Computer Graphics and Geometric Ornamental Design

Craig S. Kaplan

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Washington

2002

University of Washington

Graduate School

This is to certify that I have examined this copy of a doctoral dissertation by

Craig S. Kaplan

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Chair of Supervisory Committee:

_____

David H. Salesin

Reading Committee:

_____

Brian Curless

_____

Branko Grünbaum

_____

David H. Salesin

Date: _____

University of Washington

Abstract

Computer Graphics and Geometric Ornamental Design

by Craig S. Kaplan

Chair of Supervisory Committee:

Professor David H. Salesin
Computer Science & Engineering

Throughout history, geometric patterns have formed an important part of art and ornamental design. Today we have unprecedented ability to understand ornamental styles of the past, to recreate traditional designs, and to innovate with new interpretations of old styles and with new styles altogether.

The power to further the study and practice of ornament stems from three sources. We have new mathematical tools: a modern conception of geometry that enables us to describe with precision what designers of the past could only hint at. We have new algorithmic tools: computers and the abstract mathematical processing they enable allow us to perform calculations that were intractable in previous generations. Finally, we have technological tools: manufacturing devices that can turn a synthetic description provided by a computer into a real-world artifact. Taken together, these three sets of tools provide new opportunities for the application of computers to the analysis and creation of ornament.

In this dissertation, I present my research in the area of computer-generated geometric art and ornament. I focus on two projects in particular. First I develop a collection of tools and methods for producing traditional Islamic star patterns. Then I examine the tesselations of M. C. Escher, developing an "Escherization" algorithm that can derive novel Escher-like tesselations of the plane from arbitrary user-supplied shapes. Throughout, I show how modern mathematics, algorithms, and technology can be applied to the study of these ornamental styles.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGMENTS

I have never once regretted my decision to come to the University of Washington. It has been

an honour to work in the department's friendly, supportive, and cooperative environment. Thanks in general to all my friends among the faculty, staff, and students, and in particular to Doug Zongker for more lunch conversation than I could possibly count.

Thank you to my parents for their unflagging dedication and absolute confidence in me, and to my brother, grandmother, and extended family for their perpetual care and support.

Finally, it is impossible to believe that I might have completed this work without the support, advice, generosity, devotion, and occasional skepticism of my wife Nathalie. To her, and to my daughter Zoë, I owe my acknowledgment, my dedication, and my love.

Chapter 1

INTRODUCTION

All the majesty of a city landscape
All the soaring days of our lives
All the concrete dreams in my mind's eye
All the joy I see thru' these architect's eyes.

— David Bowie

The creation of ornament is an ancient human endeavour. We have been decorating our objects, our buildings, and ourselves throughout all of history and back into prehistory. From the moment humans began to build objects of any permanence, they decorated them with patterns and textures, proclaiming beyond any doubt that the object was an artifact: a product of human workmanship. The primeval urge to decorate is bound up with the human condition.

As we evolved, so did our talents and technology for ornamentation. The history of ornament is a reflection of human history as a whole; an artifact's decoration, or lack thereof, ties it to a particular place, time, culture, and attitude.

In the last century, we have developed mathematical tools that let us peer into the past and analyze historical sources of ornament with unprecedented clarity. Even when these modern tools bear little or no resemblance to the techniques originally used to create designs, they have an undeniable explanatory power. We can then reverse the analysis process, using our newfound understanding to drive the synthesis of new designs.

Even more recently, we have crossed a threshold where these sophisticated mathematical ideas can be made eminently practical using computer technology. In the past decade, computer graphics has become ubiquitous, affordable, incredibly powerful, and relatively simple to control. The computer has become a commonplace vehicle for virtually unlimited artistic exploration, with little fear of committing unfixable errors or of wasting resources. Interactive tools give the artist instant

feedback on their work; non-interactive programs can solve immense computational problems that would require considerable amounts of hand calculation or vast leaps of intuition.

The goal of this work is to seek out and exploit opportunities where modern mathematical and technological tools can be brought to bear on the analysis and synthesis of ornamental designs. The goal will be achieved by devising mathematical models for various ornamental styles, and turning those models into computer programs that can produce designs within those styles. The complete universe of ornament is obviously extremely broad, constrained only by the limits of human imagination. Therefore, I choose to concentrate here on two particular styles of ornament: Islamic star patterns and the tesselations of M. C. Escher. During these two investigations, I watch for principles and techniques that might be applied more generally to other ornamental styles.

The rest of this chapter lays the groundwork for the explorations to come, discussing the history of ornament and its analysis, and the roles played by psychology, mathematics, and computer science. In Chapter 2, I review the mathematical concepts that underlie this work. Then, the main body of research is presented: Islamic star patterns in Chapter 3, and Escher's tilings in Chapter 4. Finally, in Chapter 5, I conclude and offer ideas for future work in this area.

1.1 The study of ornament

The practice of ornament predates civilization [22]. The scholarly study and criticism of this practice is somewhat more recent, but still goes back at least to Vitruvius in ancient Rome. Gombrich provides a thorough account of the history of writings on ornament in *The Sense of Order* [60], a work that will no doubt become an important part of that history.

What is ornament? To attempt a formal definition seems ill-advised. Any precise definition will omit important classes of ornament through its narrowness, or else grow so broad as to encompass an embarrassing assortment of non-ornamental objects. In the propositions that open Jones's classic *The Grammar of Ornament* [91], we find many comments on the structure and common features of ornament, but no definition. Racinet promises to teach "more by example than by precept [121, Page 13]." Ornament, like art, is hard to pin down, always evading definition on the wings of human ingenuity.

On the other hand, the works of both Racinet and Jones teach very effectively by example. Their

marvelous collections contain a multitude of designs from around the world and throughout history. Based on these collections, and the definitions that have been offered in the past, we may identify some of the more common features of ornament. We will adopt these features not as a definition, but as guidelines to make the analysis of ornament possible here.

- **Superficiality:** Jensen and Conway attribute the appeal of ornament to its "uselessness [90]." They are referring to the fact that ornament is precisely that which does not contribute to an object's function or structure. Anything that is "without use," superficial, or superfluous is an ornamental addition. As they point out, uselessness frees the designer to decorate in any way they choose, without being bound by structural or functional concerns.

- **Two-dimensionality:** Most ornament is a treatment applied to a surface. The surface may bend and twist through space, but the design upon it is fundamentally two-dimensional. A common use of ornament is as a decoration on walls, floors, and ceilings, and so adopting this restriction still leaves open many historical examples for analysis and many opportunities for synthesis.

- **Symmetry:** Symmetry is a structured form of order, balance, or repetition (it will be defined formally in Section 2.2). Speiser, one of the first mathematicians to use symmetry in studying historical ornament, required that all ornament have some degree of symmetry [135, Page 9]. This requirement seems overly strict, as there are forms of repetition that cannot be accounted for by symmetry alone, and there are many examples of ornament that repeat only in a very loose sense. Therefore I use symmetry here to refer more generally to a mathematical theory that accounts for the repetition in a particular style of ornament. At the end of this dissertation, I return to the question of the applicability of formal symmetry theory and discuss alternatives.

The history of ornamentation, particularly in the context of architecture, has been marked by the constant pull of two opposing forces. At one extreme is *horror vacui*, literally "fear of the vacuum." This term has been used to characterize the human desire to adorn every blank wall, to give every surface of a building decoration and texture. Taken to its logical conclusion, *horror vacui* produces the stereotypical Victorian parlour, saturated with ornament. A more appealing

historical example is the *Book of Kells*, an illuminated Celtic manuscript whose pages are intricately ornamented (prompting Gombrich to suggest the more positive *amor infiniti* in place of *horror vacui*).

Opposing the use (and abuse) of ornament wrought by believers in *horror vacui*, we have what Gombrich calls the "cult of restraint." He uses the term to refer to those who reject ornament because of its superficiality, and praise objects that convey their essence without the need to advertise it via decoration.

