Chapter 3

ISLAMIC STAR PATTERNS

3.1 Introduction

The rise and spread of Islamic culture from the seventh century onward has provided us with 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 chapter, I will focus on the last category: abstract geometric patterns. Specifically, I study the construction of *Islamic star patterns* such as the ones famously catalogued by Bourgoin [16]. These patterns adorn buildings, particularly mosques and tombs, 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 [88, 129].

Broadly speaking, an Islamic star pattern is a periodic arrangement of motifs, many of which are star-shaped. As with many other forms of ornament, it would be counterproductive to attempt a more rigorous definition. Instead, I work from the many published collections of star patterns [2, 16, 19, 27], letting them teach by example as Racinet suggested. The examples in these collections surround the space of relevant patterns with a fuzzy boundary, and in this chapter I will show how that boundary may be probed both mathematically and computationally.

There is some controversy in the question of why Islamic art tends so strongly towards geometric abstraction. Many European and American scholars assert that this tendancy is due to a strict Muslim prohibition on representation in art. One claim made is that representation is the sole dominion of Allah, who by one of His many names is known as the Giver of Form (Al-Mussawir). This dogmatic position can easily be refuted by the traditional Islamic arts of portraiture and miniature painting. A more credible point of view holds that in Islam, God is perceived as being so perfect, so pure, that no mere worldly image could hope to express His nature. The only appropriate means of religious exaltation lies then in art with mathematical, crystalline perfection [2].

There is a certain seductive element to the study of Islamic star patterns because little is known about how they were originally constructed. The design methods were the private domain of the artisans who practiced them. The knowledge was passed down from master to apprentice over generations and ultimately was lost as the practice of Islamic star patterns declined during the fifteenth century. Except for a few scattered remnants of this technical knowledge, such as the sixteenth century Topkapı scroll [116], the design of Islamic star patterns is a mystery. As a guide, we have only the end product: hundreds of beguiling geometric designs found all over the world.

One thing we know with certainty is that star patterns are deeply mathematical in nature. The most effective ones are little gems of geometry, conveying a kind of inevitability of design that belies the hard work originally required to discover them. The artisans who developed the patterns were well versed in geometry; in their pursuit of mathematical knowledge, early Islamic scholars translated Euclid's *Elements* into Arabic. And so even though we cannot peer back through time to understand their design techniques, we can at least be confident that their constructions were firmly rooted in geometry.

We should not expect a single construction technique to capture the structure of all star patterns. Broad families of patterns that seem to share a common structure are counterbalanced by remarkable one-of-a-kind artifacts no doubt conceived in a flash of inspiration. My goal is not universality, but usefulness; the construction techniques I develop in this chapter can express many common star patterns but make no claim at expressing them all.

Today, we have mathematical tools of a sophistication undreamed of by the Islamic scholars of a thousand years ago. These tools can be brought to bear on the analysis of star patterns and might even whisper geometric secrets that the inventors of those patterns were unaware of. Obviously, a technique based on modern mathematics is unlikely to bear much resemblance to the original methods. On the other hand, the goal here is not archaeological; any technique that can create a large variety of well-known patterns can be judged a success. Modern mathematics might even reveal degrees of freedom in pattern construction that were unavailable in the past because the tools to understand them had not yet been developed.

More recently, we have also experienced a revolution in manufacturing. We now have a variety of computer-controlled manufacturing systems that can build real-world artifacts from synthetic computer descriptions. These systems allow computer-generated star patterns to be built and deployed in the same architectural settings as their handmade historical counterparts.

In this chapter, I present my work on the construction and execution of Islamic star patterns. The central focus is the development of a tiling-based construction method that decomposes the design problem into two parts. First, a *template tiling* is given; it guides the large-scale layout of the final design. Then, *motifs* are constructed for each individual tile shape. When the template tiles are assembled into a tiling, the motifs join together to create a star pattern. I build this method in stages, culminating in a new technique that can produce star patterns in Euclidean and non-Euclidean geometry. I also show some ways that star patterns can be constructed based on aperiodic tilings.

3.2 Related work

Over the years, many mathematicians and art historians have focused their attention on the mystery of how star patterns were originally constructed. Many techniques have been proposed, and all are successful in various ways. The wide variety of successful techniques reflects the improbability that there was ever a single historical design method. More likely, the artisan's toolkit held an assortment of mathematical ideas.

Bourgoin created one of the first European collections of Islamic star patterns [16]. His book serves as a valuable set of examples for artists and mathematicians. Each pattern has a small section that appears to be inscribed with construction lines. One should not attempt to read too much into these lines. If anything, they are indications of Bourgoin's transcription process, guidelines he discovered while tackling each individual pattern. They do not provide any presciption of how to construct patterns in general.

Dewdney [36] presents a complete method for constructing designs based on reflecting lines off of a regular arrangement of circles. Although this technique could be used to construct many well-known designs, Dewdney admits that he requires many intuitive leaps to arrive at a finished design. Dispot's recent **Arabeske** software [37] allows the user to construct star patterns using an approach similar to Dewdney's.

In his book, Castéra [19] 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 eight-pointed stars and elongated hexagons called "safts" (a reference to the shuttle used in weaving), and fills the remaining space with additional shapes. With carefully chosen skeletons, he is able to create designs of astonishing beauty and complexity. Castéra imposes no *a priori* restrictions on a design's symmetries, though by the nature of his construction technique he tends to obtain designs with global eightfold symmetry. Castéra's designs reflect the Moroccan aesthetic of complex patterns centered around a single large star, and not the large body of periodic star patterns that I will address here.

The idea of using a tiling as a guide to the construction of star patterns is a common thread that ties together the investigations of many scholars. Evidence of such a tiling-based (or at least tiling-aware) construction can be found in the centuries-old Topkapı scroll [116]. In 1925, E. H. Hankin [70] wrote of his discovery of a Turkish bath where the star patterns on the walls were accompanied by a lightly-drawn polygonal tiling. Wade [134] 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 [99] mentions the "polygons in contact technique," stating that new star patterns might be constructed by searching for polygonal tesselations.

Jay Bonner, an architectural ornamentalist in New Mexico, has devoted considerable time and energy to the classification and generation of Islamic star patterns. In an unpublished manuscript [13], he details his techniques for producing star patterns, using a tiling-based construction technique very much like the one presented in Section 3.4. He cites several pieces of evidence suggesting that this approach was the predominant means by which star patterns were originally conceived. Section 3.8 discusses ways that his approach differs from the one presented in this chapter.

3.3 The anatomy of star patterns

Artistic and architectural renderings of Islamic star patterns are richly decorated, often made up of coloured regions bounded by interlaced strands. Many of these lush renderings are expertly reproduced in a nineteenth century collection by Prisse d'Avennes [31]. To understand them mathematically, we must factor these artifacts into two pieces: an underlying geometric component, and a decoration style that has been applied to it. The two pieces should be orthogonal, allowing one to mix and match geometry with style.

Grünbaum and Shephard explain how one may extract from a rendered pattern the essential geometric content, which they call the *design* [69]. We discard all colour information and reduce thickened bands to lines. Where there is interlacing, we ignore it; as they point out, in a design that admits interlacing the pattern of crossings is uniquely determined by the underlying geometry, up to a global exchange of over and under. For designs derived from real-world artifacts, we must also imagine this finite piece of geometry extended across the entire plane in a natural way. For periodic designs, it is usually obvious how to carry out this extension. When precision is called for, I follow the nomenclature of Grünbaum and Shephard, referring to a finished, decorated work as a "pattern," based on an underlying "design."

A design is a collection of line segments in the plane that do not intersect each other except possibly at their endpoints. We may regard such a collection as an infinite planar map. The map consists of a set of vertices, each of which has a position in the plane, and a set of edges that connect pairs of vertices. The *valence* of a vertex is the number of edges that have it as an endpoint. In the periodic case, this planar map can be conveniently represented by its restriction to a single translational unit.

In what follows, I restrict my attention to the class of designs with the following two properties:

1. Every vertex has valence two or four.

2. The valence four vertices are *perfect crossings*. That is, the four edges that meet at the vertex can be interpreted as two line segments that intersect at the vertex.

The first property allows us to view the design as a well-defined collection of *strands*, paths through the design that ultimately form closed loops or extend to infinity. A vertex of degree two is a bend in a strand; a vertex of degree four is a place where two strands cross.

This first property is also sufficient to ensure that a design admits an interlacing. At every crossing, one strand can be chosen to pass over the other. When every vertex has degree two or four, this assignment can be carried out globally in such a way that as one follows a given strand, it passes alternately over and under any strands it crosses. When the design is a connected graph, the assignment can be made by choosing the over/under relationship at a single vertex, and propagating the assignment throughout the planar map. The single boolean choice at the designated vertex leads to two self-consistent, opposite assignments (when the map has multiple components, the choice must be made once per component). As an aside, the redundancy implied by the equivalence of these assignments can be formalized through the inclusion of a new symmetry operation, one that leaves the plane fixed but exchanges the roles of over and under everywhere. Mixing this reweaving operation in with the regular isometries produces an enrichment of the usual symmetry groups that can provide a finer classification of interlaced figures (see, for example, Cromwell [29] and Shubnikov and Koptsik [127, Chapters 5 & 8]).

The second property is motivated more from aesthetic considerations than mathematical constraints. The crossing at a degree four vertex is most clearly rendered when the two strands that meet there pass right through the intersection without changing course. There is evidence in the literature on Gestalt psychology to support the aesthetic superiority of perfect crossings.

The space of star patterns that satisfy these two conditions is still very large. A quick survey shows that approximately 70% of the designs reproduced by Bourgoin are admissible. And as I will demonstrate, within this framework there is a vast space of designs to be explored.

3.4 Hankin's method

I begin the process of developing a tiling-based star pattern construction method by discussing the creation of motifs. Later, I will consider the tilings themselves.

The tiling-based approach seems to have first been articulated in the west by Hankin in the early part of the twentieth century. In a series of papers [70, 71, 72, 73], he explains his discoveries and gives many examples of how the technique can be used. Hankin's description of his technique provides an excellent starting point for an algorithmic approach.

> In making such patterns, it is first necessary to cover the surface to be decorated with a network consisting of polygons in contact. Then through the centre of each side of each polygon two lines are drawn. These lines cross each other like a letter X and

> are continued till they meet other lines of similar origin. This completes the pattern [70, Page 4].

Hankin's description immediately suggests an algorithm based on "growing" edges out of every tiling edge midpoint, and cutting those edges off where they intersect each other. His description of a letter X at each edge midpoint precisely fits the requirement of perfect crossings. The places where edges meet other edges and are cut off complete the design with bivalent vertices.

Given a tiling, Hankin's recipe has one remaining degree of freedom: the angle formed by the growing motif edges with the tiling edges they emanate from. I call this the *contact angle* (see Figure 3.1). A software implementation of these growing edges should accept a tile and a contact angle as input, and attempt to grow a motif for that tile using the given contact angle. When this operation is run on every tile in a tiling, a star pattern emerges.

I have developed an interactive Java program that implements what I call "Hankin's Method." At its core is an "inference algorithm": a subroutine that takes a polygonal tile and a contact angle as input and returns a motif for that tile. The inference algorithm runs quickly enough that the contact angle can be varied in real time, with the changes to the resulting design displayed interactively.

In his manuscript, Bonner gives a systematic presentation of this technique over a vast space of tilings (which he calls "polygonal sub-grids"). Although I became aware of the tiling-based approach from Hankin's writing, it is through Bonner's work that I came to appreciate its importance as a step leading up to the work in the rest of this chapter. Bonner's library of results shows just how large a space of star patterns may be obtained using just Hankin's Method.

Bonner's work is intended to provide a resource for designers, and not an algorithm for software writers. Therefore, he does not present a formal algorithm for inferring motifs in template tiles. Besides its use in the automatic generation of star patterns, the inference algorithm presented here could ultimately help designers as well, by guiding their intuition in the construction of motifs.

Hankin's method begins by identifying those places from which pieces of the design will originate. Given a tiling by polygons, we define the *contact points* of a tile to be the set of midpoints of its edges. When the tiles are assembled into a tiling, neighbouring tiles will often have coincident contact points. It is from these shared contacts that an X-shaped arrangement of edges will grow.

Consider a single tile, and let θ be the desired contact angle. To each contact point p we can

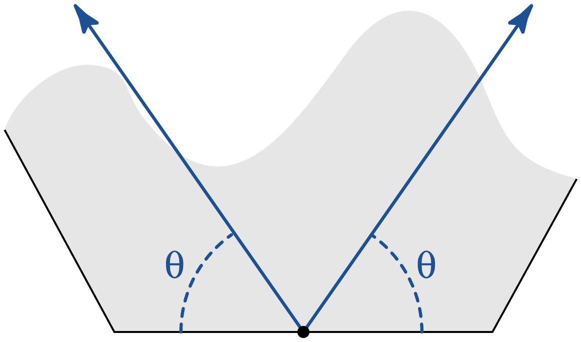


Figure 3.1 In the first step of Hankin's method, a pair of rays is associated with every contact position on every tile. Here, a single contact position gets its two rays, each of which forms the contact angle θ with the contact edge.

associate a pair of rays starting at p and making an angle of θ with the edge containing p. These rays are illustrated in Figure 3.1.

To create a final motif, we will need to truncate every one of these rays somewhere along its length. Because the goal is to create unbroken strands, every ray will have to be truncated where it meets some other ray, either creating a bend, or occasionally an unbroken straight line segment. If every ray meets up with some other ray, we have a complete motif. Based on this description, we can specify a motif by giving a pairing of the rays, a set of unordered pairs of rays in which each ray appears exactly once. The ultimate goal of any inference algorithm is then to choose from among all possible pairings for the one that best satisfies some sort of aesthetic goal.

As an aesthetic goal, I choose the minimization of the cost of drawing the motif, measured as the total length of all the line segments that make it up. This goal makes some amount of intuitive sense. As was mentioned earlier, in Islamic star patterns we find an inevitability of design, a sense in which the design's geometry is expressed with the greatest possible economy. This economy can be seen to arise by choosing the simplest completion of a motif that fits the global design rules. Furthermore, the principle of minimizing drawing cost is borne out by many well-known examples. By developing Hankin-like tilings for historical designs, motifs can in fact be seen to take the simplest route to completion. An approach based on simplicity would seem to be predicated upon a certain amount of intelligence in the choice of tilings. This issue will be raised again later in Section 3.8.

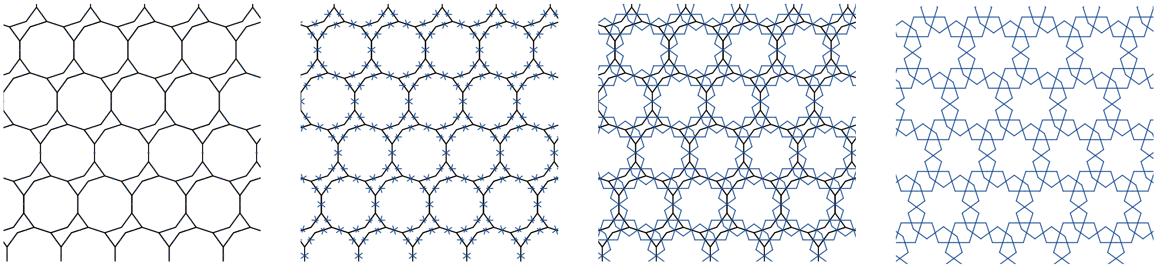


Figure 3.2 A demonstration of Hankin's method. The frame on the left shows the original tiling. Rays are grown out of every contact position, and continue until they meet other rays in a manner dictated by the inference algorithm. When the original tiling is removed, the result is the Islamic star pattern on the right.

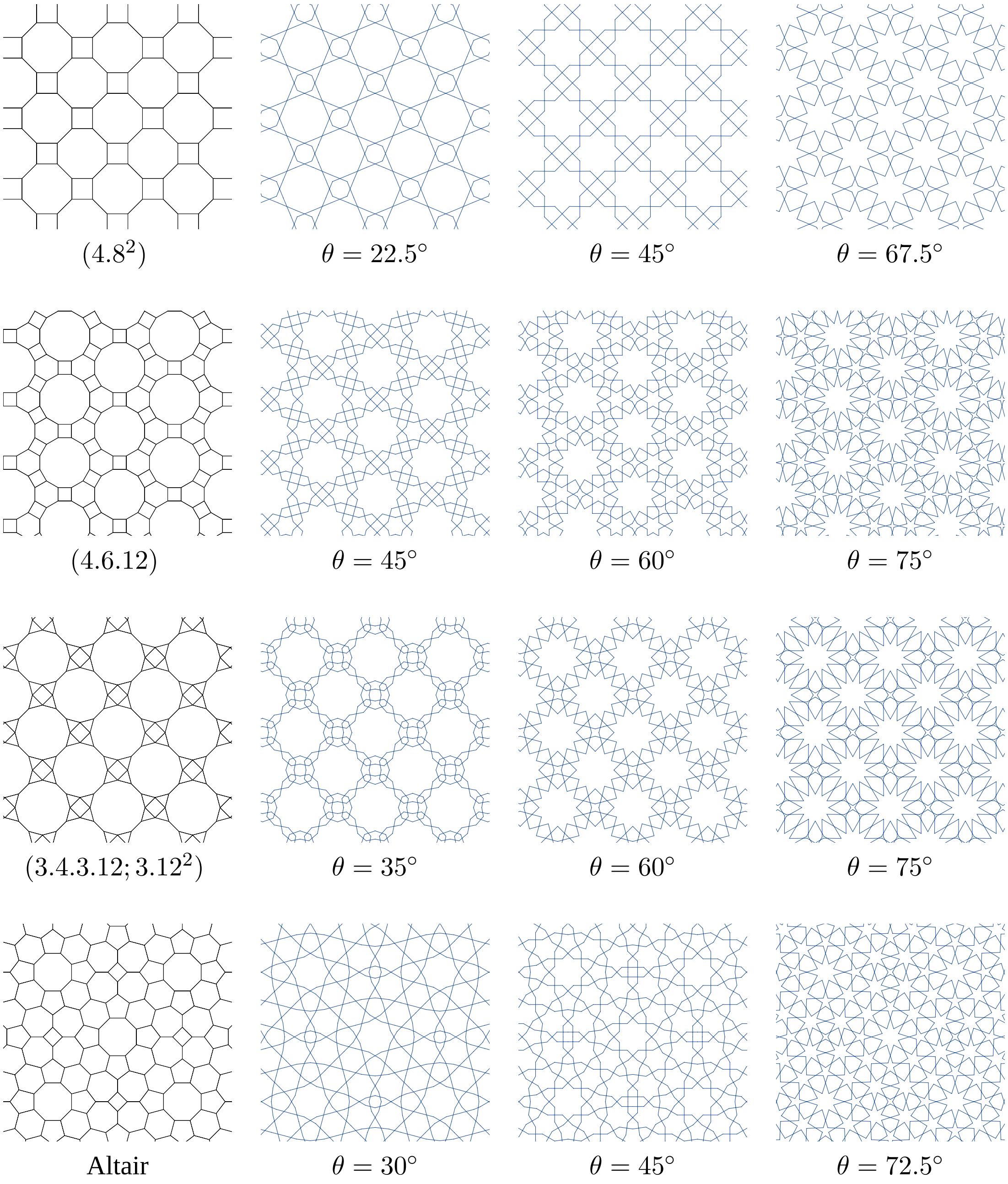
As a first attempt at inference, we might then proceed as follows. Iterate over all possible pairings, evaluating the total cost of each. From among all pairings that use the most possible rays, choose the one with the lowest total cost. Unfortunately, this algorithm is not practical. In an n-sided tile, there will be $2n$ rays. The number of pairings between rays is the number of ways to partition the $2n$ rays into disjoint pairs. This number works out to be $\frac{n!}{(n/2)!2^n} = 1 \cdot 3 \cdot 5 \cdot 7 \ldots (n-1)$. We would need to evaluate more than half a billion possibilities for a region with 10 sides.

A simpler approach is to use a greedy algorithm. Given a list of rays, we first build a list of all pairs of rays that intersect, sorted in order of increasing cost. We then traverse the list, incorporating the segments induced by every pair in order provided both of the rays that make up that pair are still unused.

In practice, this algorithm performs very well on many of the typical tile shapes that arise in template tilings. It certainly performs perfectly on regular polygons, where it constructs star-shaped motifs. In the cases where it fails, it usually does so not because it is greedy, but because the underlying idea of using edge midpoints and a single contact angle is overly simplistic. Bonner demonstrates some cases of template tilings for which contact positions must be moved slightly off-center and contact angles changed. These special cases are discussed in greater detail in Section 3.8.

Figure 3.2 illustrates the process of growing rays from contact positions. Figure 3.3 shows some typical designs that can result from using my implementation of Hankin's method.

There are some cases where simple modifications to the basic inference algorithm can improve



(4.8^2) $\theta = 22.5°$ $\theta = 45°$ $\theta = 67.5°$

$(4.6.12)$ $\theta = 45°$ $\theta = 60°$ $\theta = 75°$

$(3.4.3.12; 3.12^2)$ $\theta = 35°$ $\theta = 60°$ $\theta = 75°$

Altair $\theta = 30°$ $\theta = 45°$ $\theta = 72.5°$

Figure 3.3 Examples of star patterns constructed using Hankin's method. Each row shows a template tiling together with three designs that can be derived from it using three different contact angles. The bottom row features an amusing tiling by nearly regular polygons. It is reproduced from Grünbaum and Shephard [68, Page 64], where it serves as a reminder of the danger of over-reliance on figures. A related design also appears in Bourgoin [16, Plate 163].

