE, D. 1981. Creating repeating hyperbolic patterns. In *Computer Graphics (Proceedings of SIGGRAPH)*. ACM, New York, 215–223.
- EPSTEIN, D. B. A., CANNON, J. W., HOLT, D. F., LEVY, S. V. F., PATERSON, M. S., AND THURSTON, W. P. 1992. *Word Processing in Groups*. Jones and Bartlett.
- GREENBERG, M. J. 1993. *Euclidean and Non-Euclidean Geometries: Development and History*, 3rd ed. W. H. Freeman and Company.
- GRÜNBAUM, B. 1984. The emperor's new clothes: Full regalia, G string, or nothing? *Math. Intel.* 6, 4, 47–53.
- GRÜNBAUM, B. AND SHEPHARD, G. C. 1987. *Tilings and Patterns*. W. H. Freeman.
- GRÜNBAUM, B. AND SHEPHARD, G. C. 1992. Interlace patterns in Islamic and Moorish art. *Leonardo* 25, 331–339.
- GUNN, C. 1993. Discrete groups and visualization of three-dimensional manifolds. In *Proceedings of SIGGRAPH'93*. ACM, New York, 255–262.
- HANKIN, E. H. 1925. *The Drawing of Geometric Patterns in Saracenic Art*. Memoirs of the Archaeological Society of India, vol. 15. Government of India.
- IRVING, W. 1931. *The Alhambra*. Macmillan and Co.
- KAPLAN, C. S. 2000a. Computer generated islamic star patterns. In *Bridges 2000 Proceedings*, R. Sarhangi, Ed.
- KAPLAN, C. S. 2000b. Taprats. <http://www.cgl.uwaterloo.ca/~csk/washington/taprats/>.
- KAPLAN, C. S. AND HART, G. W. 2001. Symmetrohedra: Polyhedra from symmetric placement of regular polygons. In *Bridges 2001 Proceedings*, R. Sarhangi, Ed.
- KAPLAN, C. S. AND SALESIN, D. H. 2000. Escherization. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 2000)*. ACM, New York, 499–510.
- KAY, D. C. 1969. *College Geometry*. Holt, Rinehart and Winston, Inc., New York.
- LEE, A. 1987. Islamic star patterns. *Muqarnas* 4, 182–197.
- ACM Transactions on Graphics, Vol. 23, No. 2, April 2004.

- LEVY, S. 1993. Automatic generation of hyperbolic tilings. In *The Visual Mind: Art and Mathematics*, M. Emmer, Ed. MIT Press, Cambridge, Mass., 165–170.
- LIPPMAN, S. B. AND LAJOIE, J. 1998. *C++ Primer*, 3rd ed. Addison-Wesley, Reading, Mass.
- MARTIN, G. E. 1975. *The Foundations of Geometry and the Non-Euclidean Plane*. Intext Educational Publishers.
- NECIPOĞLU, G. 1995. *The Topkapı Scroll—Geometry and Ornament in Islamic Architecture*. The Getty Center for the History of Art and the Humanities.
- OSTROMOUKHOV, V. 1998. Mathematical tools for computer-generated ornamental patterns. In *Electronic Publishing, Artistic Imaging and Digital Typography*. In Lecture Notes in Computer Science, vol. 1375. Springer-Verlag, New York, 193–223.
- PEDAGOGUERY SOFTWARE INC. 2000. Tess. <http://www.peda.com/tess/welcome.html>.
- SHUBNIKOV, A. V. AND KOPTSIK, V. A. 1974. *Symmetry in Science and Art*. Plenum Press, New York.
- STEWART, D. 1974. *The Alhambra*. Wonders of Man Series. Newsweek Book Division, New York.
- WADE, D. 1976. *Pattern in Islamic Art*. The Overlook Press.
- WASHBURN, D. K. AND CROWE, D. W. 1992. *Symmetries of Culture*. University of Washington Press.

Received March 2003; revised September 2003; accepted October 2003