The most recent revival of the cult of restraint came in the form of the modernist movement in architecture. Its pioneers were architects like Mies van der Rohe and Le Corbusier, as well as Gropius (who founded the Bauhaus in Germany) and the Italian Futurists. They rebelled against an overuse of ornament, and reveled in the beauty of technology and machines that promised to change the world for the better. To the modernists, ornament was tied to an erstwhile philosophy and way of life, and the immediate rejection of ornament was a first step to embracing the new ideals of the twentieth century [90]. Architecture of the period has a distinctly spare, austere style with blank walls and right angles.

Modernism came as a breath of fresh air after a century of stifling ornamental saturation. Unfortunately, many architects who lacked the talent of masters like Mies van der Rohe latched on to the modernist movement as a license to erect buildings in the shapes of giant, featureless concrete boxes. Thus was born yet another backlash, this time a cautious return to *horror vacui* in the form of what Jensen and Conway term *ornamentalism* [90]. Today we see some highly visible buildings that experiment with "uselessness"; a recent example is Seattle's *Experience Music Project*, designed by Frank Gehry. Overall, it seems as if the forces of modernism and ornamentalism are both active in contemporary architecture. I do not propose to sway opinion one way or the other. But if architects and other designers are willing to explore the use of geometric ornament, the work presented here could help them turn their explorations into real artifacts.

1.2   The psychology of ornament

The great majority of ornament exhibits some degree of symmetry. The reason must in part be tied to the practicalities of fabricating ornament. As a simple example, fabrics and wallpapers are

printed from cylindrical templates, so their patterns will necessarily repeat in at least one direction. Looking more at the human experience of ornament, there is also a significant neurological and psychological basis for our appreciation of symmetry. This section discusses some of the reasons why there is an innate human connection between symmetry and ornament.

The science of *Psychoaesthetics* attempts to quantify our aesthetic response to sensory input. Research in psychoaesthetics shows that our aesthetic judgment of a visual stimulus derives from the arousal created and sustained by the experience of exploring and assimilating the stimulus. They test their theories by measuring physical and psychological responses of human subjects to visual stimuli.

Detection of symmetry is built in to the perceptual process at a low level. Experiments with functional brain imagining show that humans can accurately discern symmetric objects in less than one twentieth of a second [132]. The eye is particularly fast and accurate in the detection of objects with vertical mirror symmetry. The common explanation for this bias is that such symmetry might be characteristic of an advancing predator. Rapid perception can take place even across distant parts of the visual field, indicating that a large amount of mental processing is expended in locating symmetry. Furthermore, once symmetry is perceived, it is exploited. By tracking eye fixations during viewing of a scene, Locher and Nodine [106] show that in the presence of symmetry the eye will explore only non-redundant parts of that scene. Once the eye detects a line of vertical mirror symmetry, it goes on to explore only one half of the scene, the other half taken as understood.

In another experiment, Locher and Nodine show that an increase in symmetry is met with a reduction in arousal. When asked to rate appreciation of works of art, subjects rated asymmetric scenes most favourably and symmetric scenes decreasingly favourably as symmetry increased. Psychoaesthetics might help to explain this result; a more highly ordered scene requires less mental processing to assimilate, resulting in less overall engagement. While this result might appear to bode poorly for the effectiveness of symmetric ornament, mitigating factors should be considered. Most importantly, they tested the effect of symmetry by adding mirror symmetries to pre-existing works of abstract art. This wholesale modification might have destroyed other aesthetic properties of the original painting, such as its composition.

On the other hand, the reduction in arousal associated with symmetry might be appropriate for the purposes of ornamental design. In many cases, particularly in an architectural setting, the goal

of ornament is to please the eye without unduly distracting it. Locher and Nodine support this claim, mentioning that as complexity of a scene increases, the rise in arousal "is pleasurable provided the increase is not enough to drive arousal into an upper range which is aversive and unpleasant [106, Page 482]."

Other research supports the correlation between symmetry and perceived goodness. In the limited domain of points in a grid, Howe [85] shows that subjective ratings of goodness correlated precisely with the degree of symmetry present. In a similar domain, Szilagyi and Baird [131] found that subjects preferred to arrange points symmetrically in a grid. In their recent review of the perception of symmetry, Møller and Swaddle simply state that humans find symmetrical objects more aesthetically pleasing than asymmetric objects [113].

Moving from the experimental side of psychology to the cognitive side, the theory of Gestalt psychology might be invoked to explain our positive aesthetic reaction to ornament. Gestalt is concerned with understanding the perceptual grouping we perform at a subconscious level when viewing a scene, and the effect this grouping has on our aesthetic response. Perhaps the most compelling explanation for the attractiveness of symmetric ornament is the "puzzle-solving" aspect of Gestalt. A symmetric pattern invites the viewer into a visual puzzle. We sense the structure on an unconscious level, and struggle to determine the rules underlying that structure. The resolution of that puzzle is a source of psychological satisfaction in the viewer. As Shubnikov and Koptsik say, "The aesthetic effects resulting from the symmetry (or other law of composition) of an object in our opinion lies in the psychic process associated with the *discovery* of its laws." [127, Page 7]

In a philosophical passage, Shubnikov and Koptsik go on to discuss the psychological and sociological effects of specific wallpaper groups [127, Page 155] (the wallpaper groups will be introduced in Section 2.2). In their theory, lines of reflection emphasize stability and rest. A line unimpeded by perpendicular reflections encourages movement. Rotational symmetries are also considered dynamic. For the various wallpaper groups, they give specific applications where ornament with those symmetries might be most appropriate.

We should not attempt to use the evidence presented in this section as a complete justification for the use of symmetry in art and ornament. But these experiments and theories reveal that we do have *some* hard-wired reaction to symmetry, a reaction that affects our perception of the world. This evidence provides us with a partial explanation for the historical importance of symmetry in

ornament, and some confidence in its continued aesthetic value.

1.3   Contributions

This dissertation grew out of an open-ended exploration of the uses of computer graphics in creating geometric ornament. As such, the goals were not always stated at the outset, but were discovered along the way as my ideas developed and my techniques became more powerful. As with the artistic process in general, we cannot aim to achieve a specific goal or inspire a specific aesthetic response. But when some interesting result is found, we can then reflect on the method that produced that result and its applicability to other problems.

Here are the main contributions that this work makes to the greater world of computer graphics and computer science:

- **A model for Islamic star patterns.** The two main themes of this dissertation are presented in Chapters 3 and 4. Each of these central chapters makes a specific, thematic contribution. Chapter 3 develops a sophisticated theory that can account for the geometry of a wide range of historical Islamic star patterns. This theory is used to recreate many traditional examples, and to create novel ones.

- **A model for Escher's tilings.** Another specific contribution is the model in Chapter 4 for describing the tesselations created by M. C. Escher. The model accounts for many of the kinds of tesselations Escher created and culminates in an "Escherization" algorithm that can help an artist design novel Escher-like tesselations from scratch.

- **CAD applications.** Computer-controlled manufacturing devices are becoming ever more flexible and precise. The range of materials that can be manipulated by them is continuing to grow. Many computer scientists and engineers are investigating ways these tools can be used for scientific visualization, machining, and prototyping. I add to the list of applications by demonstrating how computer-generated ornament can be coupled with computer-aided manufacturing to produce architectural and decorative ornament quickly and easily.

- **The geometric aesthetic.** George Hart is a mathematician and sculptor who creates wonderful polyhedral sculptures in various media. He states [76] that his work "invites the viewer to partake of the geometric aesthetic." An aesthetic is a particular theory or philosophy of beauty in art. It is the set of psychological tools that allow someone to appreciate art in a particular genre or style. The geometric aesthetic is therefore a form of beauty derived primarily from the geometry of a piece of art, from its shape and the mathematical relationships among its parts. I believe that the geometric aesthetic extends beyond art to account for a feeling of elegance in mathematics. The same mindset that allows one to appreciate Hart's sculptures accounts for the sublime beauty of what Erdős called a "proof from the book," a truly ingenious and insightful proof [82].

  The work presented here is steeped in the geometric aesthetic, and in part has the goal of creating new examples of geometric art. In this regard, its contributions are intended to take part in the artistic discourse on the geometric aesthetic, to increase interest in it, and hopefully to enrich it with the many results presented here.