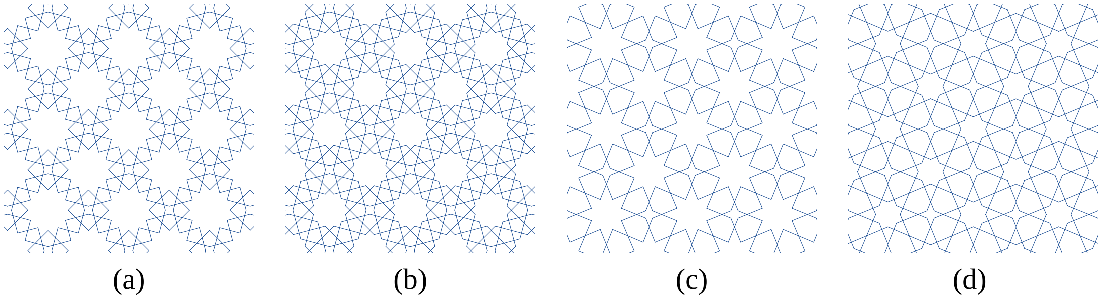


Figure 3.4 A demonstration of two cases where an extension to the inference algorithm can produce a slightly more attractive motif. In (a), a star pattern is shown with large unfilled areas that were the centers of regular dodecagons in the template tiling. Adding an additional layer of inferred geometry to the inside of the motif produces the improved design in (b). The process is repeated with a different template tiling in (c) and (d).

the generated motif. Consider, for example, the star pattern given in Figure 3.4(a). This pattern contains large regions, derived from regular dodecagons, that are left unfilled. A more attractive motif can be constructed using a second pass of the inference algorithm, building inward from the points where the rays from the first pass meet. The resulting design, shown in Figure 3.4(b), is more consistent with tradition. In the inference algorithm, it is easy to recognize when the provided tile shape is a regular polygon and to run the second round of inference when desired. An alternative solution to this problem, based on an explicit parameterization of star shapes, is given in the next section.

A further enhancement is to allow the contact position to split in two, as shown in Figure 3.5. This splitting can be accomplished quite naturally by providing the inference algorithm with a second real-valued parameter δ that specifies the distance between the new starting points of the rays. The parameter δ can be allowed to vary from zero (giving the original construction) up to the length of the shortest tile edge in the tiling. This construction gives what Bonner calls “two-point patterns,” a set of designs that are historically important in Islamic art. Examples of two-point patterns constructed using the δ parameter are shown in Figure 3.6. The designs corresponding to two-point patterns tend to be made up of very short closed strands, each one forming a loop around a single tiling vertex in the template tiling.

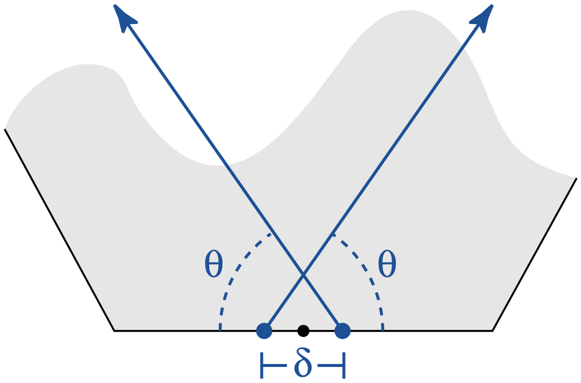


Figure 3.5 A visualization of how δ is incorporated into Hankin's method. The contact position is split in two, resulting in two rays whose starting points are separated by distance δ. Motifs will still line up provided the same δ is used throughout the design.

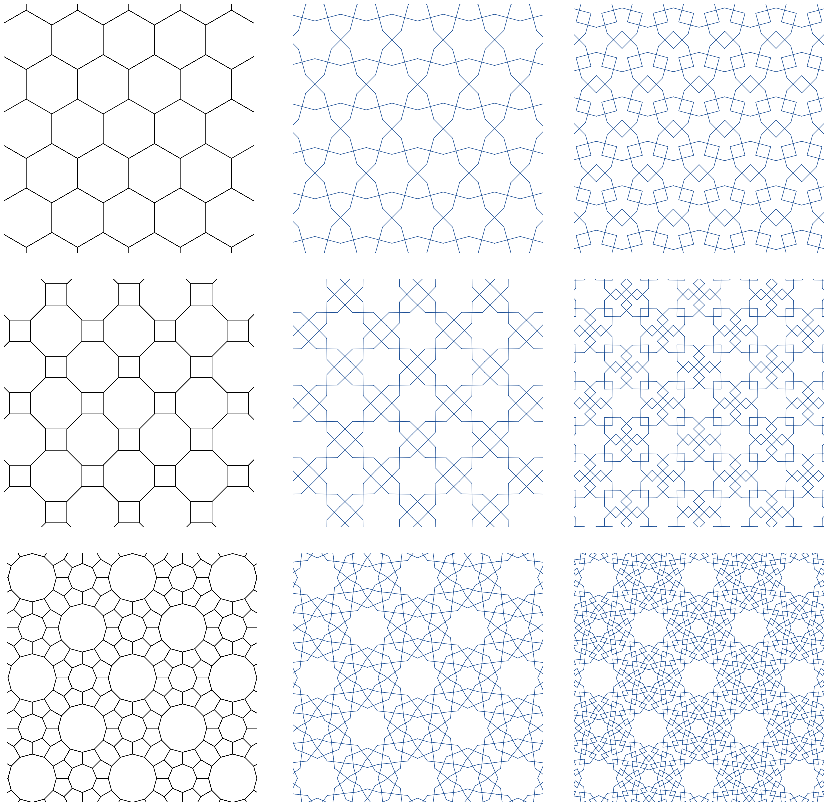


Figure 3.6 Examples of two-point star patterns constructed using Hankin's method. Each row shows a template tiling, a star pattern with $\delta = 0$, and a related two-point pattern with non-zero δ. The structure of the tiling in the bottom row will be explained in Section 3.8.

The success of a tiling-based approach in the construction of star patterns does not seem so surprising given the foregoing discussion. When applied to a regular polygon, the inference algorithm presented will always produce a star. When many regular polygons are assembled into a tiling, the result is a pattern containing many stars. This simple process can therefore be applied to a wide variety of well-known tilings to produce designs that are recognizably in the tradition of star patterns. On the other hand, the first of the two extensions mentioned above suggests that for tiles that are regular polygons, we might recognize the regularity and choose to insert a motif richer than that provided by the inference algorithm alone. This observation is the basis for the system presented in Section 3.5.

3.4.1 Islamic parquet deformations

Parquet deformations are a style of ornamental design created by William Huff, a professor of architectural design, and later popularized by Hofstadter in *Scientific American* [83, Chapter 10]. They are a kind of “spatial animation,” a geometric drawing that makes a smooth transition in space rather than time. Parquet deformations are certainly closely related to Escher’s *Metamorphosis* prints, though unlike Escher’s work they are purely abstract, geometric compositions. They will be discussed in more detail in Section 5.4.

Hankin’s method can be used as the basis for a simple but highly effective method of constructing Islamic patterns in the spirit of parquet deformations. I lay out a strip of the template tiling and then run a modified inference algorithm where the contact angle at every contact point is determined by the location of that point in the strip. Smoothly varying the contact angle results in a gently changing geometric design that is still recognizably Islamic. The construction process is illustrated in Figure 3.7; two more examples appear in Figure 3.8. These parquet deformations occupy an interesting place in the world of Islamic geometric art. They have enough overall structure and balance to satisfy the Islamic aesthetic, but they would not have been produced historically because very little repetition is involved. The effort of working out the constantly changing shapes by hand would have tested the patience of any ancient designer.

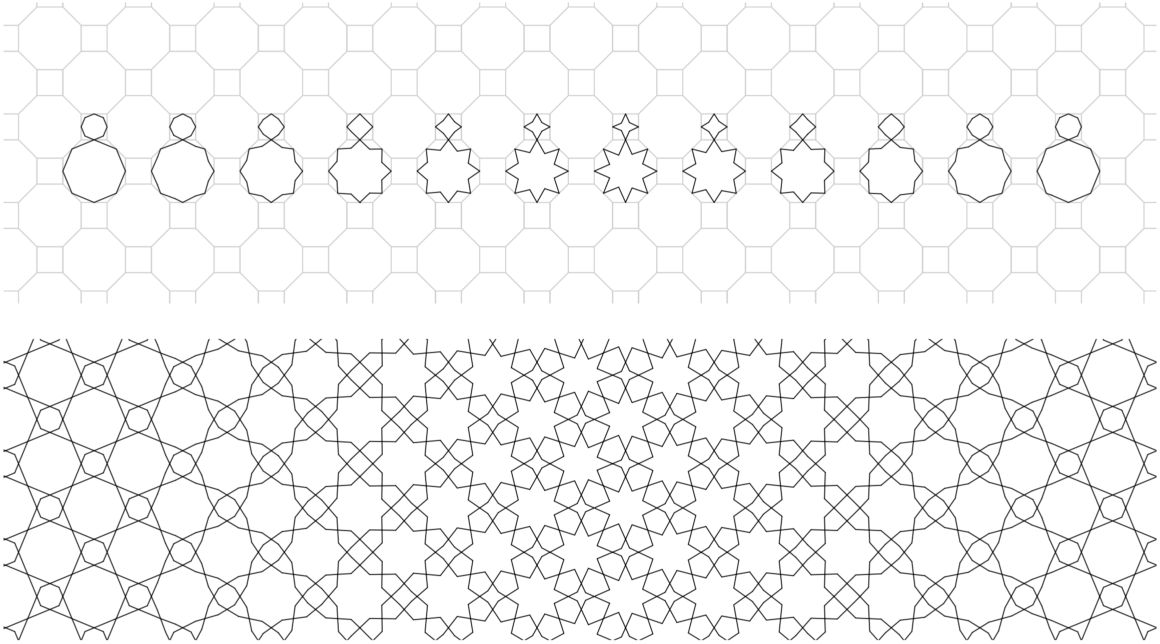


Figure 3.7 The construction of an Islamic parquet deformation based on Hankin's method. The top rows shows the effect of continuously varying the contact angle of a ray depending on the x position of the ray's starting point in the design. When the process is carried to all other tiles, the design in the second row emerges.

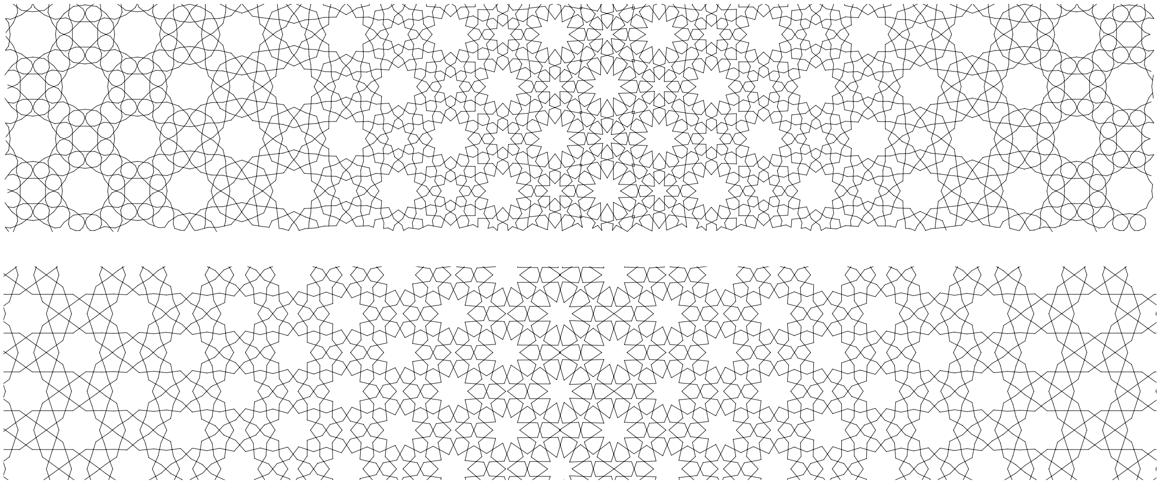


Figure 3.8 More examples of Islamic parquet deformation based on Hankin's method.

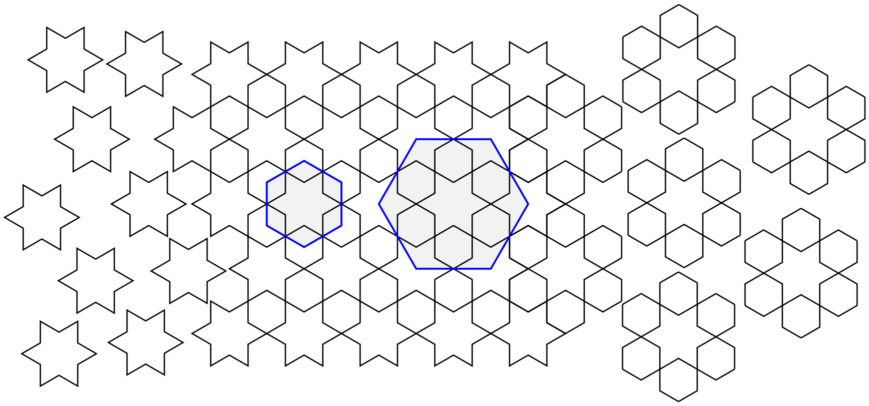


Figure 3.9 The discovery of a complex symmetric motif in a star pattern. The original design was constructed by placing six-pointed stars in a regular tiling by copies of the smaller hexagon. Around each star, we find a larger d_6 motif by adjoining the six neighbouring regular hexagonal regions. An example of the larger motif, called a *rosette*, is outlined by the larger blue hexagon. The entire design can be seen as constructed from copies of the rosette.

3.5 Design elements and the Taprats method

Hankin's method, described in the previous section, can produce a wide range of well-known and unknown Islamic star patterns. The method is based on a kind of Occam's razor of aesthetics: in deciding on a motif for a tile shape, the simplest possible motif is the best choice. When the tile is a regular polygon, that simplest motif will be a star shape.

Yet in surveying the many historical examples of star patterns we can see the repeated occurence of radially symmetric motifs more complex than mere stars. We already encountered one example in Figure 3.4, where many-pointed stars are given an additional inner layer of geometry to help fill the large areas of otherwise empty space. It might be possible to give a contrived tiling that accounts for these more complex motifs, but ultimately we are better off hypothesizing that the original designers of star patterns understood higher-level motifs directly.

Once we make this hypothesis, many other recurrent motifs assert their identities as first-class objects. A classic example is given in Figure 3.9. Here, a pattern composed of six-pointed stars with hexagonal holes can be reinterpreted by adjoining to a star its six neighbouring hexagons. Copies of this *rosette* fit together to recreate the original pattern, now leaving behind stars as holes. It is reasonable to assume that these rosettes were well understood in their own right, and not merely as an incidental by-product of constructing star patterns.

I build upon Hankin's method by giving a general theory of radially symmetric motifs (motifs with symmetry group d_n, in the language of Section 2.2) that can be associated with regular polygons. I then use this theory to build a set of *design elements*, parameterized families of historically important motifs. The design elements represent our opportunity to interpret most explicitly 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.

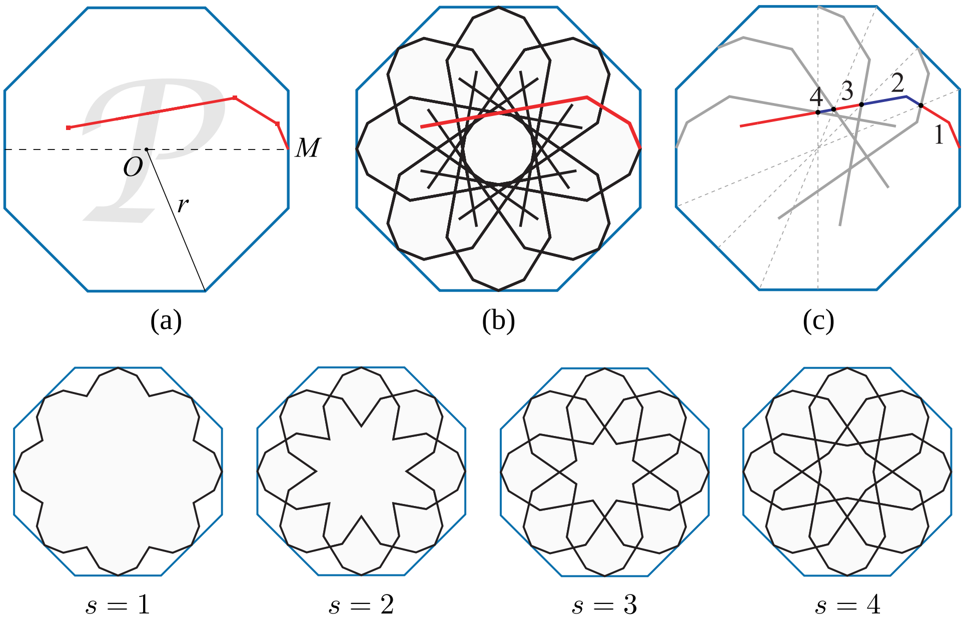
The design elements can take the place of the default inference algorithm in the construction of motifs for regular polygons. I combine the design elements and the inference algorithm in an interactive tool called **Taprats** that has been used successfully to design, render, and execute many traditional and novel star patterns.

3.5.1 Path-based construction of design elements

Let $\mathcal{P}$ be a regular n-sided polygon with center O, inscribed in a circle of radius r. Designate the midpoint of one of the polygon's sides by M, and for convenience orient the polygon so that M lies due east of O. A design element can then be represented as as a piecewise-linear path that starts at M and wanders around inside $\mathcal{P}$. A d_n-symmetric motif can then be constructed by combining all images of the path under the symmetries of the surrounding polygon. The process of turning a path into a symmetric motif is illustrated in Figure 3.10.

During this replication process, the original path will intersect rotated and reflected copies of itself. The intersections occur on successive lines of reflection of $\mathcal{P}$. As shown in Figure 3.10, the integer parameter $0 < s \leq n/2$ controls how many of these subpaths to keep. The number s turns a single path into a family of related motifs. It is generalized from its standard use in describing star polygons [99].

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. Given n and r as above, such a model produces a path inside $\mathcal{P}$, starting at M. The replication process above, combined with the



(a) (b) (c)

$s = 1$ $s = 2$ $s = 3$ $s = 4$

Figure 3.10 The path-based construction of Section 3.5.1 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.

s parameter, can then be applied to the path to obtain a planar map representing the finished motif. Every design element will include n, r, and s as variables. Naturally, the procedural model may include its own additional degrees of freedom.

I have created parameterized design elements for stars and rosettes, which together capture the majority of symmetric motifs found in star patterns. I have also developed a generic "extension" mechanism that wraps an additional layer of geometry around any other motif. These models know only how to construct a piecewise-linear path starting at M. All models share a common implemenation of the algorithm above that turns a path and a value of s into a radially symmetric planar map.

3.5.2 Stars

At the very heart of Islamic star patterns we find the star. Islamic art features stars with many different numbers of points, up to a remarkable 96 [19, Page 220]. We have already seen how a star can arise naturally by running the inference algorithm on a regular polygon. It can also be useful to express stars as higher-level design elements, in order to exert more direct control over them.

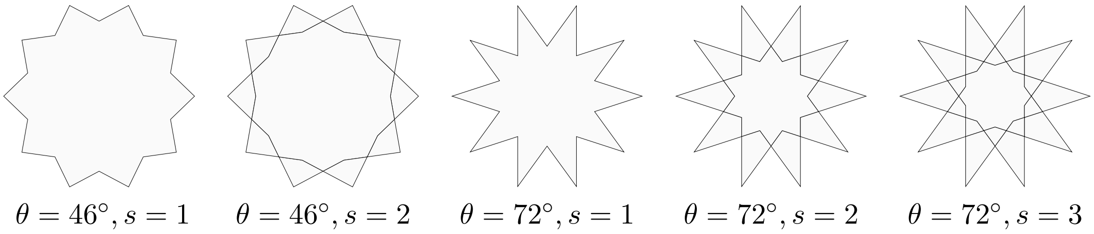
Grünbaum and Shephard [68, Section 2.5] show how star polygons may be specified using the notation $\{n/d\}$, where n and d are integers, $n \geq 3$, and $1 \leq d < \lceil n/2 \rceil$. The star is constructed by placing points $v_1, \ldots, v_n$ at the vertices of a regular n-gon, and joining every v_i to v_{i+d}, where indices are taken modulo n. For the purposes of ornamental design, Lee [99] adds an s parameter equivalent to the one presented above, arriving at the final description $\{n/d\}s$.

Instead of relying on $\{n/d\}s$ notation, it is more convenient to parameterize stars by giving the contact angle θ directly. My implementation of Hankin's method is based fundamentally on the user's choice of contact angle, and so making this angle the basis of design elements allows for smoother integration with the construction technique already given.

In Taprats, 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 has length $2r$. It is parameterized by a single degree of freedom, the contact angle θ. Some examples of stars constructed this way are given in Figure 3.11.

The parameterization, based on the tuple (n, r, s, θ), generalizes the original star notation in a straightforward way. Given a star $\{n/d\}s$ and a radius r, a little trigonometry shows that this star can be reparameterized as $(n, r, s, \pi d/n)$. As an extension, the angle θ can take on any real value in the range $(0, \pi/2)$ (as Lee mentions [99], a similar extension can be carried out on the original notation by permitting non-integral values of d).

For a given regular n-gon of radius r and a contact angle θ, we can now see that the inference algorithm used to implement Hankin's method will produce $(n, r, 1, \theta)$. Moreover, the special case illustrated in Figure 3.4, where a star receives an extra internal layer of geometry, is simply $(n, r, 2, \theta)$. In general, an appropriate value of s can usually be decided for stars automatically from n and θ: typically, $s = 2$ when $n > 6$ and $\theta > 2\pi/n$, and $s = 1$ otherwise.



$\theta = 46^\circ, s = 1$ $\theta = 46^\circ, s = 2$ $\theta = 72^\circ, s = 1$ $\theta = 72^\circ, s = 2$ $\theta = 72^\circ, s = 3$

Figure 3.11 Examples of stars constructed using the technique of Section 3.5.2. Each example is of the form $(10, 1, s, \theta)$ for varying choices of θ and s.

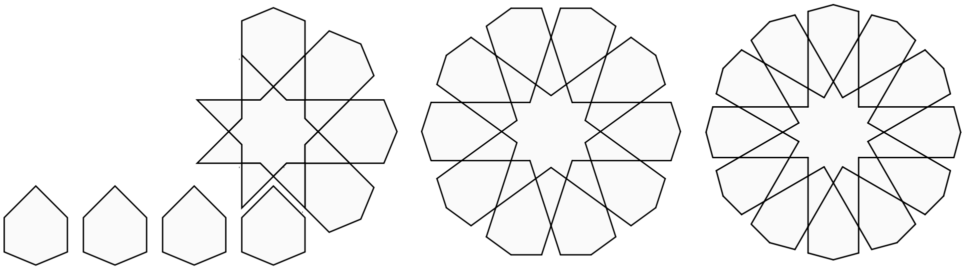


Figure 3.12 Examples of eight-, ten-, and twelve-pointed rosettes. The eight-pointed rosette on the left is partially decomposed into an internal $\{8/3\}2$ star and the eight surrounding hexagons.

3.5.3 Rosettes

The rosette is one of the most characteristic motifs of Islamic art. We may hypothesize that rosettes were first observed in the design of Figure 3.9, and that through experimentation they were gradually adapted to more general contexts. A rosette may be viewed as a star to which hexagons have been attached in the concavities between adjacent points (see Figure 3.12). 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. Referring to the labeling shown in Figure 3.13, the first segment, MG, becomes part of the outer edge of a hexagon. The path bends at G, what Lee calls the “shoulder,” and continues in a segment through C that becomes the hexagon’s flank and forms the inner star. Because the second segment effectively acts as a ray, its length is irrelevant, and the four degrees of freedom implied by the path are reduced to three. The problem then is to encode these degrees of freedom in a way that makes it easy to express rosettes with meaningful, intuitive properties. As usual, we choose the contact angle as a first parameter. To derive two more parameters, we first must understand what an "ideal" rosette might look like, and then provide as parameters deviations from this ideal.

Lee [99] provides an ideal construction, demonstrated in Figure 3.13. Given the surrounding polygon, point C is found as the point on $\overline{OA}$ with $AC = AM$, and G is found as the intersection of $\overline{MM'}$ with the line through C parallel to $\overleftrightarrow{OM}$. The result is a motif where $GC = GM$ and $\angle ACG = \angle AMG$.

Before generalizing Lee's construction, let α be the "natural" height of point G in the ideal rosette, as a fraction of the surrounding polygon's side length. Referring to Figure 3.14(a), let point H be the projection of G onto segment $\overline{OM}$. Then $\alpha = GH/AM$. The value α provides a scale-independent measure of the default height of G.

The three degrees of freedom in the generalized rosette model are now encoded as deviations from the ideal path implied by Lee's construction. As shown in Figure 3.14(b), a rosette is specified via six parameters: $(n, r, s, \theta, h, \phi)$. Parameters n, r, and s are as given for all path-based motifs. The shoulder G is defined as point with height $h\alpha$ above line OM such that $|\angle GMA| = \theta$. The second point in the path is then chosen so that the resulting ray leaving G forms an angle ϕ with the horizontal. Observe that as usual, θ is chosen to encode the desired contact angle, which allows for easy integration with Hankin's Method. Also note that under this parameterization, Lee's ideal rosette can be recovered as $(n, r, s, \pi/n, 1, 0)$. Figure 3.15 shows some examples of the effect of varying θ, h, and ϕ.

Every rosette contains a central star. As was mentioned, in constructing stars there is usually a natural value for s, determined from n and θ. The same is true here: the choice of s can be obtained by determining the correct value for the central star and adding 1 to account for the additional geometry of the surrounding hexagons. Accordingly, most large rosettes will have $s = 3$.

3.5.4 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 3.16, forming a larger

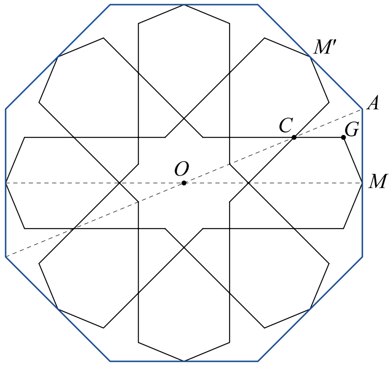


Figure 3.13 A diagram used to explain the construction of Lee's ideal rosette [99]. The construction is explained in Section 3.5.3.

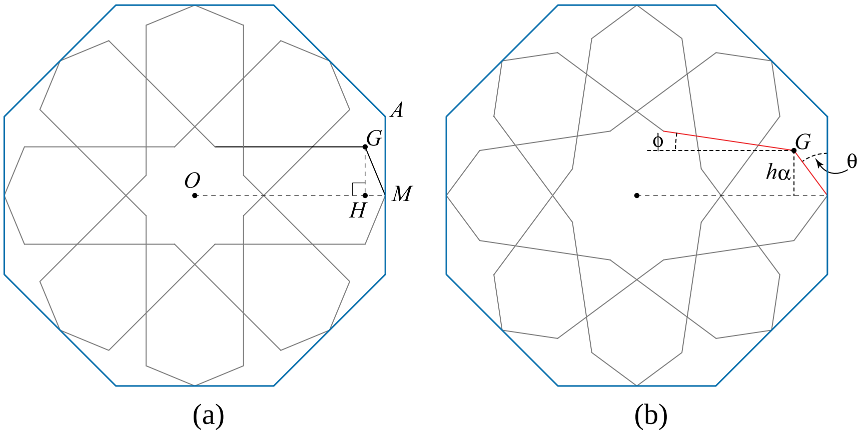
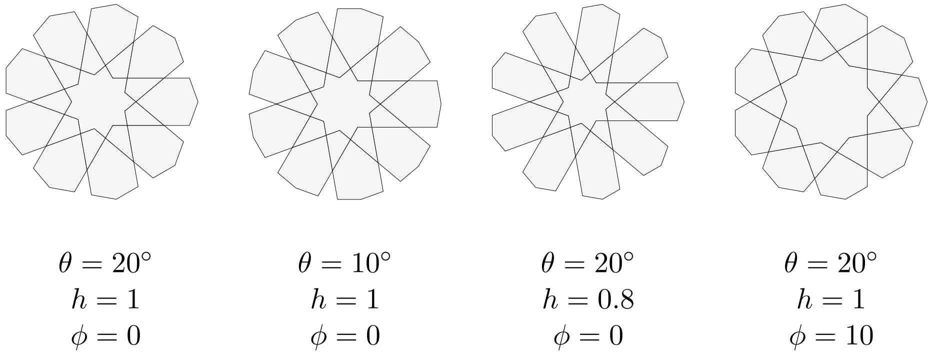


Figure 3.14 Two diagrams used to explain the construction of generalized rosettes as presented in Section 3.5.3.

motif with d_n symmetry. We refer to this process as *extension*. It is important to offer an extension facility, as "extended rosettes" figure prominently in many historical examples.

The procedural model for extension takes as input any other procedural model that includes the contact angle θ as a parameter and constructs directly an extended version of that model's elements



$\theta = 20°$ $h = 1$ $\phi = 0$

$\theta = 10°$ $h = 1$ $\phi = 0$

$\theta = 20°$ $h = 0.8$ $\phi = 0$

$\theta = 20°$ $h = 1$ $\phi = 10$

Figure 3.15 Examples of rosettes constructed using the technique of Section 3.5.3. Each example has $n = 9$, $r = 1$, and $s = 3$, and shows the effect of varying θ, h, and ϕ.

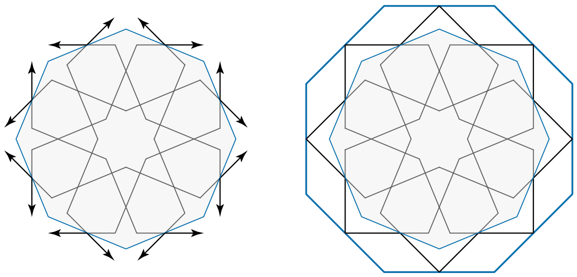


Figure 3.16 The extension process for design elements. The contact edges of the inner element are extended until they meet to become the contacts of the outer element.

inside a given polygon.[1] Given n, r, s, and θ, a little trigonometry shows that if we take

$$\theta' = \frac{\theta}{2}, \qquad r' = r\left[\frac{1}{\cos\frac{\pi}{n}} - \frac{\sin\theta\tan\frac{\pi}{n}}{\cos\left(\frac{\pi}{n} - \theta\right)}\right]$$

then an inner motif with contact angle θ' inscribed in a polygon of radius r' can be extended into a motif that fits perfectly in the outer n-gon. The child model is passed n, r', $s - 1$, and θ', along with unchanged values for any remaining parameters. The resulting motif must be rotated by π/n about its center in order to bring the motif into alignment with the outer polygon's contact points.

[1] Extension might be considered a higher-order model, *i.e.*, a function from models to models.

3.5.5 Taprats

I have taken the tiling-based approach of Hankin's method, the inference algorithm presented in the previous section, and the parameterized design elements, and combined them in a single Java-based program called **Taprats**. Taprats provides an interactive interface for creating, editing, decorating, and rendering star patterns. The system can operate on any periodic tiling, and has a number of common Islamic template tilings built in, derived from experimentation and examination of historical sources. Taprats has been available on the internet since 2000 as an applet, and more recently as a downloadable application. Both versions have received a great deal of positive feedback and interest from computer scientists, artists, architects, and educators.

Given a periodic tiling, Taprats presents a user inferface featuring an editor for the motifs in each tile shape. For regular polygons, that editor allows the user to choose from among stars, rosettes, and extended rosettes, and to edit the degrees of freedom in the design elements directly. For the remaining irregular tile shapes, the system runs the inference algorithm discussed in the previous section. It also allows the user to edit inferred motifs directly in case the inference algorithm produces an unacceptable result. The line segments that make up the motifs are then assembled into planar maps that are joined to cover a region of the plane. The final map can be decorated in various ways (a discussion of decorating star patterns will be postponed until Section 3.7) or exported to a number of manufacturing processes (see Section 3.9). A visualization of how Taprats assembles a star pattern is shown in Figure 3.17. Some other examples of patterns generated by Taprats appear in Figure 3.18.

3.6 Template tilings and absolute geometry

In the previous sections, I have developed a set of tools that can be used to construct a wide range of Islamic star patterns. However, one aspect of the process remains unexplained. Each design is ultimately based on a template tiling, and nothing has been said so far about how these tilings might be specified. In theory, we could simply stop here, and rely on a large body of hand-coded tilings derived from collected examples and the work of researchers like Bonner. This approach was taken in writing Taprats, with acceptable results. But stopping here seems unsatisfying since there is then no rhyme or reason to the set of available tilings, no abstraction that encompasses them. In effect,

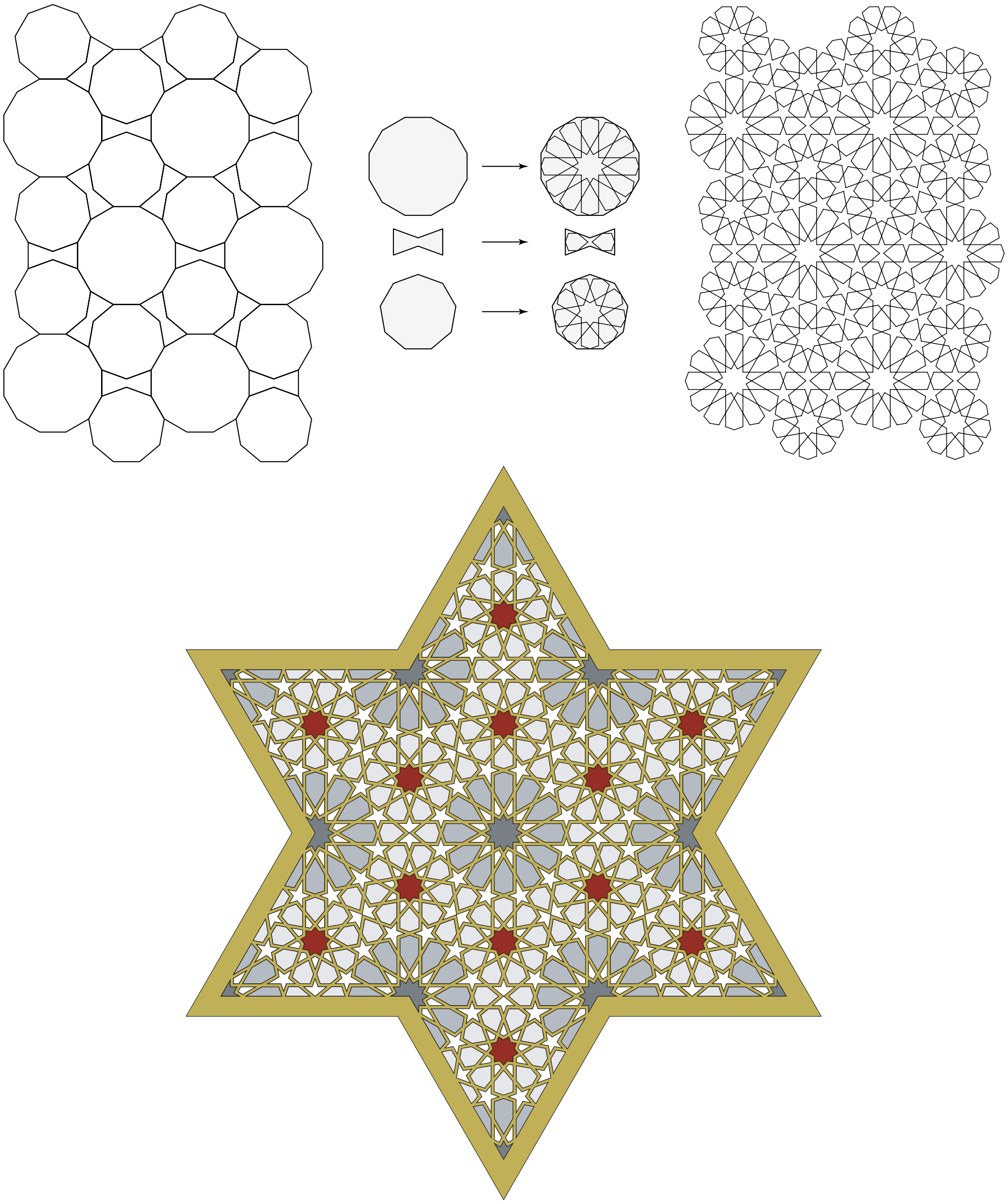


Figure 3.17 A visualization of how Taprats assembles a star pattern. On the top left, a tiling is selected. Each different tile shape is assigned a motif. In this case the dodecagons and enneagons receive rosettes; the bowtie's motif is inferred. When the motifs are copied into the tiles and the tiling is erased, the design on the upper right is revealed. That design can then be decorated and turned into a final rendering as shown below. The decoration process is described in Section 3.7.

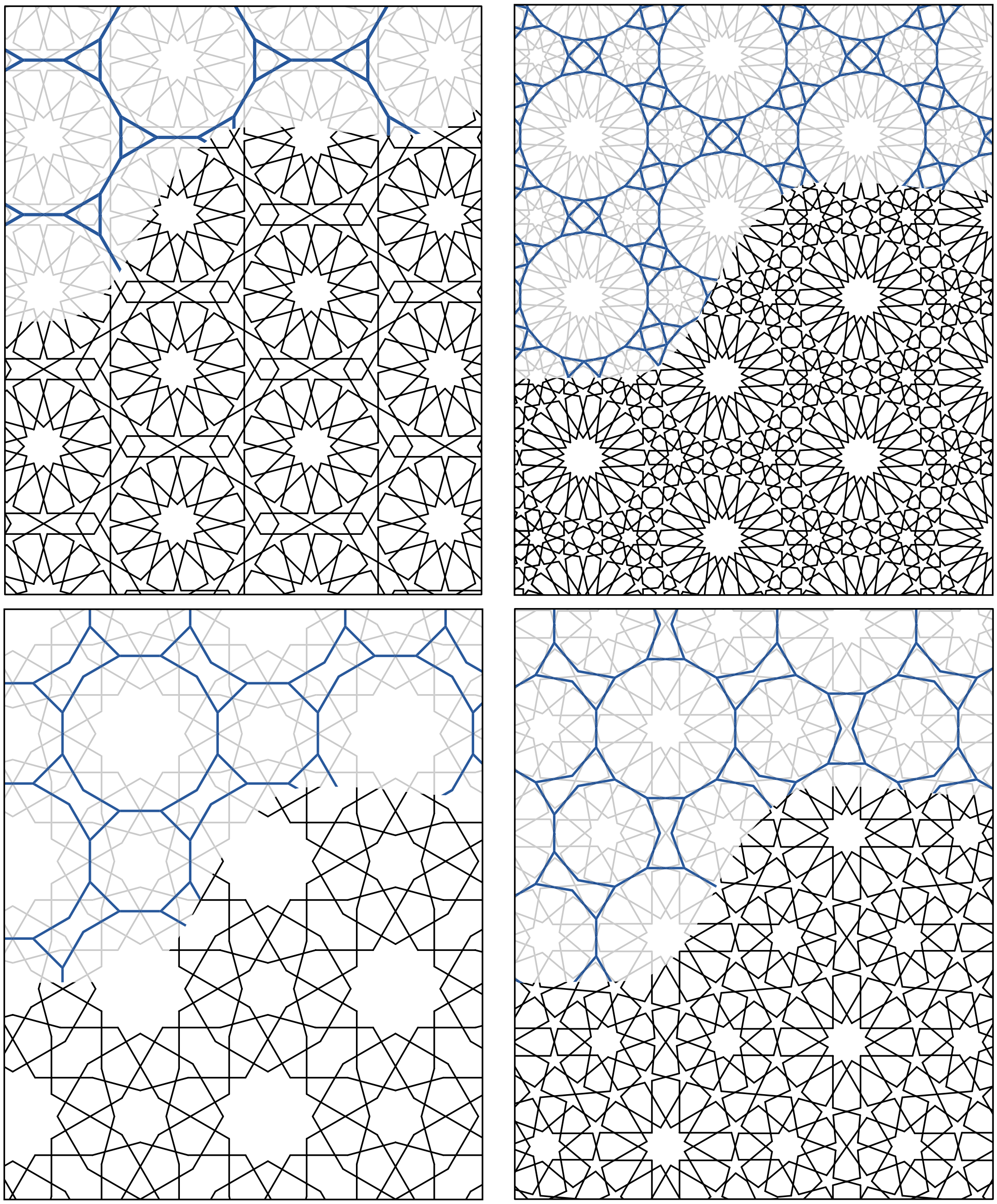


Figure 3.18 Examples of designs constructed using Taprats. Each example shows part of the template tiling from which the design was produced.

the resulting system embodies a model that has many continuous degrees of freedom, but is limited to a multiway switch in the choice of tiling. In this section, I demonstrate how the design space of Islamic star patterns can be fleshed out to include a parameterized model for the template tilings.

We know from the theory developed so far in this chapter that tilings containing many regular polygons as tiles make particularly good templates for Islamic star patterns because the regular polygons lead directly to stars or other traditional motifs. The tiling may contain other tiles that are not regular polygons, as long as they are not so oddly-shaped that inference fails on them. The main contribution of this section is a parameterized family of template tilings based on the explicit placement of regular polygons. Any leftover space is then divided into irregular tiles as necessary.

The most exciting fact about the tilings I present here is that they are carefully developed in a way that avoids any dependence on the parallel postulate; nothing in their construction is tied to Euclidean geometry. Looked at the right way, these tilings are constructions in the absolute geometry presented in Section 2.1.3 (the truth of this statement requires some logical precision and will be examined at the end of this section). From this specification of a tiling, a geometry-agnostic construction technique can then be applied seamlessly to produce Islamic star patterns in the Euclidean plane, the hyperbolic plane, and on the sphere.

The creation of non-Euclidean Islamic star patterns has been explored to some extent in the past. There is at least one historical case where a well-known Euclidean star pattern was reinterpreted in spherical geometry.[2] More recently, several designers have produced star patterns on the surfaces of the regular and Archimedean polyhedra [2, 98]; the most popular of these are Bonner's Geodazzlers [12], a commercially-available set of foldable polyhedral models. These polyhedral models are closely related to symmetric patterns in spherical geometry. Fewer examples exist of hyperbolic Islamic star patterns. When Lee wrote about star patterns in 1995, he knew of no such examples [99]. Among the star patterns Abas displays in his online gallery, there is a single, somewhat cumbersome hyperbolic design [1]. Building on his considerable experience with hyperbolic patterns, Dunham has recently produced numerous hyperbolic interpretations of Islamic geometric patterns [42], though he does not consider star patterns in particular. In most of these cases, the non-Euclidean patterns are found through a modification of an initial Euclidean pattern. By developing

[2] Jay Bonner, personal communication.

designs in absolute geometry, I take a more unified approach, constructing non-Euclidean designs directly without a translation step through the Euclidean plane. The result is more elegant in terms of both the patterns that are produced and the mathematical theory that underlies them.

Non-Euclidean star patterns represent a fairly valuable addition to the body of Islamic art. Islamic culture has always been interested in the beauty and elegance of mathematics, and non-Euclidean patterns are a fairly direct visualization of deep truths in geometry. Furthermore, although I am an outsider to Islam, I would speculate that non-Euclidean patterns sit well with the intent of star patterns in general to proclaim the crystalline perfection of Allah. The sphere is a marvelous visualization of boundlessness, and the hyperbolic plane of infinity (we will see in Section 4.7 how Escher exploited this fact in his tilings).

In this section, I present my family of tilings using only facts from absolute geometry. I then give revised design elements that have all reliance on parallelism removed. This generalized technique is then implemented in a C++ program called **Najm** (pronounced **Nazhm**, arabic for "star"). I discuss the implementation of Najm and techniques for decorating star patterns. Finally, I examine in more detail the statement that Najm is a construction in absolute geometry.

3.6.1 Template tilings

Although regular polygons in a template tiling represent regions of higher local symmetry than is describable using a wallpaper group, those regular polygons still tend to interact closely with the group's symmetries. Regular polygons will tend to be centered on the rotational axes. I use this observation to build a system of tilings where regular polygons are explicitly placed around rotational axes of a symmetry group.