1.4 Other work

This section discusses some recent work by others that is generally related to the computer generation of geometric ornament. Chapters 3 and 4 each contain additional discussions of related work limited to their respective problem domains.

1.4.1 Floral ornament

An important precursor to the work in this dissertation is the paper by Wong *et al.* on floral ornament [139]. They provide a modern approach to the analysis and creation of ornament, including a taxonomy by which ornament may be classified and a "field guide" for recognizing the common features of designs. Subsequently, they develop a system capable of elaborating floral designs over finite planar regions.

Their algorithm decomposes the problem of creating floral designs into the specification of a collection of primitive motifs that make up the designs, and the elaboration of those primitives over a given region. The paper is concerned with the elaboration process, and leaves the construction of

suitable motifs to the artist.

Elaboration is handled by a *growth model*, a synthetic method of distributing design elements over a region in an approximately uniform way. Growth is accomplished by applying rules to extend the design from existing motifs into currently empty parts of the region. Beginning with a set of "seeds," the algorithm iteratively applies rules until no more growth is possible. The final design can then be rendered by applying the drawing code associated with each of the motifs.

The value of the work of Wong *et al.* is that their innovations do not come at the expense of tradition. Their approach is clearly respectful of the centuries of deeply-considered thought that preceded the advent of computer graphics. Their algorithms emerge from an understanding of the intent and methods of real ornamentation, and are not developed *ex nihilo* as devices that merely appear consistent with historical examples.

For example, they eschew more traditional botanical growth models such as L-systems. The most compelling reason they give is that L-systems are a powerful tool for modeling real plants, which is exactly what floral ornament is not. There is no reason to believe that a simulation of the biological process of growth should lead to attractive designs. Their growth model represents the artist's process is creating a stylized plant design, not the growth of an actual plant.

Although the approach of Wong *et al.* lists repetition as a principle of ornament, their repetition is very loose and not constrained by global order such as symmetry. Therefore, while their results might be appropriate for an illuminated manuscript (or web page) where the surface to be decorated is small, it might be less successful in an architectural setting. Their repetition without order would deprive the viewer of any global structure to extract from the design. The visual puzzle of non-symmetric ornament is less interesting because there *is* no puzzle, only the incompressible fact of the whole design.

1.4.2 *Fractals and dynamical systems*

The computer has not only been used as a tool for recreating preexisting ornamental styles. Computers have also made possible styles that could not have been conceived of or executed without their capacity for precise computation and brute-force repetition.

Fractals are probably the ornamental form most closely associated with computers. They have

a high degree of order, but little symmetry. The Mandelbrot set has but a single horizontal line of mirror reflection, but such a stunning degree of self-similarity that order is visible at every point and at every scale. The correspondence between parts of the Mandelbrot set is always approximate, creating an engaging visual experience. Many computer scientists continue to research interesting ways to render the Mandelbrot set and fractals like it.

Chaos is closely related to fractal geometry. Field and Golubitsky [52] have created numerous ornamental designs by plotting the attractors of dynamical systems. In particular, they have developed dynamical systems whose attractors have finite or wallpaper symmetry. In their work, we find a true rebirth of ornamental design in the digital age.

1.4.3 Celtic knotwork

The art of the Celts was always non-representational and geometric [89]. With the arrival of Christianity to their region in the middle of the first millennium C.E. came the development of the distinctive knotwork patterns most strongly associated with the Celts. Knotwork designs appear carved into tombstones, etched into personal items, and most prominently rendered in illuminated manuscripts such as the Lindisfarne Gospels and the Book of Kells. A design is formed by collections of ribbons that weave alternately over and other each other as they cross. Often, human and animal forms are intertwined with the knotwork, with ribbons becoming limbs and hair.

Celtic knotwork is the intellectual cousin of the Islamic star patterns to be discussed in Chapter 3. Both can be reduced from a richly decorated rendering to an underlying geometric description. Both are heavy users of interlacing as an aesthetic device. But most intriguing is the fact that in both cases, the historical methods of design are now lost. Research into both Celtic knotwork and Islamic star patterns has at times required the unraveling of historical mysteries.

For Celtic knotwork, one possible solution to the mystery is offered by George Bain [7], who built upon the earlier theories of J. Romilly Allen. Allen suggested that knotwork was derived from a transformation of plaitwork, the simple weave used in basketry. Bain presents a method based on breaking crossings in plaitwork and systematically rejoining the broken ribbons.

Also building on the work of Allen, Cromwell [29] presents a construction method similar to Bain's, based on an arrangement of two dual rectangular grids. Cromwell explores one-dimensional

frieze patterns that appear in Celtic art and shows how the structure of generated designs relates to the arrangement of broken crossings in the underlying plaitwork.

The algorithms of Bain and Cromwell adapt readily to the computer generation of Celtic knotwork. In a series of papers, Glassner describes Bain's method and several significant extensions, creating highly attractive knotwork imagery [56, 57, 58]. Zongker [141] implemented an interactive tool similar to the one presented by Glassner. Other popular treatments of Celtic knots on the internet are given by Mercat [111] and Abbott [3].

In an interesting alternative approach, Browne [17] uses an extended tile-based algorithm to fit Celtic knots to arbitrary outlines (letterforms in his case). The technique works by filling the interior of a region with a tiling whose tiles are as close as possible to squares and equilateral triangles. Using a predefined set of tiles decorated with fragments of Celtic knotwork, he assigns motifs to tiles in such a way that the fragments link up to form a continuous Celtic knotwork design. In some cases, the result bears a strong resemblance to the illuminated letters of the ancient Celtic manuscripts. Browne's approach is certainly not the one used by the original artisans, although the final results are fairly successful.

Chapter 2

MATHEMATICAL BACKGROUND

2.1 Geometry

The formalization of geometry began with the ancient Greeks. They took what had been an *ad hoc* collection of surveying and measuring tools and rebuilt them on top of the bedrock of logic. A remarkable journey began then and continues to the present day. The story is brilliantly told by Greenberg [62] and summarized concisely by Stewart [130, Chapter 5]. Other valuable presentations of Euclidean and non-Euclidean geometry are given by Bonola [14], Coxeter [25], Faber [51], Kay [97], and Martin [109].

In his monumental *Elements*, Euclid attempts to reduce the study of geometry to a minimal number of required assumptions from which all other true statements may be derived. He arrives at five postulates, primitive truths that must be accepted without proof, with the rest of plane geometry following as a reward. Over the years, some of the postulates (particularly the fifth) have drifted to alternate, logically equivalent forms. One statement of Euclidean geometry, adapted from Greenberg [62, Page 14], is as follows:

I. Any two distinct points lie on a uniquely determined line.

II. A segment AB may always be extended by a new segment BE congruent to a given segment CD.

III. Given points O and A, there exists a circle centered at O and having segment OA as a radius.

IV. All right angles are congruent.

V. (*Playfair's postulate*) Given a line l and a point P not on l, there exists a unique line m through P that is parallel to l.

The development of a logical system such as Euclid's geometry is a process of abstraction and distillation. Euclid presented his five postulates as the basis for all of geometry, in the sense that

any other true statement in geometry could be derived from these five using accepted rules of logical deduction. Today, we know that Euclid's postulates are incomplete — they are an insufficient foundation upon which to build all that he wanted to be true. Many mathematicians have since provided revised postulates that preserve the spirit of Euclid's geometry and hold up under the careful scrutiny of modern mathematics. In all cases the core idea remains to distill all possible truths down to a minimal set of intuitive assertions.

There are two other senses in which this distillation must occur in order to make the logical foundation of geometry self-contained. First, the *objects* of discourse must be reduced to a suitable primitive set. The postulates mention only points, lines, segments, circles, and angles. No mention is made of polygons, parabolas, or a multitude of other geometric objects, because all such objects can be defined in terms of the five mentioned in the postulates. Even segments, circles, and angles can be defined in terms of points and lines. Euclid attempts to take this process to the limit, providing definitions for points and lines. However, his definitions are somewhat enigmatic. Today, we know that just as all truth must eventually bottom out in a set of primitive postulates, all identity must reduce to a set of primitive objects. So we reduce plane geometry to two sorts of objects: points and lines. These objects require no definition; as Hilbert famously remarked, geometry should be equally valid if it were phrased in terms of tables, chairs, and beer mugs. The behaviour of these abstract points and lines is determined by the postulates. We keep the names as an evocative reminder of the origins of these objects.