The construction presented here adapts itself to any symmetry group of the form $[p, q]$, as presented in Section 2.2. Accordingly, let p and q be given, with $p, q > 2$.

The regular tiling $\{p, q\}$ has centers of p-fold, 2-fold, and q-fold rotation at its face centers, edge midpoints, and vertices, respectively. When the tiling's symmetry group $[p, q]$ is visualized through copies of its generating triangle, these rotational axes correspond exactly to triangle vertices A, B, and C as labeled in Figure 3.19. These vertices will serve as the set of potential centers for regular polygons.

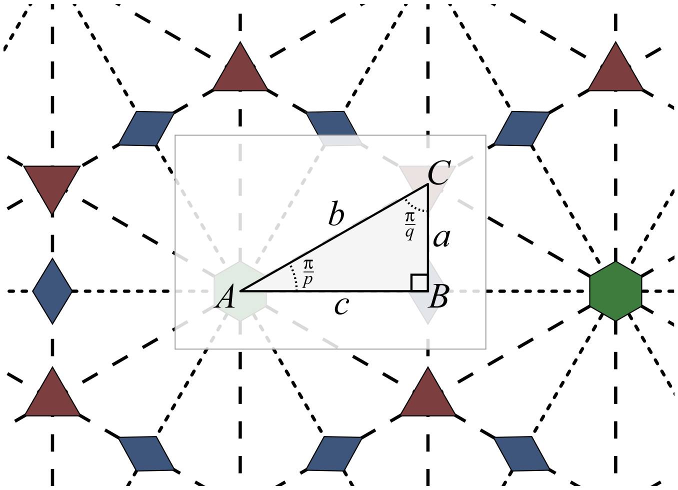


Figure 3.19 The canonical triangle used in the construction of Najm tilings.

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 3.20. We are therefore left with the following free parameters in defining an 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 twofold axes, with the exception that we do not permit $m_B = 1$, which would result in a degenerate two-sided polygon.

To record the orientations o_A, o_B and o_C unambiguously, we use the designated rays $\overrightarrow{AB}$, $\overrightarrow{BC}$, and $\overrightarrow{CA}$ respectively at vertices A, B, and C. The symbols v and e can then be used to determine whether a polygon should present a vertex or an edge midpoint on its designated ray.

Given a symmetry group $[p, q]$, we represent a given set of multipliers and orientations using the notation $([p, q]; m_A o_A, m_B o_B, m_C o_C)$. 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 a regular polygon with pm_A sides should be centered on vertex A, oriented according to o_A. The polygons at B and C are

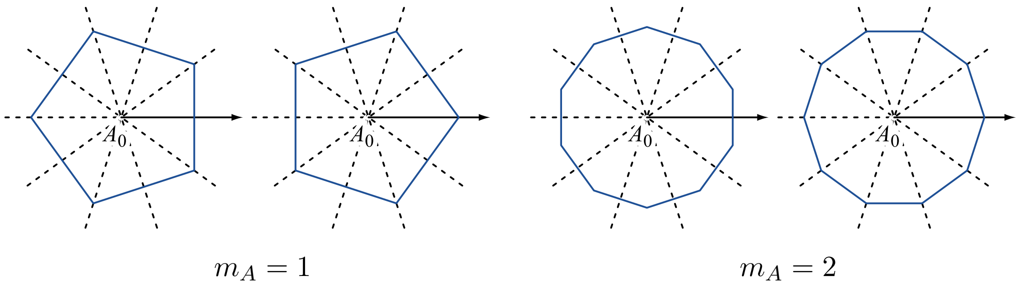


Figure 3.20 Examples of valid orientations for on-axis polygons around a fivefold rotational axis. The first and third examples have edge midpoints lying on the designated ray (marked by an arrow). The second and fourth have vertices on the ray. The notation $o_A = \mathsf{e}$ and $o_A = \mathsf{v}$ refers respectively to these two cases.

similarly determined.

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 give explicit radii that achieve contact, these scaling operations are fundamental enough that we make them an integral part of the notation.

I refer to the scaling process applied to the regular polygons as “inflation.” When it is a 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 a 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 non-zero 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.

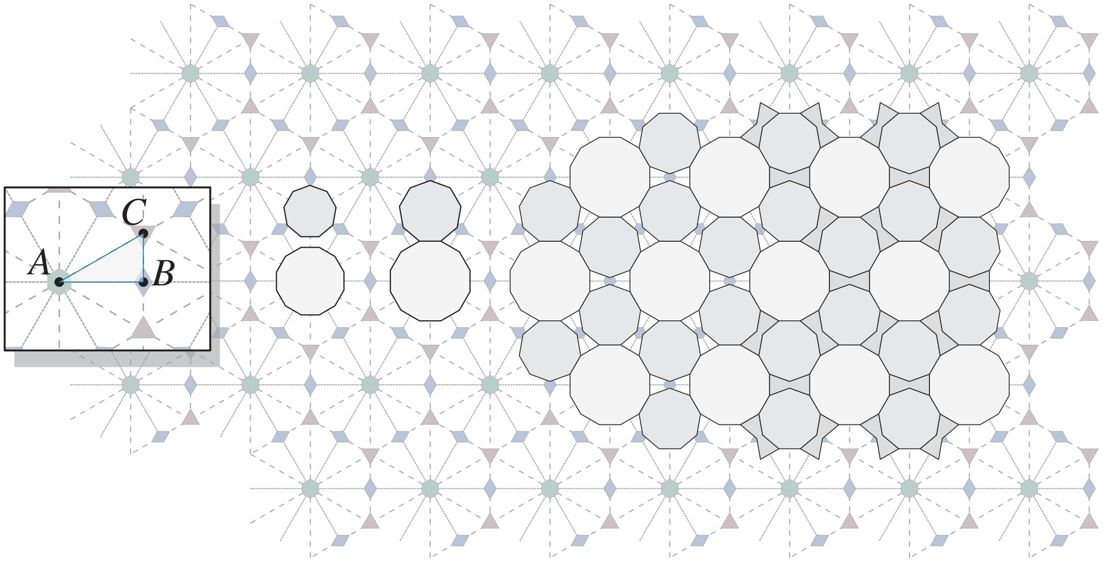


Figure 3.21 An example showing step by step how the tiling $([6,3];2\mathsf{e},0,3\mathsf{e})$ is constructed. The inset on the left shows the labels on a single fundamental region. Next, dodecagons and enneagons are placed at vertices A and C, respectively. The polygons are then scaled until they meet and have the same edge length. These polygons can be copied to all other fundamental regions, leaving behind a set of bowtie-shaped holes. Finally, the holes are filled in using additional tiles.

$A = r$, $B = r$, $C = r$ ($r \in \mathbb{R}$): 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 the formulae of absolute trigonometry. Radii for the three on-axis polygons can be solved for in closed form, though it is typically more straightforward (and nearly as precise) to solve them numerically.

Some definitions simplify the presentation of the formulae 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 as above. Let variables α, β, and γ represent a permutation of the triangle vertices A, B, and C. If m_α is nonzero, let $\mathcal{P}_\alpha$ be the regular polygon centered at vertex α.

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 (α, β) 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_α is nonzero, then $\mathcal{P}_\alpha$ is a regular n-gon centered on vertex α with some radius r_α. 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 $\mathrm{T}(l_{\alpha\beta}) = \mathrm{T}(r) \cos \frac{\pi}{n}$ (the function $\mathrm{T}(x)$ is defined in Section 2.1.4). Note that this calculation can be reversed as well; given one of a polygon's extents, we can determine the radius.

Given these definitions, there are four inflation operations to solve:

1. **Inflating a polygon to another polygon.** In this case we have $\mathcal{P}_\alpha$, a regular n_α-gon with fixed radius r_α centered at vertex α, and $\mathcal{P}_\beta$, a regular n_β-gon at vertex β. 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_β that gives $l_{\beta\alpha} = d - l_{\alpha\beta}$.

2. **Inflating a polygon to the generating triangle.** Here, the inflation of regular n_α-gon $\mathcal{P}_\alpha$ centered at α 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 α. Let d be the perpendicular distance from α to the opposite edge of the triangle. We assume that α is A, or C, since we then have the simpler case that d is the length of one of the triangle edges. The case $\alpha = B$ is more complicated. It could also be solved numerically, although I omit the details because this case 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_α so that $l_{\alpha\beta} = d$. By the definition of extent, we will either have $r_\alpha = d$ or $\mathrm{T}(r_\alpha) = \mathrm{T}(d)/\cos\frac{\pi}{n}$. A similar argument yields the solution for the case $\alpha = C$.

3. **Simultaneous inflation of two polygons.** Here, we have regular polygons $\mathcal{P}_\alpha$ and $\mathcal{P}_\beta$ with n_α and n_β 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, $\bigcirc(l_{\alpha\beta}) = \bigcirc(x)/\sin\frac{\pi}{n_\alpha}$ or $\bigcirc(l_{\alpha\beta}) = \mathrm{T}(x)/\tan\frac{\pi}{n_\alpha}$ 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_β. Since n_α and n_β are given, the equation $l_{\alpha\beta} + l_{\beta\alpha} = d$ has x as its single unknown. A solution for x could be used to back out final values for r_α and r_β.

   In my implementation, I observe that the expression $l_{\alpha\beta} + l_{\beta\alpha} - d$ is monotonic in x and solve for x numerically using binary search. It is also possible to solve for x algebraically. For instance, if $\mathcal{P}_\alpha$ and $\mathcal{P}_\beta$ both meet $\alpha\beta$ at edge midpoints, we have

$$\bigcirc^{-1}\left(\frac{\mathrm{T}(x)}{\tan\frac{\pi}{n_\alpha}}\right) + \bigcirc^{-1}\left(\frac{\mathrm{T}(x)}{\tan\frac{\pi}{n_\beta}}\right) = d$$

   which, after some manipulation, gives

$$x = \mathrm{T}^{-1}\left[\bigcirc(d)\left(\frac{1}{\tan^2\frac{\pi}{n_\alpha}} + \frac{1}{\tan^2\frac{\pi}{n_\beta}} + \frac{2\mathrm{E}(d)}{\tan\frac{\pi}{n_\alpha}\tan\frac{\pi}{n_\beta}}\right)^{-\frac{1}{2}}\right]$$

   The complexity of the algebraic solution casts a solid vote for the practicality of finding x numerically.

4. **Simultaneous inflation of all three polygons.** 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.

When the inflation process is complete, the result will be one or more regular polygons centered on the vertices of a single generating triangle. By the definition of a generating triangle for $[p, q]$, we can extend the placement of regular polygons to the whole plane simply by applying successive reflections across triangle edges. The symmetry group of the resulting tiling is $[p, q]$.

As an example of this notation, consider the tiling $([6, 3]; 2\mathtt{e}, 0, 3\mathtt{e}; AC)$, the construction of which is shown step-by-step in Figure 3.21. Here, we have $m_A = 2$ in symmetry group $[6, 3]$, meaning that we will place regular dodecagons at every center of sixfold rotation. Because $o_A = \mathtt{e}$, these dodecagons will be oriented so that an edge midpoint lies on the ray $\overrightarrow{AB}$. We also have $m_C = 3$ and $o_C = \mathtt{e}$, and so a regular enneagon will be placed at every center of threefold rotation, oriented so that an edge midpoint lies on $\overrightarrow{CA}$. The inflation symbol AC indicates that we should inflate the dodecagons and enneagons until they meet and have equal edge lengths. This inflation is possible because there are two equations (one causing the polygons to meet, one equating their edge lengths) in two unknowns (the radii r_A and r_C). Once the inflation is performed, the remaining empty space in the plane is consumed by additional bowtie-shaped tiles, and the tiling construction is complete. The resulting tiling can then be used to create numerous star patterns. An example using rosettes appears in Figure 3.17. Other examples of tilings produced by this notation are given in Figure 3.22.

This notation for tilings is very flexible and can express a large number of the tilings that underlie Islamic star patterns. However, it was chosen for its adaptability to non-Euclidean geometry and not for its universality in the Euclidean plane. As a result, there are some simple, well-known Euclidean tilings such as the one in Figure 3.2 that cannot be represented via this notation. They are primarily the ones that do not have symmetry group $[4, 4]$ or $[6, 3]$; every Euclidean tiling produced by Najm will have one of these two as its symmetry group. Such tilings can still be used in Taprats, which provides a convenient user interface for drawing periodic Euclidean template tilings by hand. Alternatively, it may be possible in some special cases to extend the notation above to symmetry groups other than those of the form $[p, q]$.

At the start of this section, I pointed out the relationship between polyhedra and patterns in spherical geometry. When p and q are chosen to yield a spherical symmetry group in the notation above, many of the resulting tilings can easily be converted into polyhedra simply by taking the convex hull of the vertices of all generated regular polygons. George Hart and I have experimented with these polyhedra, which we call “symmetrohedra.” This class of polyhedra contains all of the Platonic and Archimedean solids except for the snubs [84], and generalizes them to provide a family of symmetric convex solids with many, but not necessarily all, regular faces. Examples of symmetrohedra are given in Figure 3.23. Hart and Leigh Boileau have also executed some of the novel symmetrohedra as sculptures in wood and metal.

3.6.2 Motifs in absolute geometry

Now that I have a geometry-neutral source of tilings to use in the creation of Islamic star patterns, we must reexamine the way motifs are generated to make sure that the same algorithms apply in absolute geometry. Certainly there is no problem expressing planar maps in absolute geometry; points and line segments still exist. However, parts of the construction process might require modifications to remove dependencies on the parallel postulate.

As a first step, observe that the inference algorithm of Section 3.4 can be applied unmodified in absolute geometry. We may still speak of rays emanating from contact points, forming given contact angles with a tile edge. Rays may or may not intersect, and if they do we can still calculate the length of the line segments that join the contact points to the intersection point.

The path-based construction of symmetric motifs still works. And although the design element model for stars requires no modification, the formula for converting between $\{n/d\}$ notation and the contact angle θ requires some additional work in absolute geometry. Using some trigonometric manipulation, it can be shown that the star $\{n/d\}s$, inscribed in an absolute circle of radius r, has

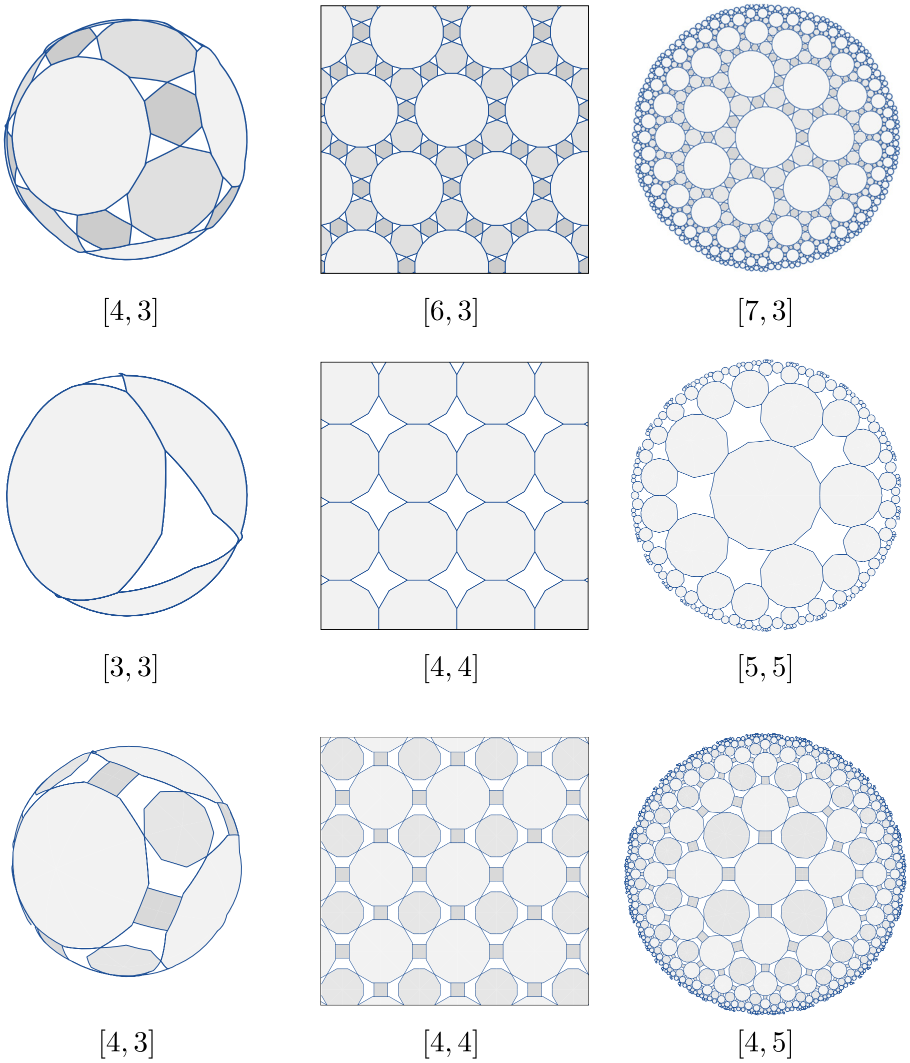
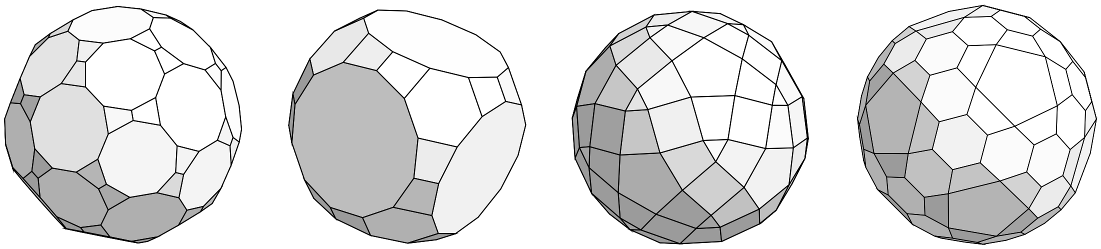


Figure 3.22 Examples of tilings that can be constructed using the procedure and notation given in Section 3.6. The tilings are of the form $([p,q]; 4\mathsf{e}, 3\mathsf{v}, 3\mathsf{e}; ABC)$ in the first row, $([p,q]; 3\mathsf{e}, 0, 0; A)$ in the second row, and $([p,q]; 3\mathsf{e}, 2\mathsf{e}, 3\mathsf{v}; A = 0.85, B, C)$ in the third. The symmetry group is indicated under each tiling.



$([5,3];2\mathsf{e},0,3\mathsf{e},AC)$ $([4,3];3\mathsf{v},0,2\mathsf{e},AC)$ $([5,3];1\mathsf{v},2\mathsf{v},0,AB)$ $([5,3];0,3\mathsf{v},2\mathsf{v},BC)$

Figure 3.23 Examples of symmetrohedra, symmetric polyhedra based on the tilings described in Section 3.6.1. Each solid is derived from its given tiling symbol by building the spherical tiling and taking the convex hull of the tiling vertices.

contact angle $\theta = \pi/2 - \phi$, where ϕ is given from the equation $\tan(\phi) = 1/(\mathrm{E}(r)\tan\frac{\pi d}{n})$.

The rosette model presents the first major change in moving to absolute geometry. The construction shown in Figure 3.13 computes point G as lying on the line through point C parallel to line $\overline{OM}$. The existence of this line relies on a direct application of the parallel postulate! Fortunately, we can sidestep Euclid by noting that to have $GC = GM$, vertex G must lie on the bisector of $\angle CAM$. Point G can therefore be found as the intersection of that bisector with $\overline{MM'}$. Even better, this construction adapts to any contact angle θ by intersecting the bisector with $\overline{MA'}$, where A' is obtained by rotating A by an angle of θ around M. The value of θ that yields the ideal rosette is then simply $|\angle AMM'|$, which depends only on n and r.

Extended motifs require some special treatment as well. As in the Euclidean case, we are given n, r, and a contact angle θ, and must produce r' and θ' so that when extension is applied to an inner motif with contact angle θ' inscribed in a regular polygon of radius r', the result is a larger motif with contact angle θ and radius r.

Once again it is possible, but complicated, to obtain an algebraic solution for r' and θ', but relatively easy to solve for these values numerically. The solution is illustrated using the diagram in Figure 3.24. The outer polygon is a regular n-gon of radius r. We can determine the location of point B as the intersection of the two blue rays $\overrightarrow{AB}$ and $\overrightarrow{A'B}$, in much the same way that we would in the inference algorithm. We can then compute θ' from the observation that $4\theta' + 2\angle ABA' = 2\pi$. Furthermore, if d is the length of segment OB, then we can obtain r' from $\mathrm{T}(r') = \mathrm{T}(d)/\cos\frac{\pi}{n}$.

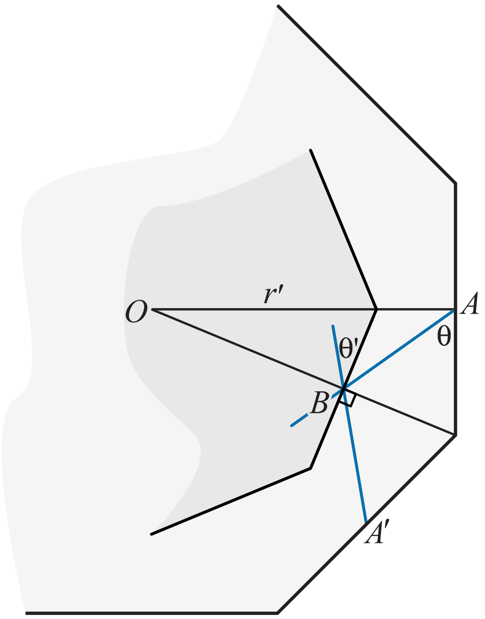


Figure 3.24 A diagram used to build extended motifs in absolute geometry. See the text in Section 3.6.2 for more details.

3.6.3 Implementation

The preceding construction for template tilings, together with the geometry-neutral versions of the inference algorithm and the design elements, have been implemented as a C++ program called **Najm**. The architecture of Najm is sufficiently interesting to warrant detailed description.

In many ways, the large-scale structure of plane geometry mimics the behaviour of classes and instances in an object-oriented language A geometry as a formal system is like the declaration of an abstract data type, with the model of that system acting as the data type's implementation. Absolute geometry is then very much an abstract base class with Euclidean, spherical, and hyperbolic subclasses. Each of the subclasses adds behaviour (specifically, the behaviour of parallel lines), but we can still do a great deal of geometry by accessing only those behaviours present in the base class.

Najm is divided into two layers. The lower layer is an independent library that provides an abstract interface to absolute geometry. Tools are then written in a geometry-independent way on top of the lower layer. 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.

The expression of this abstraction layer must be carefully designed so that the interface is as easy to work with as a familiar set of classes implementing Euclidean geometry. At the same time, the expressibility should not come at the expense of runtime speed or efficient storage. In a language like Java, where all non-primitive data types are heap-allocated, we must immediately accept the performance hit of indirection. On the positive side, we might then implement, say, an `AbsolutePoint` class with subclasses `EuclideanPoint`, `SphericalPoint`, and `HyperbolicPoint`. In this case, branching to the appropriate model (implementation) of geometry is carried out at runtime through casting and dynamic dispatch. The resulting library would be expressive, but relatively inefficient.

Instead, I implement the absolute geometry library in an efficient, typesafe, and expressive manner by using explicit specialization of templated classes in C++ [104, section 16.9]. Three "tag" classes are defined: `Euclidean`, `Spherical`, and `Hyperbolic`. These classes have no members, and act effectively as constants that branch to the proper implementation at compile time. The familiar objects of geometry such as points, lines, line segments, and symmetry groups are declared as templated classes parameterized on a single type variable. I then give specialized versions of those classes with the type variable set to one of the three tag classes. Whereas the decision to branch to a particular implementation is made at runtime in a language like Java, here we can make the decision at compile time.

For example, a generic `point<T>` class is declared but not defined. The generic declaration is then overridden by three specialized classes `point<Spherical>`, `point<Euclidean>`, and `point<Hyperbolic>`. A client can write generic code that manipulates objects of type `point<T>` (in the same way that a Java programmer might manipulate only `AbsolutePoints`), 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 speed or space overhead of indirection. This example is illustrated in Figure 3.25.

Clients that are parameterized on the tag types above can carry out any construction in absolute geometry. The concrete implementation of that construction is then written out by the compiler when the client code is instantiated with `Spherical`, `Euclidean`, or `Hyperbolic`. There is no run time penalty in using this abstraction layer, although the heavy use of C++ templates increases compilation time and the sizes of generated object files.

3.6.4 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: $[3,3]$, $[3,4] \cong [4,3]$ and $[3,5] \cong [5,3]$.[3] These three groups are finite, so we precompute rigid motions for all copies of the generating triangle and store them in tables. No fill region is specified; the sphere is simple enough that we always draw the entire pattern.
- The Euclidean groups $[3,6] \cong [6,3]$ 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 the algorithm discussed in Section 4.4.3.
- 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 [47, 101]. We base our code directly on the pseudocode presented by Dunham *et al.* [40, 43]. The regions we fill are discs centered at the origin in the Poincaré model.

3.6.5 The meaning of Najm

At first glance, the implementation of Najm may seem somewhat mysterious. After all, the construction of star patterns relies on concrete mathematics like measurements of distances and angles, intersections between lines, even the construction of planar maps. How can all these calculations be carried out in absolute geometry, a logical system where concepts like distance exist only formally?

[3] Although the *prismatic groups* $[2,q]$ and $[p,2]$ are also regular and spherical, we are not interested in them for the purposes of creating Islamic star patterns.

```
// Define the three tag classes
class Euclidean {};
class Spherical {};
class Hyperbolic {};

// Any definitions provided in this ’abstract base class’ are purely
// for documentation purposes, since they will never be called.
template<typename geo> class Point {
    double distance( const Point<geo>& other ) { return 0.0; }
};

// A point in two-dimensional Cartesian coordinates
class Point<Euclidean> {
    double distance( const Point<Euclidean>& other ) {
        return sqrt( (x-other.x)*(x-other.x) + (y-other.y)*(y-other.y) ); }
    double x, y;
};

// A point in the hyperboloid model of the hyperbolic plane
class Point<Hyperbolic> {
    double distance( const Point<Hyperbolic>& other ) {
        return acosh( z*other.z - x*other.x - y*other.y ); }
    double x, y, z;
};

// A point on the sphere
class Point<Spherical> {
    double distance( const Point<Spherical>& other ) {
        return acos( x*other.x + y*other.y + z*other.z ); }
    double x, y, z;
};

// An example of a geometry-independent client function.
template<typename geo> double perimeter( const vector<Point<geo>>& pts ) {
    double total = 0.0;
    for( size_t idx = 0; idx < pts.size(); ++idx ) {
        total + = pts[idx].distance( pts[(idx+1)%pts.size()] );
    }
    return total;
}
```

Figure 3.25 An excerpt from the absolute geometry library underlying Najm, showing the class specialization technique. C++ templates allow code to be parameterized over a choice of geometry without incurring the runtime overhead of a class hierarchy. Here, I demonstrate the declaration of a simple point class, together with its specializations and a sample client function.

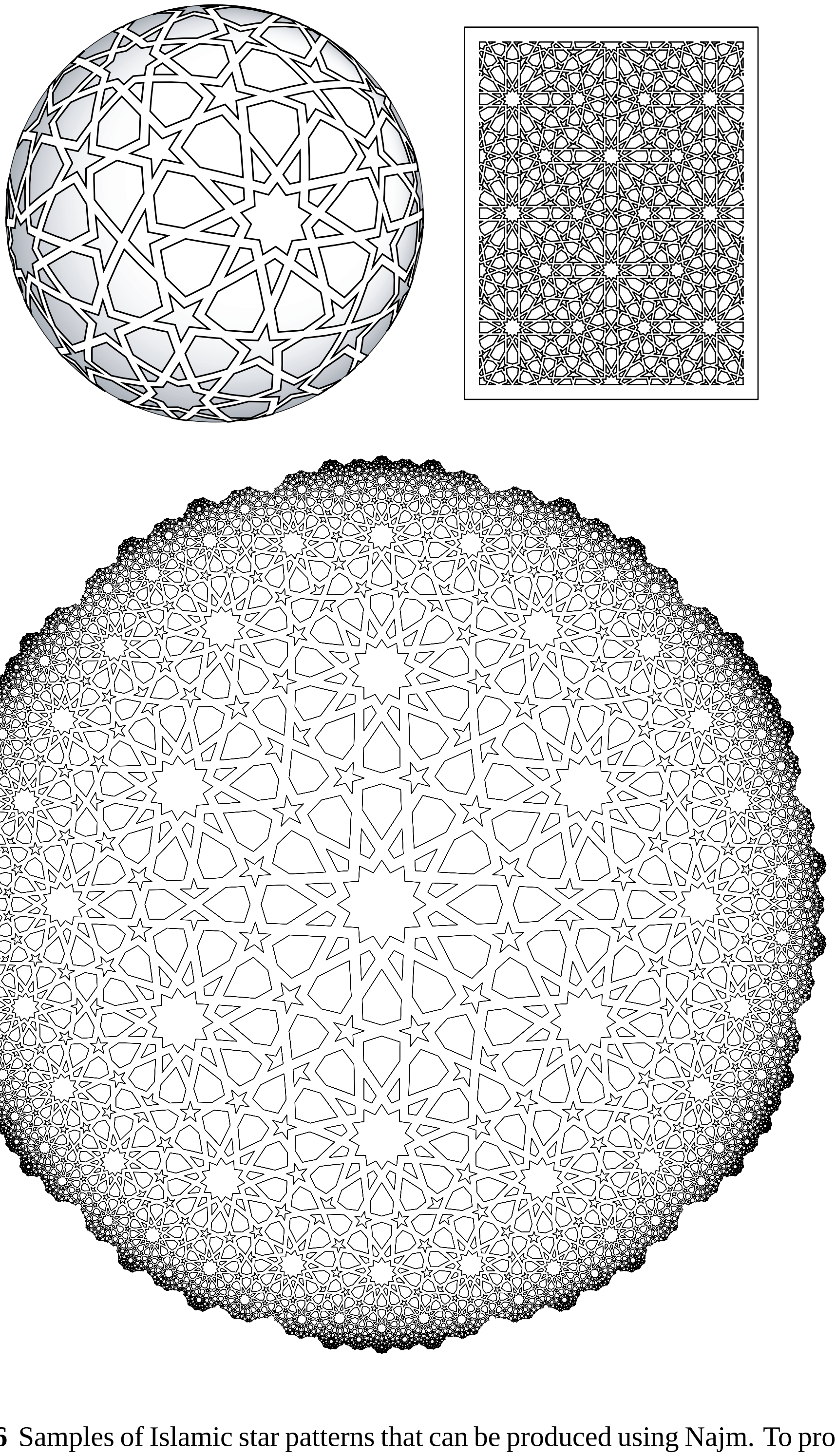


Figure 3.26 Samples of Islamic star patterns that can be produced using Najm. To provide a basis for comparing patterns across geometries, each page presents a single conceptual design interpreted in each of the three different geometries.

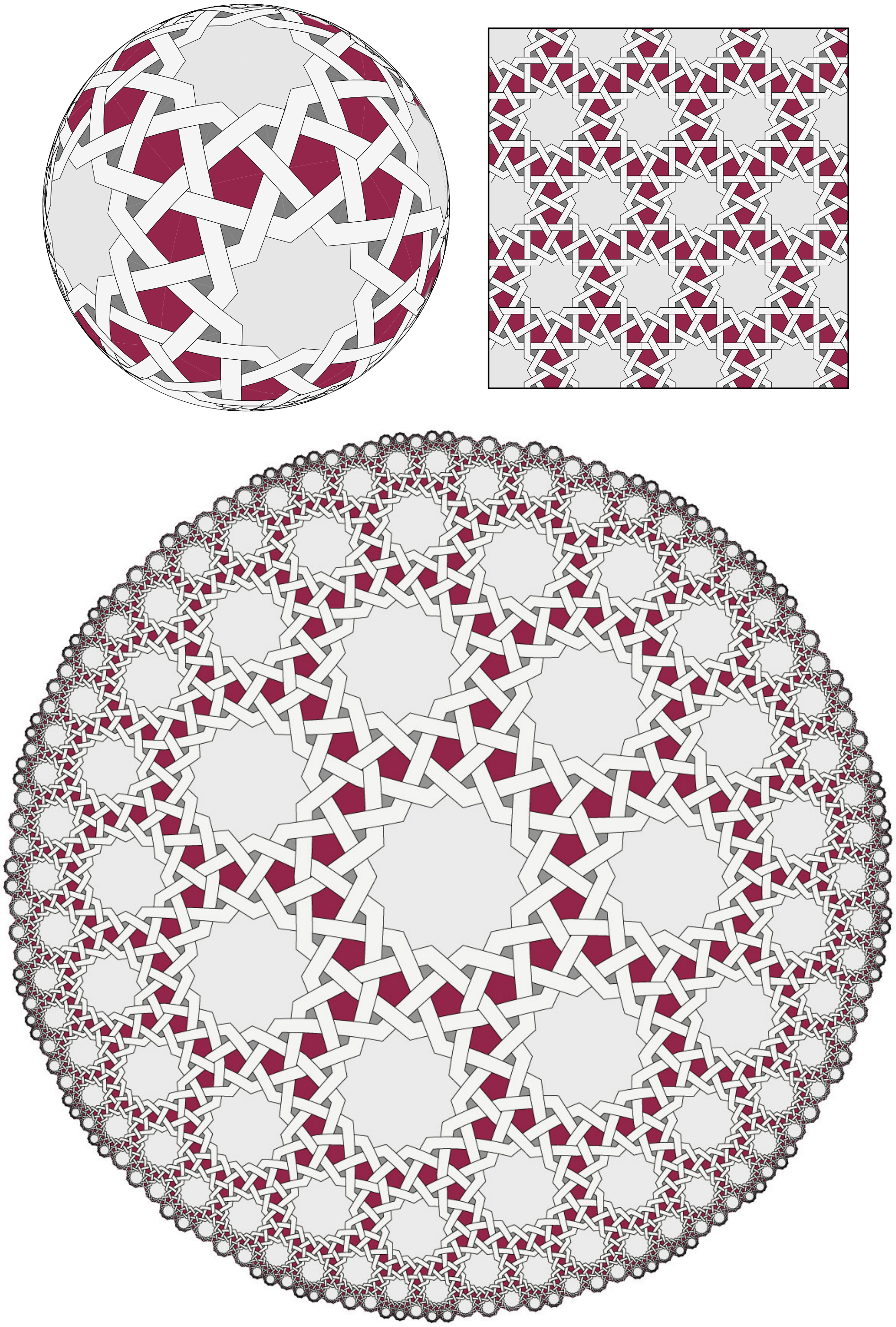


Figure 3.26 (continued)

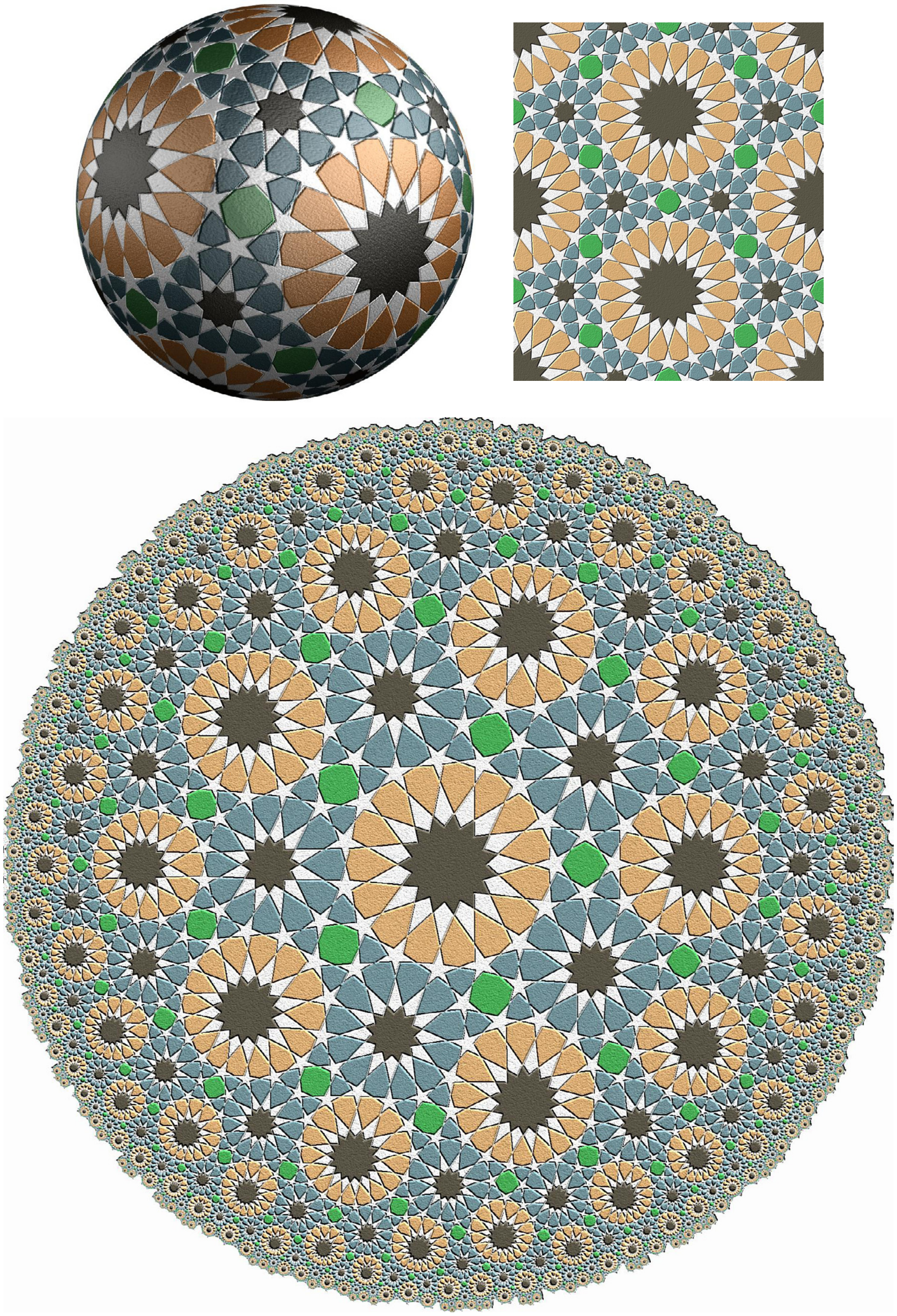


Figure 3.26 (continued)

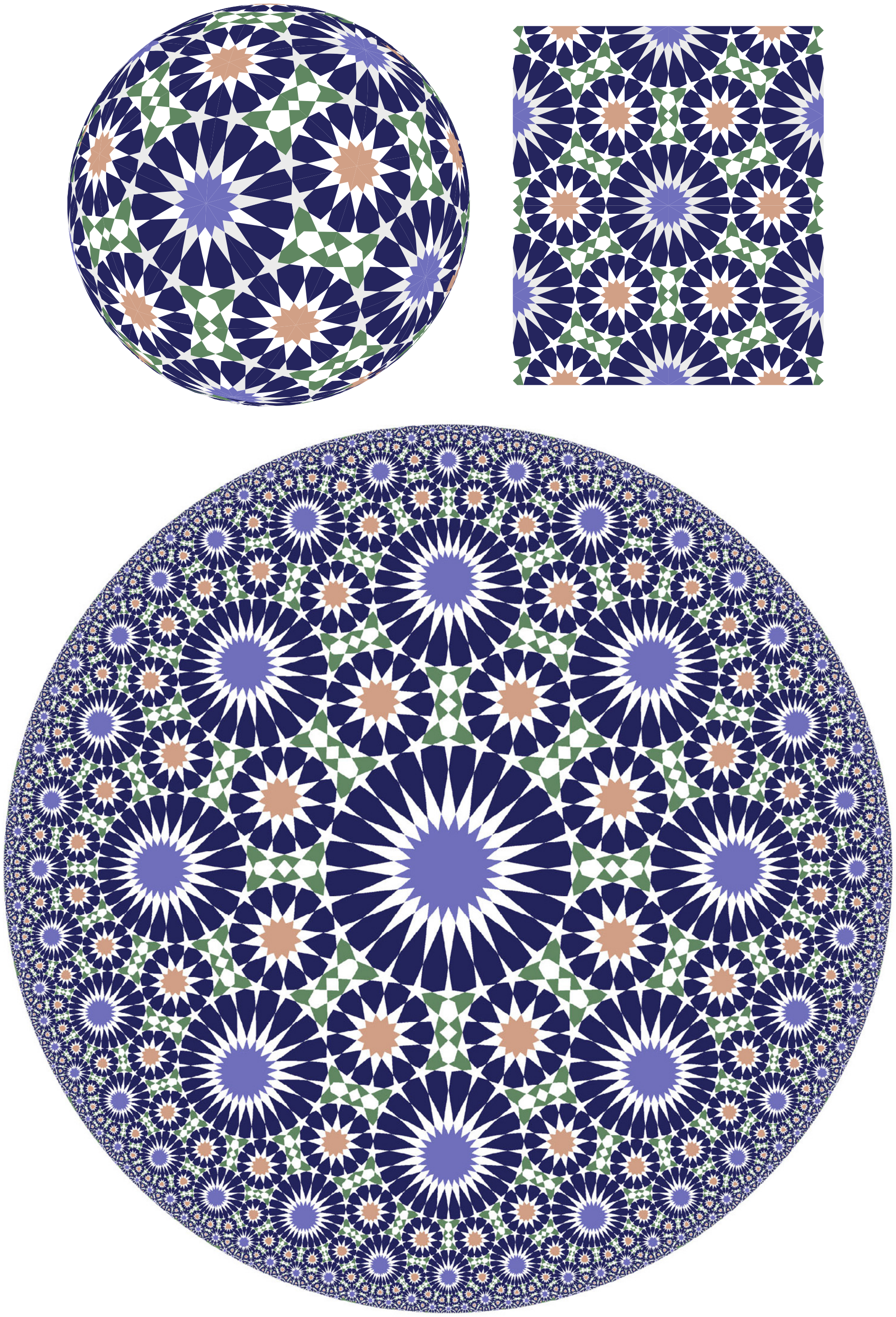


Figure 3.26 (continued)

The simple answer is that such calculations cannot be carried out. The shuffling of numerical values and the representation of objects like points using coordinates are properties of a *model* of geometry, not of geometry itself. That the distinction between the two is hard to visualize is a byproduct of the categoricalness of Euclidean geometry. As was mentioned in Section 2.1.3, we need not make a mental distinction between the axioms of Euclidean geometry and the Cartesian plane, because the latter is (up to isomorphism) the only way to represent the former.

The implementation still manages to carry out computation, though, because it makes an explicit allowance for the substitutability of models in absolute geometry. The key is that a client of the absolute geometry library is parameterized on the choice of model. This parameterization takes the form of the template argument `geo` in the `perimeter` function of Figure 3.25. This seemingly inocuous parameter is shorthand for a whole collection of data structures and algorithms that implement geometry in the Euclidean plane, the hyperbolic plane, or the sphere. That implementation is pulled in by consistently using the `geo` parameter in the client implementation, for example by referring to points as `point<geo>`.

Although the implementation itself is strong evidence that this approach works, the explanation still suffers from a subtle but interesting logical hole.

In geometry, a construction is really just a theorem, a theorem stating that the construction is possible. In some sense, the elaborate method of star pattern construction presented here is then just a very long-winded theorem of absolute geometry. What is that theorem?

The construction of star patterns is controlled by a set of parameters: the notation describing the tiling (which includes the symmetry group $[p, q]$, multipliers m_A, m_B, and m_C, and so on), and the parameters that assign motifs to the tile shapes. Let us extract from this collection the numbers p and q, and lump all remaining parameters together into a set S.