The other chain of definition concerns the relationships between objects. The postulates mention relationships like "lie on," "congruent," and "parallel." Again, the chain of definition must bottom out with some primitive set of relationships from which all others can be constructed. In modern presentations of Euclidean geometry such as Hilbert's [62, Chapter 3], three relationships are given as primitive: *incidence*, *congruence*, and *betweenness*. Incidence determines which points lie on which lines. Congruence determines when two segments or two angles have "the same shape." Betweenness is implicit in the definition of objects like segments (the segment AB is A and B together with the set of points C such that C is between A and B). Again, these relationships do not have any *a priori* definitions; their behaviour is specified and constrained through their use in the postulates.

We are left then with geometry as a purely logical system (a *first-order language*, in mathe-

matical logic [81, Section 4.2]). When establishing the validity of a statement in geometry, any connection to its empirical roots is irrelevant. We call this system *Euclidean geometry*, or sometimes *parabolic geometry*.

In a sense though, geometry is still "about" objects like points and lines. Geometry can be tied back to a concrete universe of points and lines through an *interpretation*. An interpretation of Euclidean geometry is a translation of the abstract points and lines into well-defined sets, and a translation of the incidence, congruence, and betweenness relations into well-defined relations on those sets. The postulates of geometry then become statements in the mathematical world of the interpretation. An interpretation is called a *model* of geometry when all the postulates are true statements.

The familiar Cartesian plane, with points interpreted as ordered pairs of real numbers, is a model of Euclidean geometry. But it is a mistake to say that the Cartesian plane *is* Euclidean geometry. Other inequivalent models are possible for the postulates given above. It was only in the nineteenth century, when the primacy of Euclidean geometry was finally called into question, that mathematicians worked to rule out these alternate models and make the logical framework of geometry match up with the intuition it sought to formalize. Several systems (such as Hilbert's) emerged that were *categorical*: every model of the system is isomorphic to the Cartesian plane. In such categorical systems, it is once again safe to picture Euclidean geometry as Euclid did, in terms of the intuitive notions of points and lines.

2.1.1 Hyperbolic geometry

The fifth postulate, the so-called "parallel postulate," is the source of one of the greatest controversies in the history of science, and ultimately led to one of its greatest revolutions.

In any logical system, the postulates (also called the *axioms*) should be obvious, requiring only a minimal investment of credulity. From the start, however, the parallel postulate was considered much too complicated, a lumbering beast compared to the other four. Euclid himself held out as long as possible, finally introducing the parallel postulate in order to prove his twenty-ninth proposition.

For centuries, mathematicians struggled with the parallel postulate. They sought either to replace it with a simpler, less contentious axiom, or better yet to establish it as a consequence of the first

four postulates. Neither approach proved successful; no alternate postulate was found that was uncontroversial, and any attempts to formulate a proof either led to a dead end or involved a hidden assumption, itself equivalent to the parallel postulate.

All of these efforts were based on one fundamental hidden assumption: that the only possible geometry was that of Euclid. The possibility that an unfamiliar but perfectly valid geometry could exist without the parallel postulate was unthinkable. Literally so, according to Kant, who in his *Critique of Pure Reason* declared that Euclidean geometry was not merely a fact of the physical universe, but inherent in the very nature of thought [62, Page 182].

Finally, in the nineteenth century, a breakthrough was made by three mathematicians: Bolyai, Gauss, and Lobachevski. They separately realized that the parallel postulate was in fact *independent* of the rest of Euclidean geometry, that it could be neither proven nor disproven from the other axioms. Each of them considered an alternate logical system based on a modified parallel postulate in which multiple lines, all parallel to l, could pass through point P. This new geometry appeared totally self-consistent, and indeed was later proven to be so by Beltrami.[1] Paulos [117] likens the consistency of non-Euclidean geometry to the surprising but plausible incongruity that makes riddles funny – the riddle in this case being "What satisfies the first four axioms of Euclid?"

Today, we refer to the non-Euclidean geometry of Bolyai, Gauss, and Lobachevsky as *hyperbolic geometry*, the study of points and lines in the *hyperbolic plane*. Hyperbolic geometry is based on the following alternate version of the parallel postulate:

V. Given a line l and a point P not on l, there exist at least two lines m_1 and m_2 through P that are parallel to l.

In Euclidean ornamental designs, parallel lines can play an important role. To thicken a mathematical line l into a band of constant width w, we can simply take the region bounded by the two parallels of distance $w/2$ from l. This approach presents a problem in hyperbolic geometry, where these parallels are no longer uniquely defined. On the other hand, parallelism is not the defining quality of a thickened line, merely a convenient Euclidean equivalence. What we are really after are

---

[1]Beltrami's proof hinged upon exhibiting a model of non-Euclidean geometry in the Euclidean plane. Any inconsistency in the logical structure of non-Euclidean geometry could then be interpreted as an inconsistency in Euclidean geometry, which we are assuming to be consistent. This sort of *relative consistency* is about the best one could hope for in a proof of the validity of any geometry.

the loci of points of constant perpendicular distance $w/2$ from l. These are called *equidistant curves* or *hypercycles*, and they are always uniquely defined. In the Euclidean plane, equidistant curves are just parallel lines. In the hyperbolic plane, they are curved paths that follow a given line.

There are several different Euclidean models of hyperbolic geometry; all are useful in different contexts. Each has its own coordinate system. Hausmann *et al.* [78] give formulae for converting between points in the three models.

In the *Poincaré model*, the points are points in the interior of the Euclidean unit disk, and the lines are circular arcs that cut the boundary of the disk at right angles (we extend this set to include diameters of the disk). The Poincaré model is *conformal*: the angle between any two hyperbolic lines is accurately reflected by the Euclidean angle between the two circular arcs [2] that represent them. The Poincaré model is therefore a good choice for drawing Euclidean representations of hyperbolic patterns, because in some sense it does the best job of preserving the "shapes" of hyperbolic figures. It also happens to be particularly well-suited to drawing equidistant curves; in the Poincaré model, every equidistant curve can be represented by a circular arc that does not cut the unit disk at right angles.

The points of the *Klein model* are again the points in the interior of the Euclidean unit disk, but hyperbolic lines are interpreted as chords of the unit disk, including diameters. The Klein model is *projective*: straight hyperbolic lines are mapped to straight Euclidean lines. This fact makes the Klein model useful for certain computations. For example, the question of whether a point is inside a hyperbolic polygon can be answered by interpreting it through the Klein model as a Euclidean point-in-polygon test.

The *Minkowski model* [40, 51] requires that we move to three dimensional Euclidean space. Here, the points of the hyperbolic plane are represented by one sheet of the hyperboloid $x^2 + y^2 - z^2 = -1$, and lines are the intersections of Euclidean planes through the origin with the hyperboloid. The advantage of The Minkowski model is that rigid motions (see Section 2.2 for more on rigid motions) can be represented by three dimensional linear transforms. Long sequences of motions can therefore be concatenated via multiplication, as they can in the Euclidean plane. Our software implementations of hyperbolic geometry are based primarily on the Minkowski model, with points

---

[2]The angle between two arcs is measured as the angle between their tangents at the point of intersection.

converted to the Poincaré model for output.

Note that although there are several models for hyperbolic geometry (including others not discussed here), it is still categorical. The Poincaré, Klein, and Minkowski models are all isomorphic [62, Page 236].

2.1.2 Elliptic geometry

Given a line l and a point P not on l, we have covered the cases where exactly one line through P is parallel to l (Euclidean geometry) and where several lines are parallel (hyperbolic geometry). One final case remains to be explored:

V. Given a line l and a point P not on l, every line through P intersects l.

Once again, this choice of postulate leads to a self-consistent geometry, called *elliptic geometry*. In elliptic geometry, parallel lines simply do not exist.