An initial attempt to state the "theorem of star patterns" might read as follows: "for all p, q, and S, we can construct a star pattern with symmetry group $[p, q]$, as dictated by the parameters in S." Unfortunately, this statement is absolutely untrue! For any given p and q, the existence of symmetry group $[p, q]$ is equivalent to one of the three versions of the parallel postulate, and is therefore independent from the axioms of absolute geometry. Worse still, asserting the existence of $[p, q]$ for all values of p and q simultaneously is like trying to have a single geometry where all three parallel postulates hold, an obvious impossibility.

The existence of symmetry group $[p, q]$ is the geometric fact that gets the entire construction process off the ground, but it is the one fact we cannot prove in absolute geometry. The way out of this quandary is to *suppose* that $[p, q]$ exists as a condition of the theorem: "for all p and q, if symmetry group $[p, q]$ exists then for all S we can construct a star pattern with symmetry group $[p, q]$, as dictated by the parameters in S." This subtle change eliminates all inconsistency because the theorem does not assert that any of the groups $[p, q]$ actually exist, but should one exist, star patterns can be constructed from it. This view of star pattern construction comes closest to expressing the mathematical "meaning" that underlies the implementation of Najm.

3.6.6 Results

Examples of rendered star patterns in the three planar geometries are given in Figure 3.26. In each case, a single choice of motifs and rendering style is applied across related Euclidean, hyperbolic, and spherical symmetry groups. This consistency makes clear the similarity between related symmetry groups. Typically, the change is that one of the stars in the Euclidean design will have fewer points in the spherical design, and more points in the hyperbolic design.

3.7 Decorating star patterns

Although I have already given some examples of decorated star patterns, I have postponed the discussion of how these decorations are carried out until now so that I can describe the algorithms once and for all in the absolute geometry framework presented in the previous section.

I distinguish between two kinds of decoration styles for star patterns. *Ornamentation* is the addition of non-geometric figures, such as curvilinear floral motifs, to the underlying design. I do not attempt to generate such motifs, although they could in principle be supplied by the user (and perhaps generated to fit the design by adapting the technique of Wong *et al.* [139]). My focus is on *geometric rendering*: purely mathematical operations on the vertices, edges, and faces of the planar map itself.

We can use the high degree of symmetry of Najm's template tilings to simplify decoration and rendering. We compute the restriction of the overall design to a single generating triangle. That restricted map, which I call the "fundamental map," contains all geometric information necessary to

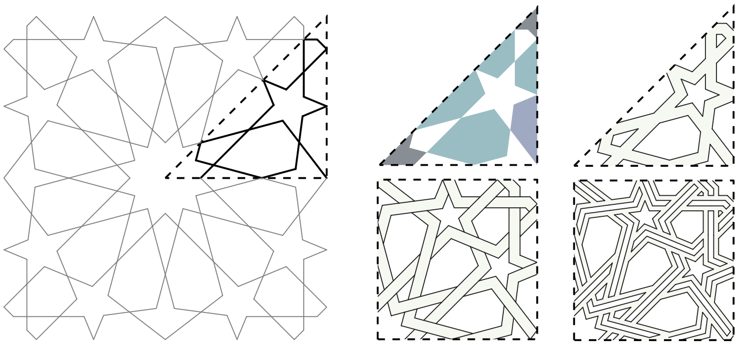


Figure 3.27 Examples of decoration styles. The undecorated fundamental map is shown on the left, followed by the filled, outline, interlaced, and outlined-interlace decoration styles. To make the over-and-under patterns consistent in the two interlaced cases, the decoration must be carried out on two adjacent fundamental maps.

render any amount of the final design. To create a decorated design, it suffices to apply a decoration style to only one or two copies of the fundamental map. Figure 3.27 shows examples of decorated fundamental maps.

3.7.1 Filling

A simple and effective decoration style is to colour the faces of the fundamental map, including faces bordered by the generating triangle. This style emulates the many real-world examples executed using coloured clay tiles.

Because the designs produced by Najm have only 2-valent and 4-valent vertices, the map can be 2-coloured. My implementation automatically 2-colours the fundamental map as a basis for user selection of face colours. Following tradition, one set of faces in the 2-colouring will typically be left white and the other set will receive a range of colours, chosen interactively by the user.

3.7.2 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, construct the two parallels at distance $w/2$ to l. Unfortunately, these parallels are not

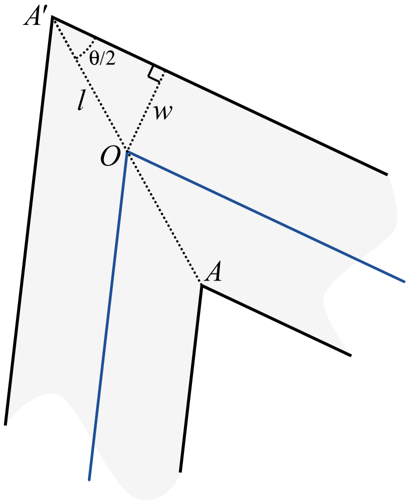


Figure 3.28 A diagram used to compute the mitered join of two line segments in absolute geometry. The diagram is used in the text of Section 3.7.2.

well-defined in absolute geometry.

As was discussed in Section 2.1, the solution is to move from parallel lines to equidistant curves, the two loci of points of constant perpendicular distance $w/2$ from l. Equidistant curves can be manipulated with ease in absolute geometry, although they cannot, for example, be assumed to be straight (another property equivalent to the Euclidean parallel postulate).

The other operation that must be translated to non-Euclidean geometry is the mitered join of two thickened absolute line segments. The situation is shown in Figure 3.28. Suppose two segments (in blue) meet at a point O, where they create an angle of θ. The mitered join depends on the two points A and A', which we know lie on the bisector of of the blue lines and are equidistant from O. It remains to determine the distance $l = OA = OA'$. This distance can be found via a direct application of one of the identities of absolute trigonometry: $\bigcirc(l) = \bigcirc(w)/\sin\frac{\theta}{2}$.

3.7.3 Interlacing

The interlaced decoration style can be derived from the outline style by drawing additional curves at every crossing to suggest an over-and-under relationship. This style is of great importance both in Islamic art and in the ornamental traditions of other cultures, such as Celtic [29] and Byzantine [91].

As shown in Figure 3.27, 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 $[p, q]$ [26, section 4.4]. This group can cover the plane using only direct (non-reflecting) symmetries, which allows the interlacing to be carried to the whole plane consistently. The replication algorithm discussed in Section 3.6.4 must be modified slightly so that it does not draw any reflected fundamental regions.

3.7.4 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. This style is particularly effective when executed as a real-world object by a computer-controlled manufacturing system. Examples appear in Figures 3.35 and 3.37.

3.8 Hankin tilings and Najm tilings

A quick visual inspection shows that the tilings presented in this chapter fall loosely into two camps. There are simple tilings such as the ones defined using the notation of Section 3.6, which for the purposes of discussion I call “Najm tilings.” They are mostly made up of regular polygons with hole fillers completing the tiling as needed. Then there are “Hankin tilings,” the more complex tilings introduced by Hankin and used in Section 3.4. In a Hankin tiling the regular polygons are often surrounded by rings of irregular pentagons. And yet, once higher-level design elements such as rosettes are added to the construction method in Section 3.5, it seems as if the two families of tilings can produce similar (and sometimes identical) designs. Figure 3.29 gives two examples of how distinct tilings can lead to the same design.

Bonner uses both kinds of tilings to create star patterns, and he too observes that a single design

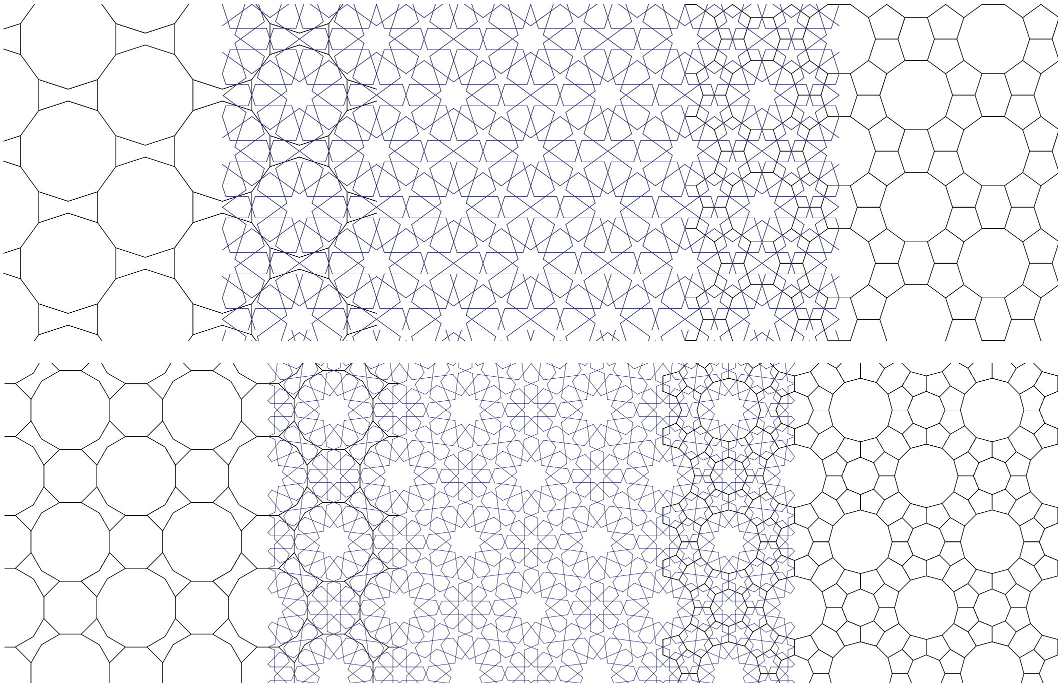


Figure 3.29 Examples of distinct tilings that can produce the same Islamic design. In each case, the tilings on the left is filled in using a combination of design elements and inference, and the tiling on the right uses inference alone. They meet in the shared design in the center.

may originate from two very different tilings (he makes this observation in the context of the tilings in the top row of Figure 3.29). In fact, there is a deep connection between the two tilings, one that should not be dismissed as coincidence. In this section, I explain the relationship between Hankin tilings and Najm tilings and discuss situations in which it might be preferable to use one family over the other.

I define an operation on Euclidean tilings called the *rosette transform*. The algorithm for the rosette transform is reminiscent of the inference algorithm: given a tiling, it constructs a planar map for each distinct tile shape. The planar maps are then assembled, this time into a new template tiling rather than a final design. The map for each tile shape is constructed in one of the following two ways:

- If the tile is a regular n-gon $\mathcal{P}$ of radius r with five or more sides, then the map is constructed

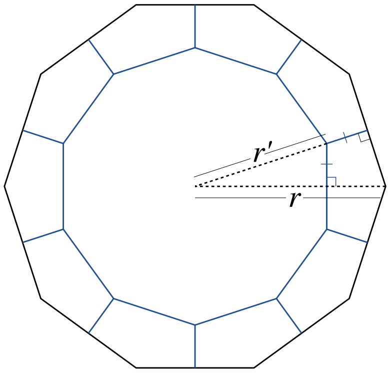


Figure 3.30 The rosette transform applied to a regular polygon. Here, a regular 10-gon of radius r (shown in black) is inscribed with a smaller regular 10-gon of radius r' (shown in blue) together with segments that join the vertices of the inner polygon to the edge midpoints of the outer one. The inner radius is chosen so that the marked edges have the same length.

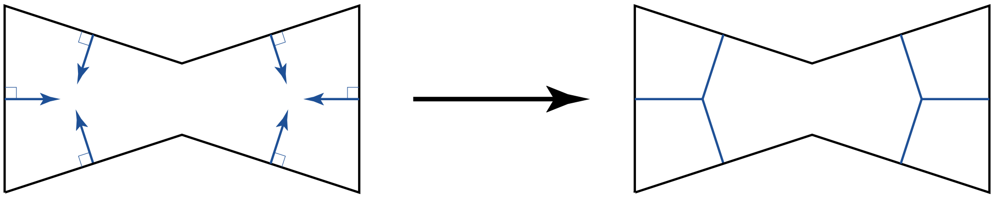
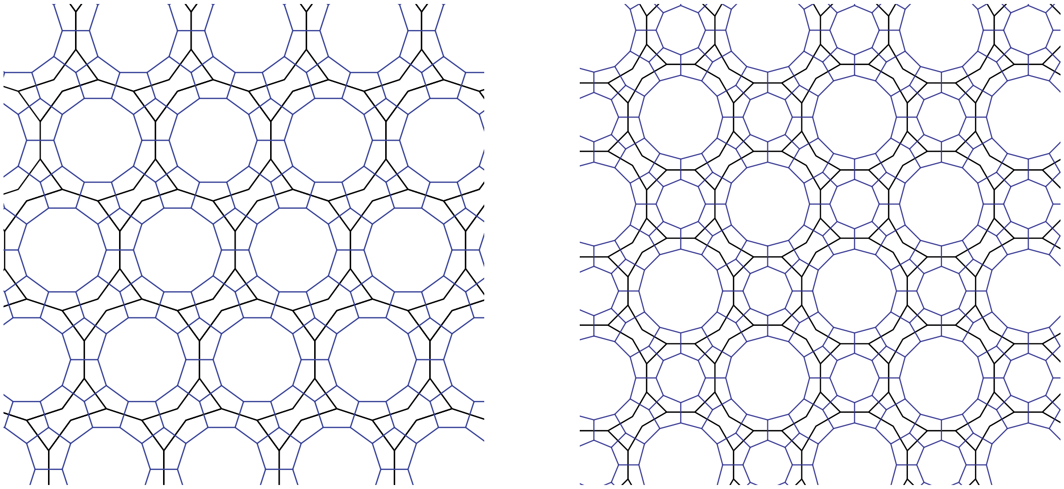


Figure 3.31 The rosette transform applied to an irregular polygon. On the left, a perpendicular bisector is drawn for every tile edge as a ray pointing to the interior of the tile. The rays are cut off when they meet each other, as with the inference algorithm.

as in Figure 3.30. We build a new regular n-gon $\mathcal{P}'$ with radius $r' < r$ and place it concentrically with the original polygon but rotated by π/n relative to it. We then add line segments connecting the vertices of $\mathcal{P}'$ to the edge midpoints of $\mathcal{P}$. The inner radius r' is chosen so that the length of each of these new segments is exactly half of the side length of $\mathcal{P}'$. Some trigonometry shows that given n and r, the correct value of r' is

$$r' = r\left(\cos\frac{\pi}{n} - \sin\frac{\pi}{n}\tan\left(\frac{\pi(n-2)}{4n}\right)\right)$$

The map returned is $\mathcal{P}'$ together with the segments joining it to the edge midpoints of $\mathcal{P}$.



(a) (b)

Figure 3.32 Two demonstrations of how a simpler Taprats tiling is turned into a more complex Hankin tiling. The simpler tiling is shown in black, and its rosette transform superimposed in blue.

- If the tile is a polygon $\mathcal{P}$ that does not satisfy the conditions above, we extend perpendicular bisectors of the sides of $\mathcal{P}$ towards its interior, as shown in Figure 3.31. The bisectors are truncated where they meet each other. The result is returned as the map for this tile shape.

  This step is similar in spirit to running the inference algorithm with a contact angle of 90° and is subject to the same pitfalls. We do not expect it to return a meaningful answer for every possible polygon, but in the cases of polygons that occur in Najm tilings, the map it discovers is “correct.” Some heuristics also work here that do not apply in the inference algorithm. One moderately successful heuristic is to consider the intersection points of all pairs of rays and to cluster those points that lie inside the tile. The clusters can then be averaged down to single points that all rays contributing to that cluster can use as an endpoint.

When this algorithm is run on Najm tilings, it tends to produce Hankin tilings. For instance, the two black tiles in Figures 3.30 and 3.31 correspond to the Najm-like tiling in the top row of Figure 3.29. When assembled into a complete rosette transform tiling, the result is the corresponding Hankin tiling. The two tilings are shown superimposed in Figure 3.32(a), along with the rosette transform of the bottom row of Figure 3.29 in (b).

As Figure 3.29 shows, when the Najm tiling is turned into a design by putting rosettes inside all the regular polygons and using inference elsewhere, the design is similar to the one obtained from the related Hankin tiling through inference alone. When the tilings are superimposed, we see that a regular polygon will generally be converted into a similar regular polygon surrounded by a ring of irregular shapes, mostly pentagons. Inference will produce a large star inside the regular polygon and five-pointed stars in the surrounding pentagons. Adjacent five-pointed stars conspire with the large inner star to create the hexagons characteristic of rosettes. The rosette transform is motivated by (and named after) the goal of making the pentagons as close as possible to regular, producing rosettes that are nearly ideal.

Although the rosette transform is given above for the Euclidean plane, it could easily be adapted to absolute geometry, thus providing a means of producing more elaborate star patterns in the hyperbolic plane and on the sphere. The only change would be a generalization of the formula for scaling regular polygons to absolute geometry. The presentation here is confined to the Euclidean plane for simplicity and to highlight the relationship with Bonner's work.

Given the seeming equivalence between these two families of tilings, why not simply choose either Najm tilings or Hankin tilings and develop star patterns based on them alone? It turns out that each approach can handle cases that the other cannot. There is a tradeoff in choosing one kind of tiling over the other. The simpler Najm tilings rely on the design elements, which allow for more direct control over the appearance of complex motifs. Design elements can occasionally produce designs that are aesthetically superior to the design related through the corresponding Hankin tiling. On the other hand, the more complex Hankin tilings need only a single inference algorithm to produce a complex design.

Consider the Archimedean tiling (4.8^2). Running the Taprats algorithm on this tiling, using eightfold rosettes for the octagons, produces the design shown in Figure 3.33(b). If instead we first compute the rosette transform of the tiling, and run Hankin's method on the result, we get the very similar design in Figure 3.33(d). Although these designs have the same general structure, the former can be considered superior because the rosette hexagons all have the same size and line up cleanly. Bonner handles the problem with rosette hexagons in (d) by moving the contact positions on the pentagons away from the edge centers. This sort of adjustment proves to be necessary when

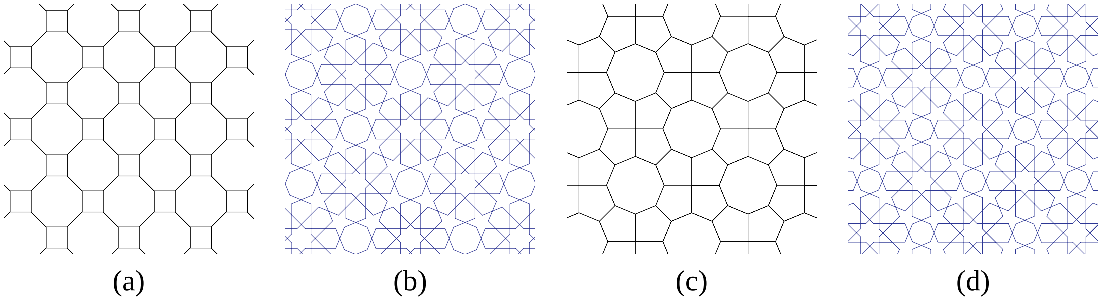


Figure 3.33 An example where the simpler tilings of the Taprats method, combined with more complex design elements, can produce a better overall design than the more complex tilings of Hankin's method. The tiling in (a) produces the design in (b) by placing rosettes in the octagons. The corresponding Hankin tiling in (c) produces a similar design in (d), but this new design is unsatisfactory because the rosette hexagons have different sizes.

applying Hankin's method to some tilings, but it is not clear how such a correction can be formalized and applied generally. In cases like this one, it seems simpler to use the equivalent Najm tilings and the parameterized design elements.

On the other hand, the Hankin tilings can be useful for expressing designs featuring stars with unusual numbers of points. Bonner exhibits several such designs including unlikely combinations like a periodic design with 11- and 13-pointed stars, reproduced in Figure 3.34. These remarkable designs work because the extra layer of smaller tiles such as irregular pentagons can absorb the error in attempting to reconcile the incompatible angles of these regular polygons. The template tiling that produces the design with 11- and 13-pointed stars is not the rosette transform of any Najm tiling. The more complex tilings are therefore essential in some cases for creating designs that are impossible to express using Najm. Of course, the rest of the system can operate on either sort of tiling — design elements can still be placed in the regular polygons if desired, and the inference algorithm can be run everywhere else.

3.9 CAD applications

I now diverge temporarily from the development of new families of star patterns to discuss some of the CAD and manufacturing applications of the patterns presented so far. In the next section, I will

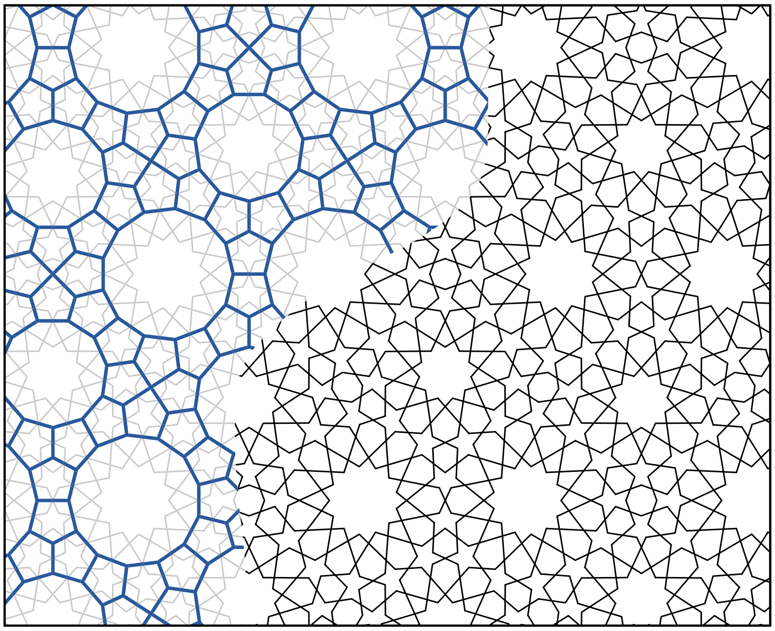
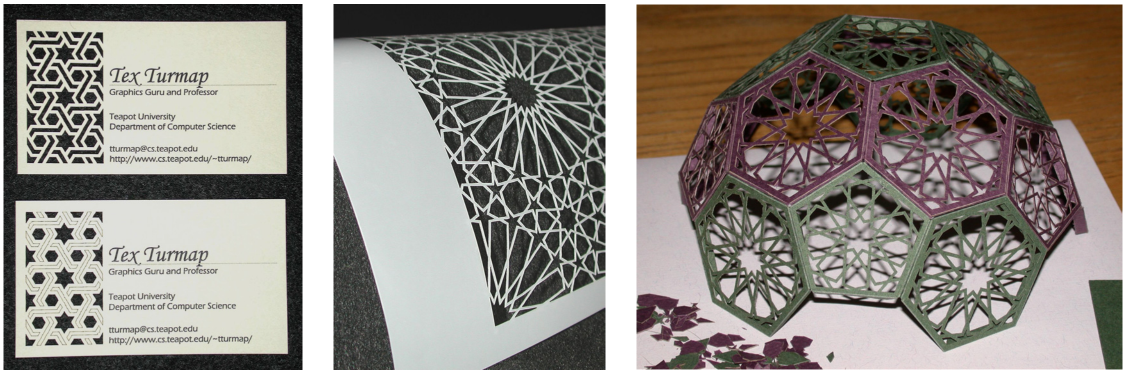


Figure 3.34 An unusual star pattern, reproduced from Bonner's manuscript, featuring 11- and 13-pointed stars.

return to the creation of novel designs one last time, creating star patterns based on aperiodic tilings.