A first attempt at modeling elliptic geometry would be to let the points be the surface of a three dimensional Euclidean sphere. Lines are interpreted as great circles on the sphere. Since any two distinct great circles intersect, the elliptic parallel property holds. This model is invalid, however, because Euclid's first postulate fails. Antipodal points lie on an infinite number of great circles.

A strange but simple modification to the spherical interpretation can make it into a true model of elliptic geometry. A point is interpreted as a *pair* of antipodal points on the sphere. Lines are still great circles. The identification of a point with its antipodal counterpart fixes the problem with the first postulate, because no elliptic "point" is now more than a quarter of the way around the circle from any other, and the great circle joining those points is uniquely defined.

Despite this antipodal identification, the elliptic plane can still be drawn as a sphere, with the understanding that half of the drawing is redundant. Any elliptic figure will be drawn twice in this representation, the two copies opposite one another on the sphere. Note also that the equidistant curves on the sphere are simply non-great circles.

2.1.3 Absolute geometry

The parallel postulate is independent from the other four, which allows us to choose any of the three alternatives given above and obtain a consistent geometry. But what happens if we choose *none* of

them? In other words, let us decide to leave the behaviour of parallel lines undefined, and develop that part of geometry that does not depend on parallelism. We refer to this geometry, based only on the first four postulates, as *absolute geometry*.

Formally, this choice presents no difficulties whatsoever. We have already assumed that the first four postulates are consistent, and so they must lead to some sort of logical system. Furthermore, we already know that many Euclidean theorems still hold in absolute geometry; these are the ones whose proofs do not rely on the parallel postulate. The first twenty-eight of Euclid's propositions have this property.

In practice, the absolute plane is somewhat challenging to work with. As always, in order to visualize the logical system represented by absolute geometry, we need a model. Such models are easy to come by, because any model of parabolic, hyperbolic, or elliptic geometry is automatically a model of absolute geometry! Of course, those models do not tell the whole story (or rather, they tell more than the whole story), because in each case parallels have some specific behaviour. This behaviour does not invalidate the model, but it imposes additional structure that can be misleading. It is perhaps easier to imagine absolute geometry as a purely formal system, one that contains all the constructions that are common to parabolic, hyperbolic, and elliptic geometry.

The model of elliptic geometry presented above can be somewhat difficult to visualize and manipulate. In some ways, it would be desirable to work directly with the sphere with no identification of antipodal points. From there, perhaps an absolute geometry could be developed that unifies the Euclidean plane, the hyperbolic plane, and the sphere in a natural way.

Unfortunately, as we have seen, the native geometry on the sphere violates Euclid's first postulate. However, it turns out that by giving a slightly revised set of axioms, we can in fact develop a consistent geometry modeled by the Euclidean sphere without the identification of antipodal points. This geometry is called *spherical geometry*, or sometimes *double elliptic geometry*. Moving from elliptic to spherical geometry requires some reworking of Euclid's postulates, but is justified by the convenience of a far more intuitive model.

Kay [97] develops an axiomatic system for spherical geometry. The trick is to start with *ruler and protractor postulates*, axioms that provide formal measures of distance and angle. A (possibly infinite) real number α is then defined as the supremum of all possible distances between points. On the sphere, α is half the circumference; in the Euclidean and hyperbolic planes, α is infinite. Kay

then insinuates α into his axioms, using it to do the bookkeeping necessary to avoid problematic situations. For instance, his version of Euclid's first postulate is as follows:

I. Any two points P and Q lie on at least one line; when the distance from P to Q is less than α, the line is unique.
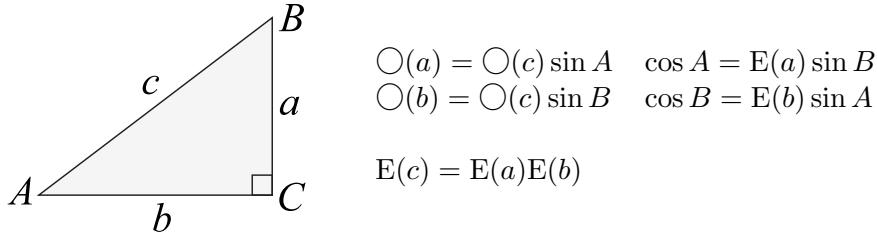
Kay's presentation carefully postpones any assumption on parallelism until the final axiom. As a result, we can consider the geometry formed by all the axioms except the last one. This is a form of absolute geometry that can be specialized into parabolic, hyperbolic, and spherical (as opposed to elliptic) geometry.

In the absence of any single model that exactly captures its features, one may wonder how absolute geometry can be made practical. We do know that any theorem of absolute geometry will automatically hold in parabolic, hyperbolic, and spherical geometry, since formally they are all just special cases. By interpreting absolute geometry in various different ways, we can then view that theorem as a true statement about the Cartesian plane, the Poincaré disk, the sphere, or any of the other models discussed above. In effect, we can imagine an implementation of absolute geometry that is parameterized over the model. In computer science terms, the interface has no inherent representation of points or lines, but interprets the axioms of absolute geometry as a contract that will be fulfilled by any implementation. A client program can be written to that contract, and later instantiated by plugging in any valid model. This approach will be discussed in greater detail in Section 3.6.

2.1.4 Absolute trigonometry

Trigonometry is the study of the relationships between parts of triangles. Triangles exist in absolute geometry and can therefore be studied using *absolute trigonometry*. Bolyai first described some of the properties of absolute triangles; his work was later expanded upon by De Tilly [14, Page 113].

Following De Tilly, we define two functions on the real numbers: $\bigcirc(x)$ and $\mathrm{E}(x)$. The function $\bigcirc(x)$, or "circle-of-x," is defined as the circumference of a circle of radius x. To obtain a definition for $\mathrm{E}(x)$, let l be a line and c be an equidistant curve erected at perpendicular distance x from l. Take any finite section out of the curve, and define $\mathrm{E}(x)$ as the ratio of the length of that segment to the length of its projection onto l. It can be shown that this value depends only on x. From these

$$\bigcirc(a) = \bigcirc(c)\sin A \qquad \cos A = \mathrm{E}(a)\sin B$$
$$\bigcirc(b) = \bigcirc(c)\sin B \qquad \cos B = \mathrm{E}(b)\sin A$$

$$\mathrm{E}(c) = \mathrm{E}(a)\mathrm{E}(b)$$

Figure 2.1 Trigonometric identities for a right triangle in absolute geometry. When interpreted in Euclidean geometry, the equation $\bigcirc(a) = \bigcirc(c)\sin A$ becomes "sine equals opposite over hypoteneuse," and $\cos A = \mathrm{E}(a)\sin B$ becomes $\cos A = \sin(\frac{\pi}{2} - A)$. The equation $\mathrm{E}(c) = \mathrm{E}(a)\mathrm{E}(b)$ is vacuously true (and therefore not particularly useful) in Euclidean geometry, but in hyperbolic and spherical geometry it can be seen as one possible analogue to the Pythagorean theorem.

two functions, we can also define $\mathrm{T}(x) = \bigcirc(x)/\mathrm{E}(x)$, and the three inverses $\bigcirc^{-1}(x)$, $\mathrm{E}^{-1}(x)$, and $\mathrm{T}^{-1}(x)$.

Bolyai initiated absolute trigonometry with the observation that the sines of the angles of a triangle are as the circumferences of circles with radii equal to the lengths of the opposite sides. Expressed in the notation just given, we can say that for any triangle with vertices ABC and opposite edges abc, $\bigcirc(a)/\sin A = \bigcirc(b)/\sin B = \bigcirc(c)/\sin C$. By substituting the Euclidean definition of $\bigcirc(x)$, this relationship can be seen as a generalization of the sine law to absolute geometry. Other identities of absolute geometry that apply specifically to right triangles are summarized in Figure 2.1.

We can give formulae for $\bigcirc(x)$ and $\mathrm{E}(x)$, though their definitions must be broken down into cases. Each case corresponds to the choice of a parabolic, hyperbolic, or spherical model.

	parabolic	hyperbolic	spherical
$\bigcirc(x)$	$2\pi x$	$2\pi \sinh x$	$2\pi \sin x$
$\mathrm{E}(x)$	1	$\cosh x$	$\cos x$

While it is perfectly valid to analyze absolute triangles abstractly using absolute trigonometry, actual values for side lengths and vertex angles can be derived only by resorting to one of the models. This curious fact follows from the difference between the axiomatic and analytic views of geometry. Because Euclidean geometry is categorical, the usual trigonometric functions in the Cartesian plane

are the only ones (up to isomorphism) that satisfy the axiomatic relationships in a Euclidean triangle. However, absolute geometry is not categorical, meaning that different, inequivalent trigonometric functions can (and do) hold under different models. We are not used to making this distinction, because in ordinary Euclidean geometry there effectively is no distinction. It can be challenging to one's intuition to visualize such functions that are well-defined formally but not analytically.

2.2 Symmetry

The mathematical tools behind a formal treatment of symmetry are relatively new, but our appreciation of symmetric patterns goes back millennia [137]. In addition to its usefulness in many branches of science, symmetry is often used to study art and ornament [127, 135]. M.C. Escher interacted with the growth of symmetry theory, creating new art based on the mathematical results that emerged during his lifetime [124].

The original conception of symmetry, as conveyed by the dictionary definition, is expressed with words such as beauty, balance, and harmony. The word was and still is used to refer to a balance of components in a whole.

The contemporary non-scientific usage of the word, as Weyl points out, refers to an object whose left and right halves correspond through reflection in a mirror [137]. Thus a human figure, or a balance scale measuring equal weights, may be said to possess symmetry.

In light of the formal definition of symmetry to come, we qualify the particular correspondence described above as "bilateral symmetry." Bilateral symmetry is certainly a familiar experience in the world around us; it is found in the shapes of most higher animals. The prevalence of bilateral symmetry can be explained in terms of the body's response to its environment. Whereas gravity dictates specialization of an animal from top to bottom and locomotion engenders differentiation between front and back, the world mandates no intrinsic preference for left or right [137, Page 27]. An animal must move just as easily to the left as to the right, resulting in equal external structure on each side. Indeed, lower life forms whose structure is not as subject to the exigencies of gravity and linear locomotion tend towards more circular or spherical body plans.

Let us regard the mirror of bilateral symmetry as a reflection through a plane in space. Saying that the mirror reconstructs half of an object from the other half is equivalent to saying that the

reflection maps the entire object onto itself. We formalize the notion of symmetry by noting two properties of this reflection. It *preserves the structure of space*, just as a (flat) mirror preserves the shapes of objects, and it *maps the object onto itself*, allowing us to think of its two halves as having the same shape. By generalizing from these two properties, we will arrive at a formal definition of symmetry.

Given a mathematical space $\mathcal{S}$, we identify some important aspect of the mathematical structure of $\mathcal{S}$, and define a set of *motions* $\mathcal{M}$ to be automorphisms of $\mathcal{S}$ that preserve that structure. Then, given some figure $F \subseteq \mathcal{S}$, we can say that a motion σ is a *symmetry* of F if $\sigma(F) = F$, that is, if σ maps the figure F to itself.

This somewhat vague definition achieves rigour when we give a specific meaning to "mathematical structure." As a simple example, let $\mathcal{S}$ be the integers from 1 to n, and consider preserving no structure of $\mathcal{S}$ beyond the set-theoretic. Then the motions $\mathcal{M}$ are just the $n!$ permutations of the members of $\mathcal{S}$, and every k-element figure (subset) of $\mathcal{S}$ has $k!(n-k)!$ symmetries, each one permuting the figure internally and the remaining elements of $\mathcal{S}$ externally.

The important mathematical spaces in the present work are the parabolic, the hyperbolic, and the spherical planes. We know from Kay's presentation [97] that each of these planes has a notion of distance, defined both formally in the axioms of geometry and concretely in the models. If we let $\mathcal{S}$ be the set of points in one of these planes, then we can define the motions to be the *isometries* of $\mathcal{S}$: the automorphisms that preserve distance. This point of view leads to the most important and most common definition of symmetry: a symmetry of a set F is an isometry that maps F to itself. We will also sometimes use the term *rigid motion* in the place of isometry; isometries are rigid in the sense that they do not distort shape. In the Euclidean plane, the symmetries of a figure are easily visualized by tracing the figure on a transparent sheet and moving that sheet around the plane, possibly flipping it over, so that that original figure and its tracing line up perfectly [68, Page 28].

2.2.1 Symmetry groups

For a given $F \subseteq \mathcal{S}$, let $\Sigma(F)$ be the set of all motions that are symmetries of F. This set has a natural group structure through composition of automorphisms. The set $\Sigma(F)$ is therefore called

the *symmetry group* of F.
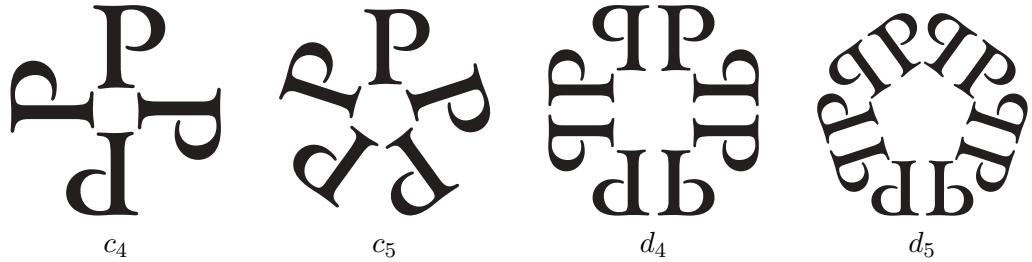
The *orbit* of a point $p \in \mathcal{F}$ under a symmetry group G is the set $\{\sigma(p)|\sigma \in G\}$. When $\mathcal{S}$ is equipped with a measure of distance, we say that the symmetry group G is *discrete* if for every point p, the orbit of p does not get arbitrarily close to p. More precisely, if $d(p, q)$ is the distance between points p and q in $\mathcal{S}$, then G is discrete if for all p, $\inf\{d(p, \sigma(p))|\sigma \in G, \sigma(p) \neq p\} > 0$. A circle is an example of a figure with non-discrete symmetry; every point on the circle is a limit point of its orbit. In this work, we restrict ourselves to discrete symmetry groups, a technical but important point that simplifies the classification of the groups we will use. There exist ornamental designs that can be profitably analyzed using non-discrete symmetry groups, but such designs will not arise here.

Symmetry is a measure of redundancy in a figure, and so we ought to be able to use our understanding of the symmetries of a figure to factor out the redundancy. The result would be a minimal, sufficient set of non-redundant information that, together with the symmetries, completely describe the original figure. For any discrete symmetry group, this set exists and is called the group's *fundamental region* or *fundamental domain*. One possible definition, given by Grünbaum and Shephard [68, Section 1.6], says that $U \subseteq \mathcal{S}$ is a fundamental region of symmetry group G if the following conditions hold:

(a) U is a connected set with non-empty interior.

(b) No two points in U have the same orbit under G (or equivalently, for all $p, q \in U$, there does not exist a $\sigma \in G$ such that $\sigma(p) = q$).

(c) U is as big as possible; that is, no other set satisfying (a) and (b) contains U as a proper subset.

The first condition ensures that the fundamental region is relatively simple topologically. The other two guarantee that the region has "exactly enough" information — condition (b) rules out redundancy and condition (c) forces every orbit to be represented by some point in U.