The primary historical use of star patterns is as architectural ornamentation. Star patterns can be found on walls, floors, and ceilings of buildings all across Europe, Africa, and Asia. Traditionally, they are executed in a variety of media: assembled from small terracotta tiles (a style known as "Zellij"), carved as a bas-relief into stone or wood, built from wooden slats into latticework, or simply painted onto a surface. All of these methods are highly costly and laborious, which might account for a modern decline in the application of star patterns. Sometimes a compromise is reached, where a square translational unit is extracted from a pattern and stamped onto large square tiles. Though pretty, this approach has only a small fraction of the visual impact of Zellij, where each coloured region is fabricated as a separate tile.

The patterns created by the programs in this chapter cry out to be made into real world artifacts. The good news is that the capability of computer graphics to invent and visualize shapes is now being matched by an incredible array of devices that can manufacture those shapes in the real world. These CAD devices hold great promise for a revitalization of ornament in everyday architecture. We


Tex Turmap
Graphics Guru and Professor
Teapot University
Department of Computer Science
tturmap@cs.teapot.edu
http://www.cs.teapot.edu/~tturmap/
Tex Turmap
Graphics Guru and Professor
Teapot University
Department of Computer Science
tturmap@cs.teapot.edu
http://www.cs.teapot.edu/~tturmap/

(a) (b) (c)

Figure 3.35 Examples of laser-cut star patterns.

have become accustomed to the featureless grey urban towers of the previous century. With tools that design and fabricate ornamentation quickly and at a reasonable cost, perhaps we can reawaken our primordial urge to decorate.

Islamic star patterns are ideally suited to experimentation with computer-aided manufacturing. They are geometric rather than image-based, meaning that precise information is available to guide the paths of cutting tools. Though tied to Islamic culture, star patterns are appreciated around the world for their harmony and simplicity. Finally, they have been used extensively in architecture already, and so we have a large library of real-world artifacts to guide the aesthetic of an automated approach.

I have used star patterns as a test case in experimenting with a variety of manufacturing processes and media. The results of some of those experiments are presented here.

One general class of devices, similar to plotters, are those in which a computer-controlled tool moves over a plane, selectively cutting or scoring parts of the material. These devices can cut away the lines that make up a star pattern, resulting in a screen resembling a traditional piece of latticework.

For thin materials like paper, the simplest approach is to use a CO_2 laser cutter. This machine is quite effective at cutting paper and mylar, as shown in Figure 3.35. I have used the laser cutter to create prototype business cards, which have proven to be very popular but time consuming to produce. The laser cutter I used was not practical for mass production, though there are larger industrial systems that can cut objects on an assembly line. One might also imagine cutting star patterns using metal die cutting; I have not attempted this approach because of the prohibitive cost of die fabrication.

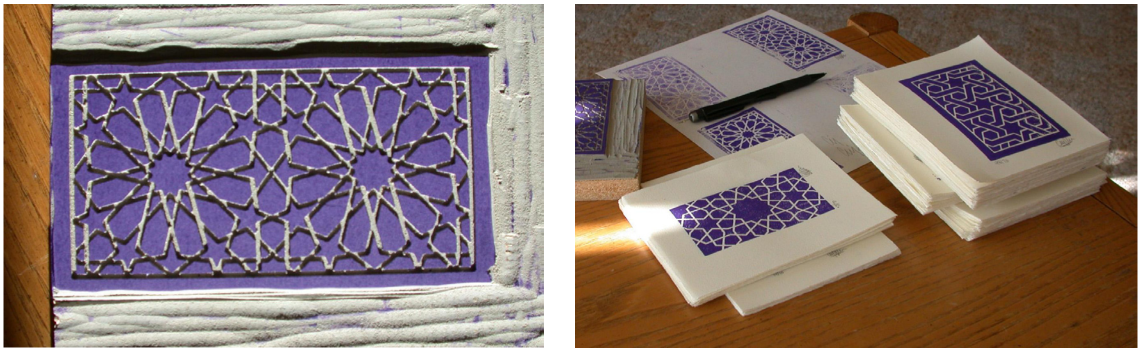
CNC milling machines have been around for some time. Here, the cutting tool is a drill bit, allowing the milling machine to automate many of the tasks performed by hand-operated power tools. Figure 3.36 shows an example of the use of the milling machine. Using a router bit, I cut star patterns into blocks used for linocut printing. The blocks were then hand printed onto paper. The result combines the rigidity of computer-generated geometry with the handcrafted quality of the printing process.

It might also be possible to address the problem of fabricating metal dies for paper cutting by machining a die using a milling machine.

At the high end of the plotting devices is the waterjet cutter. These machines cut by directing a high-pressure stream of grit-impregnated water at the material. They have an incredible range, able to cut through materials ranging in toughness from foam to titanium, even able to handle delicate materials like glass. As part of a collaboration on an architectural project, I have had access to a waterjet cutter, and have fabricated several prototype artifacts in different media. Examples are shown in Figure 3.37.

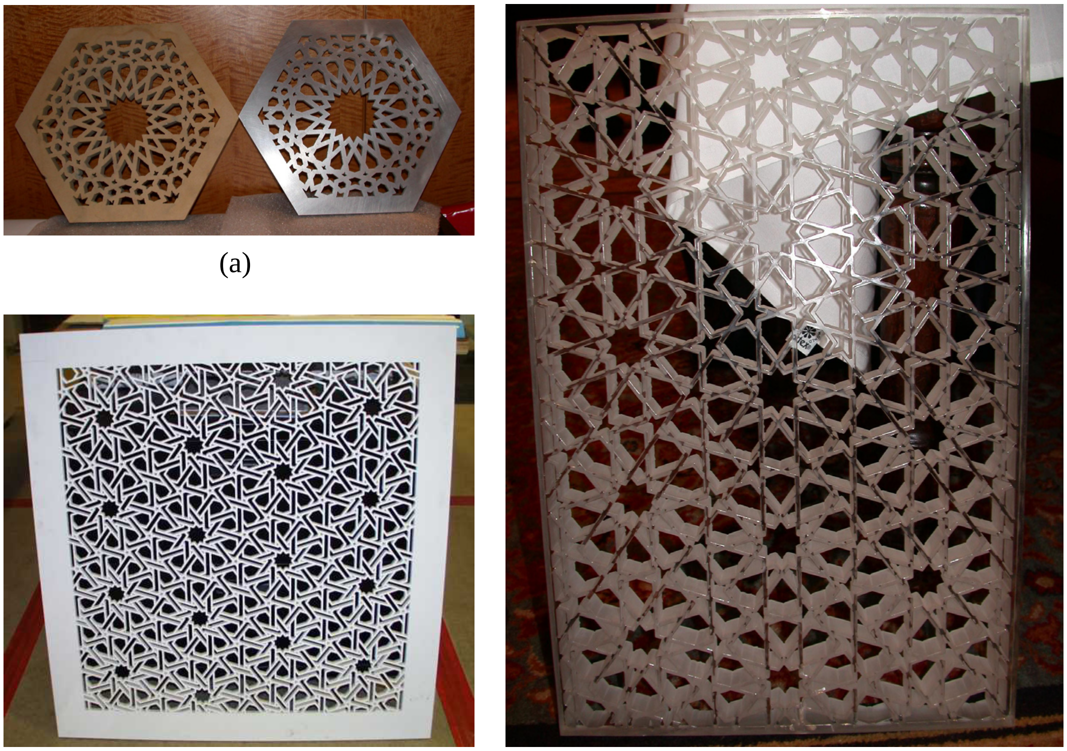
Another exciting class of manufacturing devices are rapid prototyping machines. These tools use a variety of processes to build any watertight three-dimensional model, usually by slicing the model into layers and fabricating one layer at a time from a synthetic material. Two common processes are fused deposition modeling, where a nozzle extrudes a stream of liquid ABS plastic to print each layer, and the ZCorp process, where a stream of liquid selectively bonds a powder substrate.

Rapid prototyping machines are not especially practical for executing Euclidean patterns; these are much more effectively manufactured using any of the plotting machines described above. But they are excellent at fabricating spherical patterns, which otherwise would be very difficult to machine or carve accurately by hand. Figure 3.38 shows some manufactured spherical patterns. Ideally, these models could be used to create molds for metal sculptures; Grossman has used rapid prototyping to fabricate models suitable for bronze casting via a lost-wax method [63].



(a) (b)

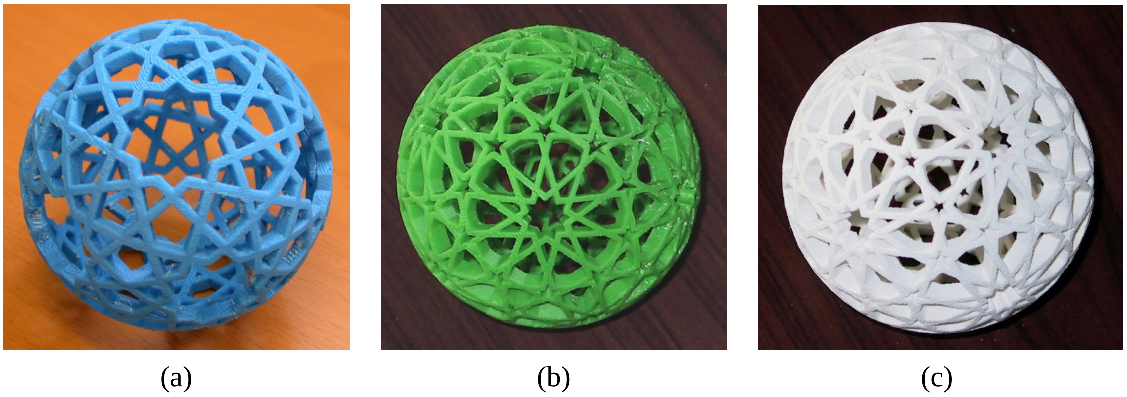
Figure 3.36 Examples of star patterns created using a CNC milling machine. The milling machine was used to route star patterns into linoleum printing blocks, from which hand prints were made.



(a)

(b) (c)

Figure 3.37 Examples of star patterns cut using a waterjet cutter. The pieces are cut from particleboard and steel in (a), from MDF (a versatile wood composite) in (b), and from plexiglass in (c).



(a) (b) (c)

Figure 3.38 Examples of star patterns fabricated using rapid prototyping tools. The first two were built using fused deposition modelling, and the last using ZCorp. Models (b) and (c) are based on the same design.

3.10 Nonperiodic star patterns

As a final excursion in the world of Islamic art, I would like to consider the construction of nonperiodic star patterns in the Euclidean plane. Although it may seem as if such an idea would have to be an exclusively modern one, there are in fact many historical examples of Islamic ornament based on a nonperiodic arrangement of elements. For example, the placement of Muqarnas (a system of ornamental corbelling, usually installed under domes or arches) is typically guided by a symmetric but not periodic patch of squares and $45°$ rhombs [19, Page 289]. Many domes are also decorated with star patterns that cannot be extended periodically.

These wonderful historical artifacts do not imply that Islamic artisans understood the mathematics of aperiodic tilings. With some experimentation, it is easy to cover a large region of the plane with a radially symmetric arragement of squares and $45°$ rhombs or Penrose rhombs. The deeper fact that these shapes are related to inherently aperiodic prototile sets need not play a role in the experimentation process.

On the other hand, it is reasonable to exploit the modern theory of quasiperiodic tilings in constructing star patterns, provided the results do not stray too far from the aesthetic of Islamic geometric art. Here, I explore several ways of deriving novel quasiperiodic template tilings, and the star patterns that can be generated from them.

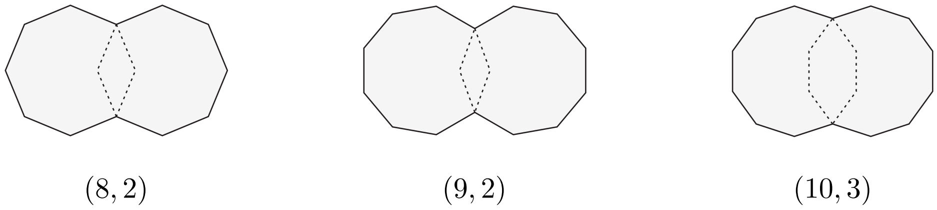
3.10.1 Lattice projection tilings

One well-known method for creating quasiperiodic tilings is the “lattice projection method,” best demonstrated by the Geometry Center’s online **Quasitiler** application [18]. In this method, an n-dimensional integer lattice is sliced through by a carefully rotated two-dimensional plane. Edges of the lattice that touch the plane are projected orthogonally onto it. The network that ends up inscribed on the plane is a quasiperiodic tiling by rhombs.

By themselves, these rhombic tilings are not suitable for use as template tilings. Rather, they can be seen as a guide for the placement of regular polygons in the formation of a new tiling. The rhombs act like fundamental regions for the tilings described in Section 3.6.1. When run on an n-dimensional lattice, the lattice projection method will yield only rhombs with smaller interior angles in the set $\{\pi/n, 2\pi/n, \ldots, \lfloor n/2 \rfloor \pi/n\}$. For example, the 5-dimensional lattice produces rhombs with interior angles of $\pi/5$ and $2\pi/5$, which are none other than the rhombs of Penrose’s aperiodic set $P3$. Castéra has demonstrated a star pattern based on Penrose rhombs that is very similar to the ones developed here [20].

As with a tiling construction based on fundamental regions of symmetry groups, the structure of angles around tiling vertices in a lattice projection tiling permits regular polygons to be placed at those vertices in a principled way. The placement of regular polygons might leave behind holes that are then filled with additional tiles. In a rhombic tiling derived from an n-dimensional lattice, we can, for every integer $k \geq 1$, place a regular $2kn$-gon at each tiling vertex and inflate them (in the same way as was done in Section 3.6.1) until they meet.

Inspired by this observation, I consider the simplest case of placing a regular $2n$-gon at every rhomb vertex. Each regular polygon is oriented so that it has an edge midpoint on the two rhomb edges that are adjacent to the vertex. They are scaled so that they meet at the center of every rhomb edge (as in the “simultaneous inflation of two polygons” in Section 3.6.1). As with the other families of tilings given in this chapter, this process will leave behind holes, which are filled as needed with new tiles. The resulting tiling can then be turned into a design for a star pattern by applying the usual inference algorithm and design elements.



$(8,2)$ $(9,2)$ $(10,3)$

Figure 3.39 Examples of (n,c)-monsters for $n=8$, $n=9$, and $n=10$.

Unfortunately, there is a slight problem with this technique. When applied to the thinnest rhomb in the tiling, the placed regular polygons will overlap in the middle of the tile, as shown in Figure 3.41(a). If the motifs for the two overlapping polygons are simply superimposed, the result is not aesthetically pleasing. Instead, we must fuse these overlapping regular polygons into a larger entity that astronomer Johannes Kepler called a *monster* (at the end of this section, I will discuss a monstrous tiling due to Kepler). But then we encounter another problem: the inference algorithm performs poorly on monsters. It tends to produce large empty areas that ought to be filled with additional geometry. In order to proceed with Quasitiler-based star patterns, we need a taxonomy of monsters and the motifs that can be associated with them.

For every $n > 6$ and $2 \leq c \leq \lceil n/2 \rceil - 2$, define the (n,c)-*monster* as the union of a pair of regular n-gons arranged so that exactly c full edges of each polygon lie inside the other. Note that the cases $c=0$ and $c=1$ are meaningful — they correspond to two polygons that share a single vertex and a single edge, respectively. But these cases do not need to be dealt with specially when designing motifs, and so we ignore them for the rest of this discussion. Figure 3.39 shows some examples of monsters for $n=8$, $n=9$, and $n=10$.

Clearly, a large part of a monster looks like parts of regular polygons, and so we should strive as much as possible to place parts of star-like motifs in those regions. It is only in the overlapping area that the star motifs interact and need to be modified.

Monsters are somewhat special, in that the motif chosen to deal with the region where the polygons overlap is usually specially tailored to that situation. In some cases, a solution will generalize to other values of n, c, and the desired contact angle. In his manuscript, Bonner provides motifs for the $(10,2)$-monster under various choices of contact angle. For the contact angle $2\pi/5$, Castera

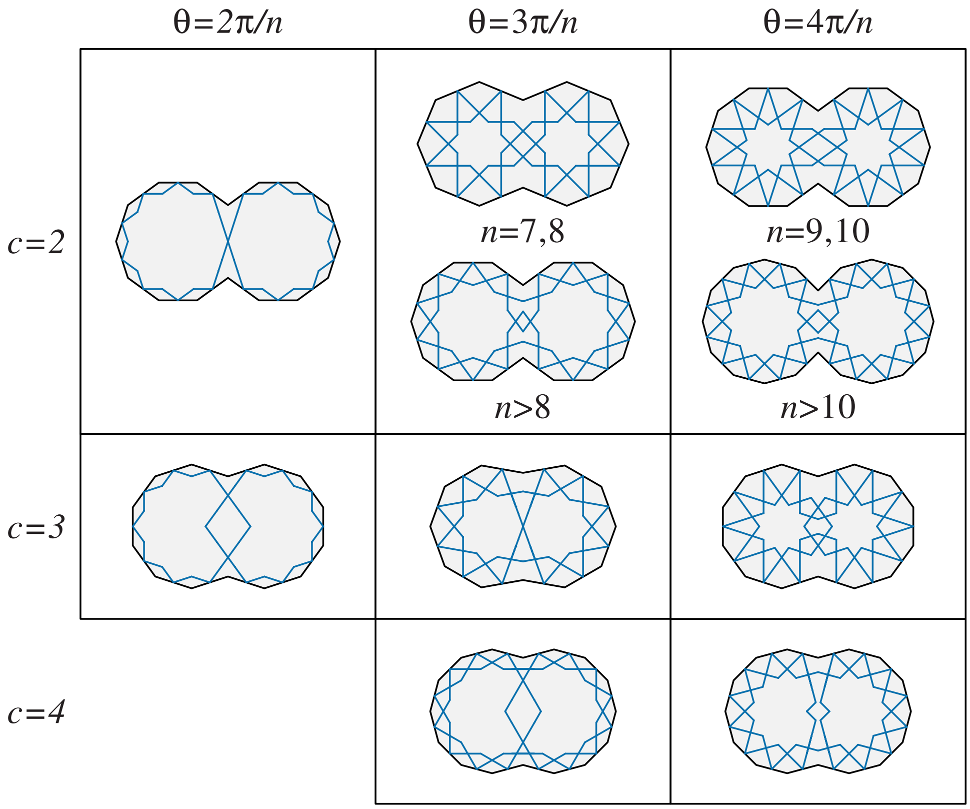


Figure 3.40 A menagerie of monsters and their motifs. This table shows a collection of (n, c)-monster motifs useful for $n \leq 12$. Each row depicts monsters with a different value of c. The columns depict varying contact angles (I consider only multiples of π/n). Except where otherwise noted, the entry given in a cell applies for all legal values of $n \leq 12$.

provides an alternate solution to Bonner's [20] that is somewhat less visually pleasing. Inspired by their efforts, I present in Figure 3.40 a small menagerie of motifs for some of the more frequently occuring monsters. I concentrate on values of n up to 12 and contact angles that are multiples of π/n (*i.e.*, contact angles corresponding to integral values of d in the traditional $\{n/d\}$ star notation).