It is important from an algorithmic standpoint to understand a symmetry group's fundamental region. The fundamental region contains a single, non-redundant copy of the information in a symmetric figure. Therefore, in order to draw the figure as a whole, it suffices to draw all images of the fundamental region under the motions of the symmetry group. The drawing process can be seen as a replication algorithm that determines the set of motions to apply, and a subroutine that draws a

c_4 c_5 d_4 d_5
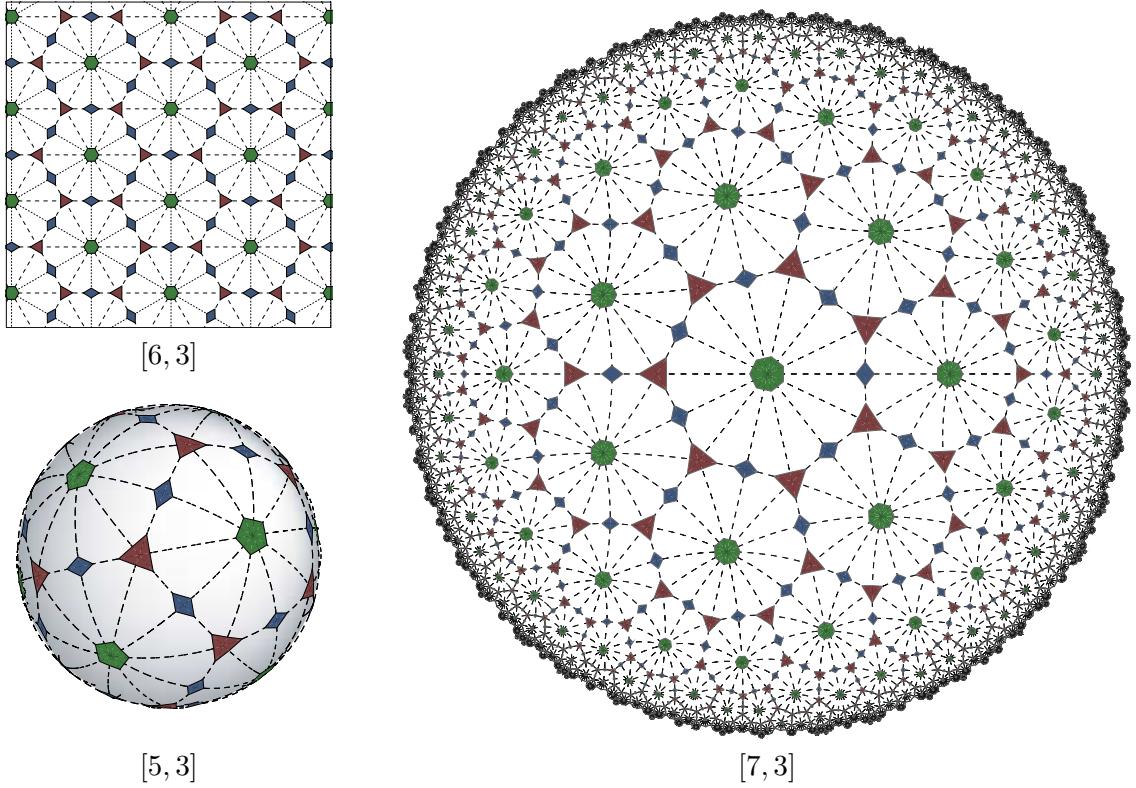
Figure 2.2 Figures in the Euclidean plane with c_n or d_n symmetry. Each figure is labeled with its symmetry group.

transformed copy of the fundamental region.

2.2.2 Some important symmetry groups

The discrete symmetry groups in the Euclidean plane are well understood. Up to isomorphism, only a limited collection of discrete groups can be symmetry groups of Euclidean figures. They can be classified according to the nature of the subgroup of the symmetry group consisting of just translations:

- If the translational subgroup is trivial, then the symmetry group must be finite, isomorphic either to c_n, the cyclic group of order n, or d_n, the dihedral group of order $2n$. The group c_n is the symmetry group of an n-armed swastika, and d_n is the group of a regular n-gon. These possibilities were enumerated by Leonardo Da Vinci, and the completeness of the enumeration is sometimes called "Leonardo's Theorem" [109, Section 30.1]. Examples of figures with c_n and d_n symmetry are shown in Figure 2.2.

- If all the translations are parallel, then the symmetry group must be one of the seven *frieze groups*. Friezes are decorations executed in bands or strips, and the frieze groups are particularly well suited to their study. A *frieze pattern* is a figure with a frieze group as its symmetry group.

- The remaining case is when the translational subgroup contains translations in two non-parallel directions. This category consists of the seventeen *wallpaper groups*. A *wallpaper*

Figure 2.3 Examples of symmetry groups of the form $[p, q]$. Each example visualizes the lines of reflection (shown as dotted lines) and centers of rotation of the symmetry group. The green, red, and blue forms represent centers of p-fold, q-fold, and twofold rotation, respectively.

pattern (sometimes called a *periodic pattern*) is a figure with a wallpaper group (or a *periodic group*) as its symmetry group. The wallpaper patterns are "all-over" patterns in the sense that the pattern repeats in every direction, not just in one distinguished direction as is the case with frieze patterns. A wallpaper pattern necessarily has a bounded fundamental region.

For images of frieze and wallpaper patterns, see for example Grünbaum and Shephard [68, Section 1.3], Beyer [11] (a lovely presentation in the context of quilt design), or Shubnikov and Koptsik [127]. Washburn and Crowe [135] give many examples and include a flowchart-based technique for classifying a given pattern.

The groups c_n and d_n can also occur as the symmetries of figures in the hyperbolic plane and on the sphere, but the frieze and wallpaper patterns are very much tied to Euclidean geometry. The linear independence implied by translational symmetry in two non-parallel directions only makes sense in an affine space; the Euclidean plane is affine, whereas the sphere and hyperbolic plane are not.

A different set of all-over patterns leads to a family of symmetry groups that span Euclidean and non-Euclidean geometry. For every $p \geq 2$ and $q \geq 2$, there is a group $[p, q]$, which can be seen as the symmetry group of a tiling of the plane by regular p-gons meeting q around every vertex [26, Chapter 5]. These regular tilings are discussed in greater detail in Section 2.3.1.
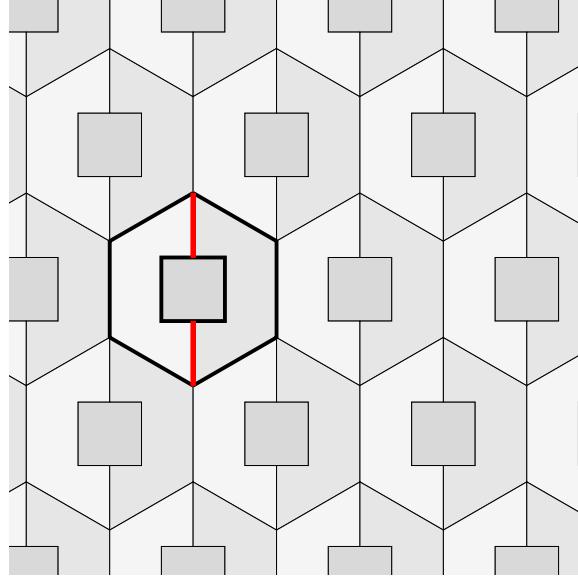
A simple calculation shows that each group $[p, q]$ is tied to one of the three planar geometries. In particular, $[p, q]$ is a symmetry group of the Euclidean, hyperbolic, or spherical plane when $1/p + 1/q$ is respectively equal to, less than, or greater than $1/2$. Note that $[p, q]$ is isomorphic to $[q, p]$, even though the regular tilings they are based on are different. Three examples of $[p, q]$ symmetry groups are shown in Figure 2.3.

The fundamental region of $[p, q]$ is a right triangle with interior angles π/p and π/q. The entire group can be said to be *generated* by reflections in the sides of this triangle, in the sense that every symmetry in the group is the product of a finite number of such reflections.

2.3   Tilings

In their indispensible book *Tilings and Patterns* [68], Grünbaum and Shephard develop an extensive theory of tilings of the Euclidean plane. They begin from first principles with a nearly universally inclusive definition of tilings, one that permits so many pathological cases that the resulting objects cannot be meaningfully studied. They then proceed to layer restrictions upon the basic definition, creating ever smaller families of tilings that yield to more and more precise analysis and classification. Although the material persented here is largely drawn from *Tilings and Patterns*, it should not be considered to apply only in the Euclidean plane. Most of the basic facts about tilings apply equally well in non-Euclidean geometry.