This library of monster motifs can now be used to create a variety of quasiperiodic designs. Some examples are shown in Figure 3.42. In this case I have not written a tool to deal with these tilings directly. Instead, I rely on Taprats to draw and edit the individual motifs, and then import the motifs into Adobe Illustrator for decoration and assembly. I simply invoke Quasitiler to obtain a quasiperiodic patch of rhombs that then guide the placement of motifs. There is no reason why

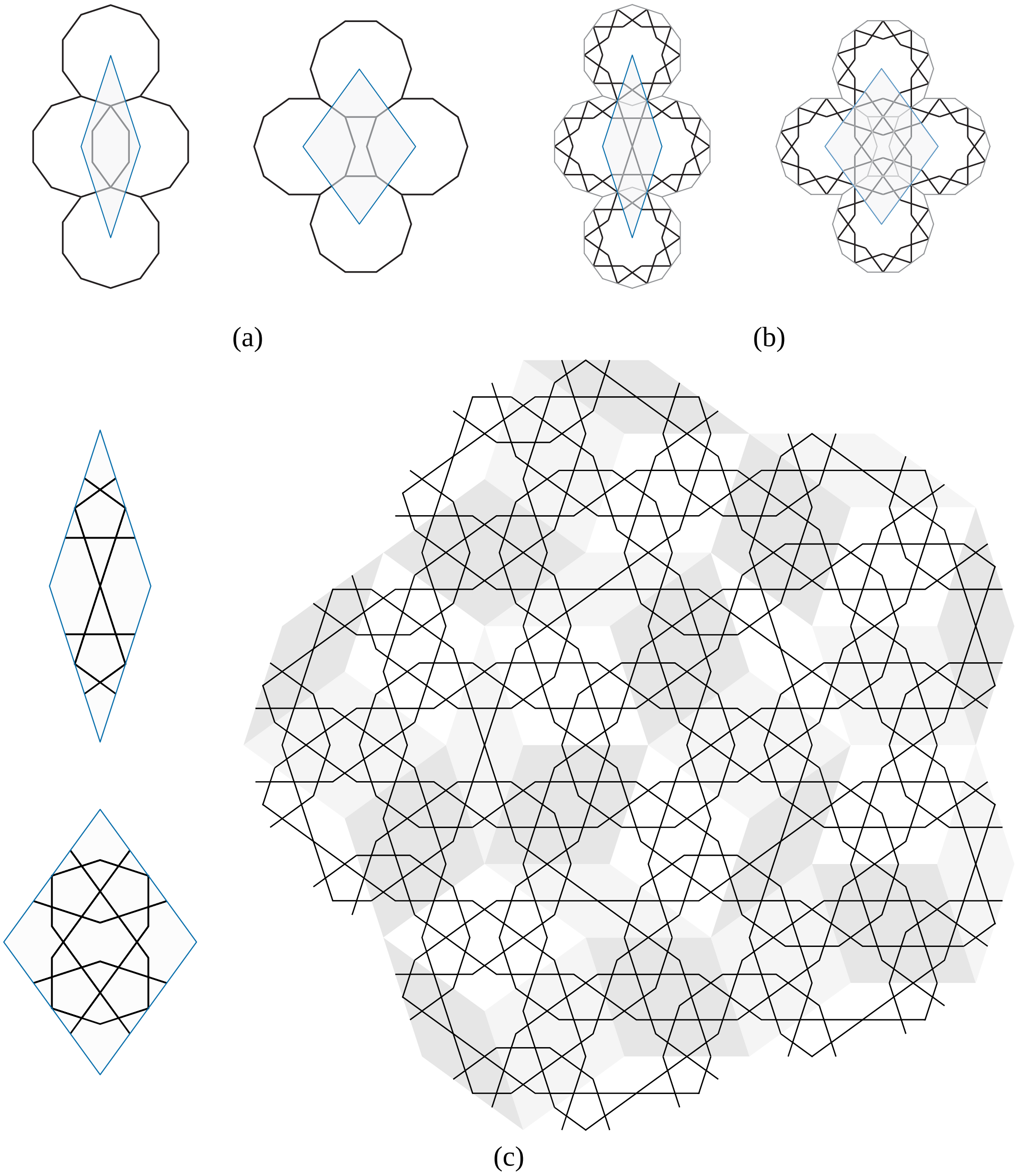
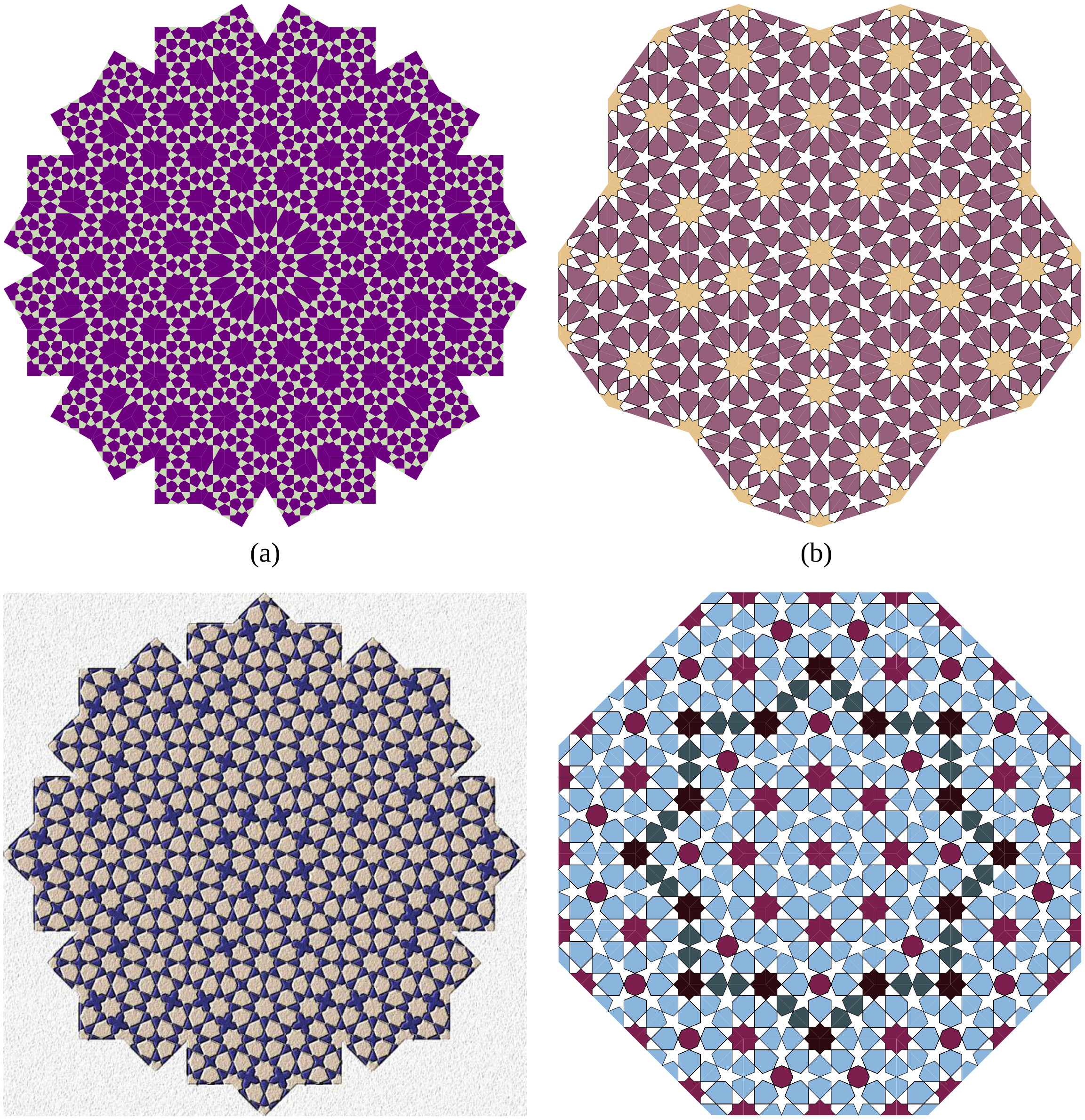


Figure 3.41 The development of design fragments for a Quasitiler-based Islamic star pattern. In (a), the two fivefold rhombs (identical to the Penrose rhombs) are shown superimposed over decagons centered at their vertices. In (c), the template tiles are given motifs, with the $(10, 3)$-monster receiving the motifs shown in Figure 3.40. Finally, in (c), the collected motifs are restricted to the rhombs and assembled into a star pattern. The rhombs are randomly coloured in the final pattern to show the structure of the underlying quasiperiodic tiling.



(a)

(b)

(c)

(d)

Figure 3.42 Examples of Quasitiler-based Islamic star patterns. The examples in (a) and (b) are based on 6- and 5-dimensional lattices, respectively. The examples in (c) and (d) are both based on 4-dimensional lattices. The designs in (b) and (d) are based on rosette transforms of tilings (as defined in Section 3.8), leading to the stronger presence of rosettes.

these individual steps could not be combined into a single program; the manually-assembled results here are presented primarily as a proof of concept of the approach.

3.10.2 Kepler's Aa tiling

I close this journey through Islamic star patterns with an example based on a single unusual and very elegant tiling.

The frontispiece of *Tilings and Patterns* reproduces a set of tilings from Kepler's seventeeth century *Harmonice Mundi*. Most of Kepler's tilings are well known, or at least easily understood. But one tiling, labeled "Aa," does not succumb to analysis so readily. Kepler seems to be attempting to fill the plane exclusively with shapes possessing fivefold symmetry, though his scheme eventually produces what we can recognize as $(10, 2)$-monsters (it was in fact Kepler who suggested the term "monster" for these fused polygons). Interestingly, it is still unknown whether such a tiling exists (with uniformly bounded tiles) [30], or indeed whether a tiling exists with c_n-symmetric tiles for $n = 5$ or $n \geq 7$ [28, Page 93].

The difficulty with Kepler's drawing of his Aa tiling is that it is far from obvious how the patch of tiles given might be extended to cover the whole plane in an orderly manner. Indeed, it is not even clear that Kepler had a specific extension in mind. Nevertheless, one possible approach was discovered by Dessecker, Eberhart and others [68, Page 89]. Their ingenious construction fills a 36° rhomb with tiles from Aa. These rhombs can then be laid out in two different ways, each preserving the consistency of the contained tiles. Rhombs can fit together edge-to-edge, or offset by the golden ratio of the rhomb edge length. The two possible configurations are exploited by filling the plane with a radially symmetric arrangement of rhombs that overlap around a pentacle at the origin. When the tiles are assembled according to this arrangement of rhombs, the result is a truly remarkable nonperiodic tiling of the plane with global fivefold symmetry. Figure 3.43 shows the steps in the construction of the Aa tiling.

Kepler's tiling is made up of regular pentagons, regular decagons, $(10, 2)$-monsters, and pentacles (stars of the form $\{5/2\}1$). Except for the pentacles, all of these tiles are now familiar ground, and motifs can be given for them easily. The one remaining obstacle in building a star pattern based on Aa is to figure out what to do with the pentacles.

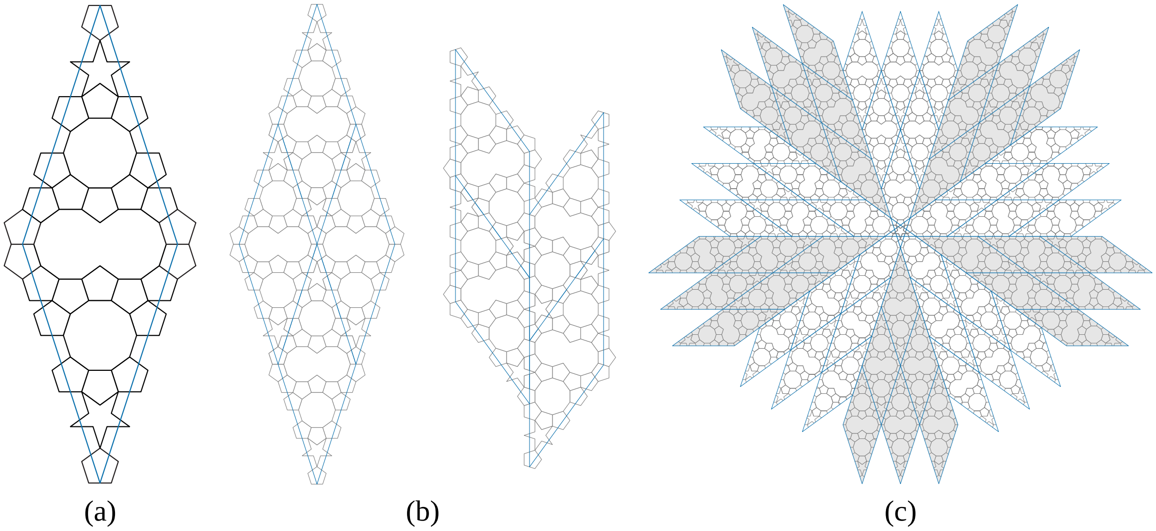


Figure 3.43 The construction of Kepler's Aa tiling, reproduced from Grünbaum and Shephard [68, Figure 2.5.10]. The arrangement of tiles inside one rhomb is shown in (a). Two distinct ways to place rhombs are shown in (b). In (c), both of these ways are used to cover the plane with rhombs, five of which overlap. When the smaller tiles are placed according to this arrangement of rhombs, a consistent aperiodic tiling of the plane is produced.

The pentacle is not particularly well-suited to the task of being a template tile. The inference algorithm produces a motif, but that motif is rather unsuccessful, leaving a large empty region that cries out for more geometry. We can observe, however, that in Kepler's tiling the pentacle is always surrounded by a ring of five pentagons. We can modify these configurations of six tiles to permit more satisfying motifs in the regions they cover. Two possible modifications are shown in Figure 3.44, along with motifs they can be used to generate. In the second of those cases, the inference algorithm is still inadequate in expressing a suitable motif; the motif given was developed by hand, inspired by consideration of traditional solutions in similar contexts.

These slightly modified versions of the original tiling can now serve as template tilings for star patterns. Two examples are shown in Figure 3.45. These star patterns are particularly satisfying as they bring together the insights of mathematicians from across centuries of time and vastly different cultures. They seem to reach close to the heart of the geometric aesthetic.

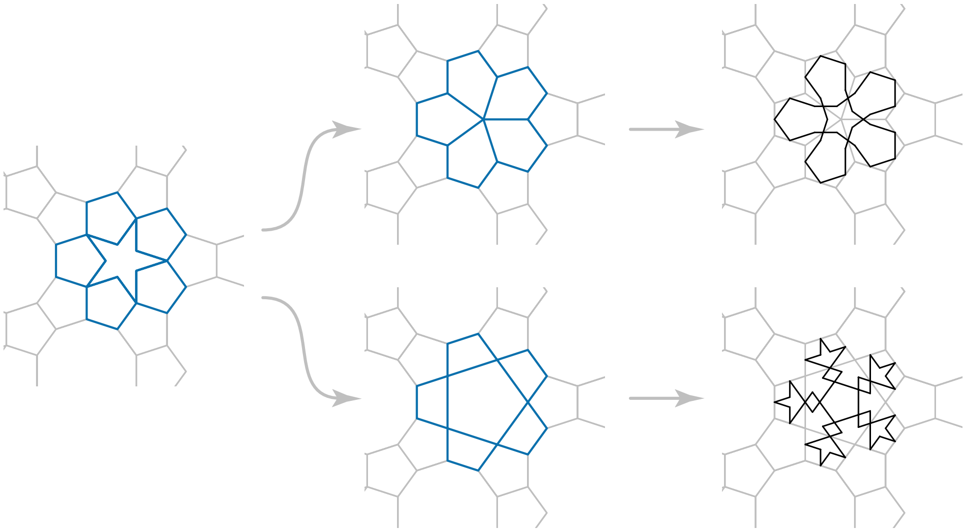


Figure 3.44 Proposed modifications to the region surrouding the pentacle in Kepler's Aa tiling that permit better inference of motifs.

3.11 Future work

This chapter has explored the riddle of Islamic star patterns. I propose one possible solution to that riddle, in the form of a sequence of algorithms and constructions that produce star patterns. Moving beyond the work presented here, there are still tremendous opportunities for future work centered on the use of computers in creating star patterns. I conclude this chapter by discussing some of the most exciting future directions.

3.11.1 Better decoration tools

The decoration tools provided by Taprats and 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 points out that for certain classes of star pattern, there is a "correct" choice of band width for the outlined and interlaced decoration styles [19]. The best width achieves a pleasing visual balance between the thickened bands and the spaces between them. Some additional analysis of historical artifacts might lead to formulae for these band widths.

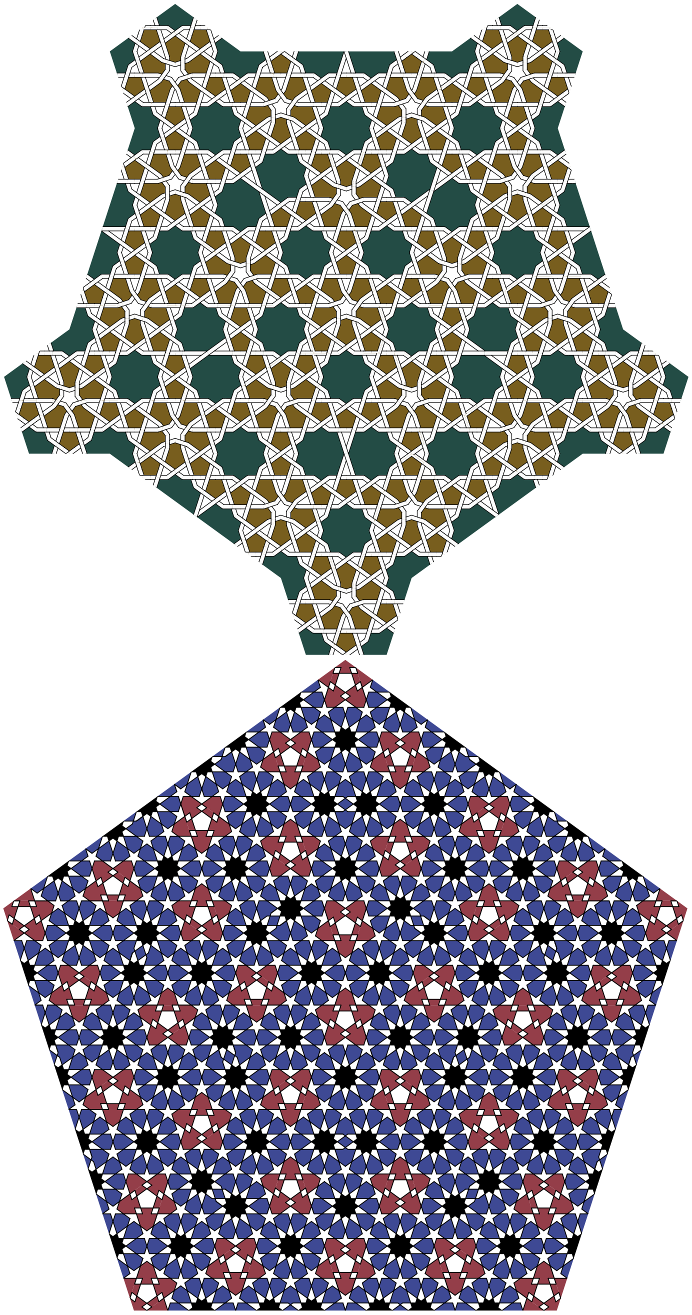


Figure 3.45 Two examples of star patterns based on Kepler’s Aa tiling.

The default behaviour for the filled decoration style is to 2-colour the design. More than two colours are almost always used in traditional renderings of star patterns, and there are conventions that govern the choice of colours and their distribution over the regions of the design. Some automation could be applied to the colouring of regions by encoding these conventions in software. The automation would rely on the ability to "parse" the regions in a design into well-known categories.

3.11.2 The use of optimization

There are cases where the simple inference algorithm of Section 3.4 fails to discover what is historically the "correct" motif for a template tile. In his manuscript, Bonner discusses these situations as they arise, pointing out special cases where a contact angle must be changed or a contact position moved slightly away from the center of a tile edge. 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.

The inadequacies of the inference algorithm may be surmountable through the use of optimization. Given a template tiling, we might imagine using the algorithms presented in this chapter to construct an initial design, one that at least has the correct topology. An optimization procedure, with an aesthetic evaluation as its objective function, might then be used to improve the design, seeking a configuration that has better visual balance. Bonner has suggested that rigorous aesthetic goals for star patterns could be derived from experience with historical examples.[4] Alternatively, the aesthetic objective could be based on measuring the visual appeal of the design according to principles from Gestalt psychology.

3.11.3 Moroccan star patterns

This chapter presents techniques for constructing a wide class of designs, but there is a completely separate historical tradition for star pattern construction that came out of Morocco and Spain. The majority of examples in Castéra's book [19] are not intended to be periodic patterns. They are presented as finite designs centered around a single, large star (with as many as 96 points). In some cases, the designs could be extended into periodic patterns, but the repeated large stars would be a

[4] Jay Bonner, personal communication.

distraction rather than a focal point. These finite designs are often installed in fountains, doors, and other clearly delimited planar regions.

As was mentioned in Section 3.2, Castéra's method is based fundamentally on the placement of a skeleton made up of stars and safts. The skeleton delineates the broad structure of the design. Then, he uses a kind of inference step to fill the regions outlined by the skeleton. His inference step seems to rely on a great deal of intuition, although some computer assistance may be possible. In particular, he always aims to create inferred regions in the shapes of well-known tiles from the Zellij style. Inference may then be regarded as a kind of puzzle assembly problem (where each piece may be used multiple times).

The more remarkable of Castéra's examples are those featuring stars with many points. An interesting challenge would be to develop a system that draws such designs automatically. This is a difficult problem; the designs are highly specialized, relying on a great deal of trickery to interface a large star with the lower-order symmetry of the area that surrounds it. Perhaps optimization could be used here to carry out this trickery with a minimum amount of disturbance to the surrounding geometry.

The quasiperiodic technique of Section 3.10.1 seems like another fruitful approach to producing large stars, as the rhomb tilings produced by Quasitiler can have a global center of high rotational symmetry. A region of the design around the rotational center could be replaced with a single large star or rosette. Again, optimization might better integrate the star into the rest of the design. Alternatively, a different means of producing quasiperiodic tilings, such as the overlay-dual described by Stampfli [128] and Zongker [142], might play a role in creating Moroccan-style patterns.

3.11.4 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 $\{8, 12\}$, $\{9, 12\}$, and $\{9, 18\}$ follow immediately from the tiling notation of Section 3.6 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. Many designs containing rosettes, for instance, might also be seen to contain distorted 5-pointed stars. A more dramatic example is the near-miss Altair tiling given in Figure 3.3, inducing a star pattern with 4-, 5-, 6-, 7-, and 8-pointed stars.

The development of attractive patterns based on unusual sets of star orders is a fascinating challenge that balances mathematics with an intuition for covering up distortion. Historical artisans seemed to delight in discovering such designs; one wonderful example shown by Bonner contains 11- and 13-pointed stars (see Figure 3.34). It is hard to imagine that a general algorithm underlies the creation of these unusual designs, but it would be interesting to search for some heuristics that could intelligently guide the placement of star centers and invent plausible hole-filling motifs to take up the space between the stars.