For the purposes of creating the kinds of ornaments we will encounter in this work, we can jump in fairly late in the process and accept the following definition, corresponding to their notion of

Figure 2.4 A tiling for which some tiles intersect in multiple disjoint curves. The intersection between the two tiles outlined in bold is shown in red; it consists of two disjoint line segments.
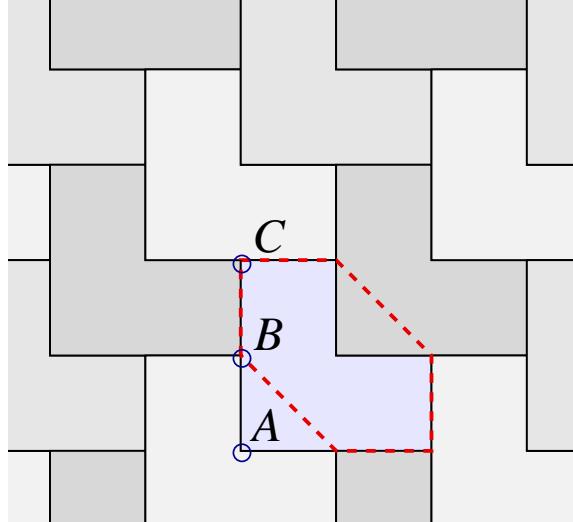
"normal tiling" [68, Section 3.2].

Definition (Tiling) *A tiling is a countable collection $\mathcal{T}$ of tiles $\{T_1, T_2, \ldots\}$, such that:*

1. *Every tile is a closed topological disk.*

2. *Every point in the plane is contained in at least one tile.*

3. *The intersection of every two tiles is empty, a point, or a simple closed curve.*

4. *The tiles are uniformly bounded; that is, there exist $u, U > 0$ such that every tile contains a closed ball of radius u and is contained in a closed ball of radius U.*

The most natural property associated with tilings, that they cover the plane with no gaps and no overlaps, is handled by conditions 2 and 3. Conditions 1 and 4 ensure that the tiles are reasonably well behaved entities that do not have exotic topological properties or become pathological at infinity.

Observe that condition 3 does more than prevent tiles from overlapping. It also prevents tilings like the one shown in Figure 2.4 from arising, where the boundary between two tiles is disconnected. When two tiles intersect in a curve, we may then refer to this well-defined curve as a *tiling edge*.

Figure 2.5 The features of a tiling with polygonal tiles. For the tile highlighted in blue, A is a shape vertex but not a tiling vertex, B is a tiling vertex but not a shape vertex, and C is both a tiling vertex and a shape vertex. The tiling polygon is shown as a dashed red outline.

Every tiling edge begins and ends at a *tiling vertex*, a place where three or more tiles meet. The tiling vertices are topologically important points in a tiling, as they determine the overall connectivity of the tiles and adjacencies between them. We will sometimes use the term "tiling polygon" to refer to the polygon joining the tiling vertices that lie on a single tile.

When the tiles in a tiling are polygons, there can be some confusion between the tiling vertices and edges as described above and the vertices and edges of individual polygons. To avoid confusion, we refer to the latter features when necessary as *shape vertices* and *shape edges*. Shape vertices and edges are properties of tiles in isolation; tiling vertices and edges are properties of the assembled tiling. Although the features of the tiling occupy the same positions as the features of the tiles, they may break down differently, as shown in Figure 2.5. When the two sets of features coincide (that is, when the tiling vertices are precisely the shape vertices), the tiling is called *edge-to-edge*.

In many of the tilings we see every day on walls and streets, the tiles all have the same shape. If every tile in a tiling is congruent to some shape T, we say that the tiling is *monohedral*, and that T is the *prototile* of the tiling. More generally, a *k-hedral* tiling is one in which every tile is congruent

to one of k different prototiles. We also use the terms *dihedral* and *trihedral* for the cases $k = 2$ and $k = 3$, respectively. Note that a tiling need not be k-hedral for any finite k.

Just as with any other object in the plane, a tiling can be symmetric. Symmetries are a property of figures in the plane; the figure of a tiling can be taken as the set of points that lie on the boundary of any tile. Alternatively, we can alter the definition of symmetry slightly, saying that a symmetry of a tiling is a rigid motion that maps every tile onto some other tile.

2.3.1 Regular and uniform tilings
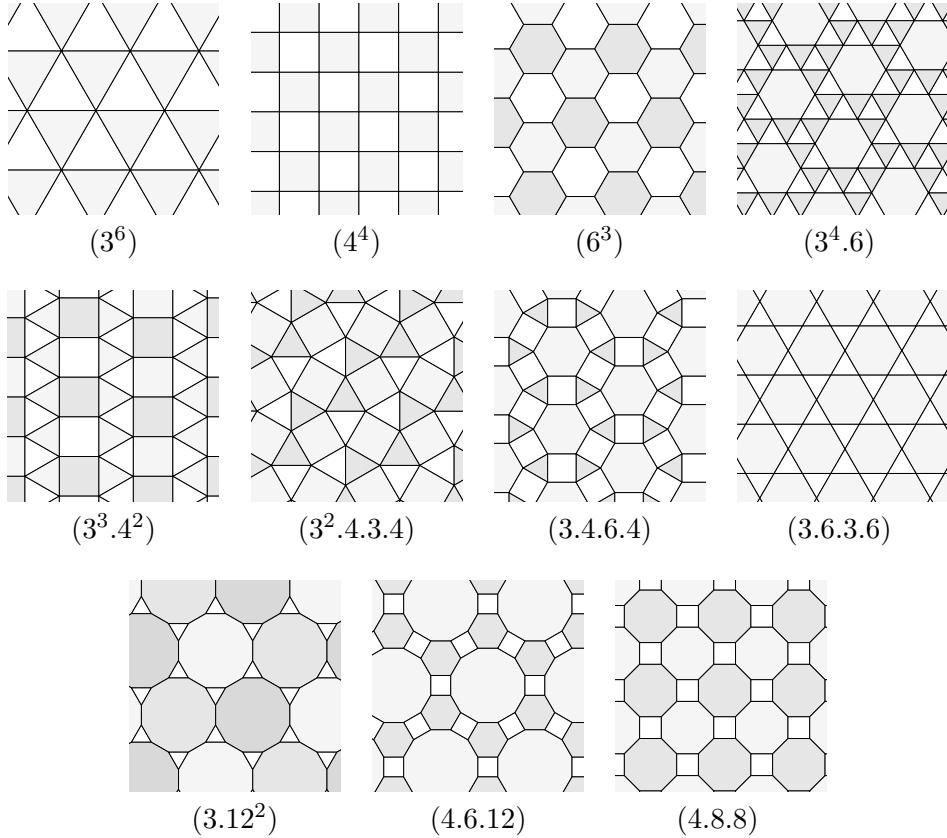
A *regular tiling* is an edge-to-edge tiling of the plane by congruent regular polygons. In the Euclidean plane, an easy calculation shows that the only regular tilings are the familiar ones by squares, equilateral triangles, and regular hexagons.

We may then ask about edge-to-edge tilings of the plane using two or more different regular polygons as prototiles. In general, we can say very little about an arbitrary tiling of this type. To get more predictable behaviour, we further require that the vertices be indistinguishable, in the sense that any vertex can be mapped onto any other via a rigid motion that is also a symmetry of the tiling as a whole.[3] Given such a restriction, we can describe the tiling using a *vertex symbol*. A vertex symbol is a sequence $p_1.p_2 \ldots p_n$ that enumerates, in order, the regular polygons encountered around every vertex in the tiling. The tilings that can be described using vertex symbols in this way are called *uniform tilings*.

In the Euclidean plane, we can enumerate all vertex symbols that correspond to legal uniform tilings of the plane [68, Section 2.1]. The result is a set of eleven distinct tilings, sometimes also known as the *Archimedean tilings*. We name these tilings by placing their vertex symbols in parentheses. Among the uniform tilings the regular tilings are the ones whose vertex symbols are of the form p^q: (4^4) for squares, (3^6) for equilateral triangles, and (6^3) for regular hexagons. (In a vertex symbol, we abbreviate blocks of repeated values using exponentiation.) Figure 2.6 shows the eleven Archimedean tilings.

There are vertex symbols that clearly cannot correspond to tilings of the Euclidean plane because

---
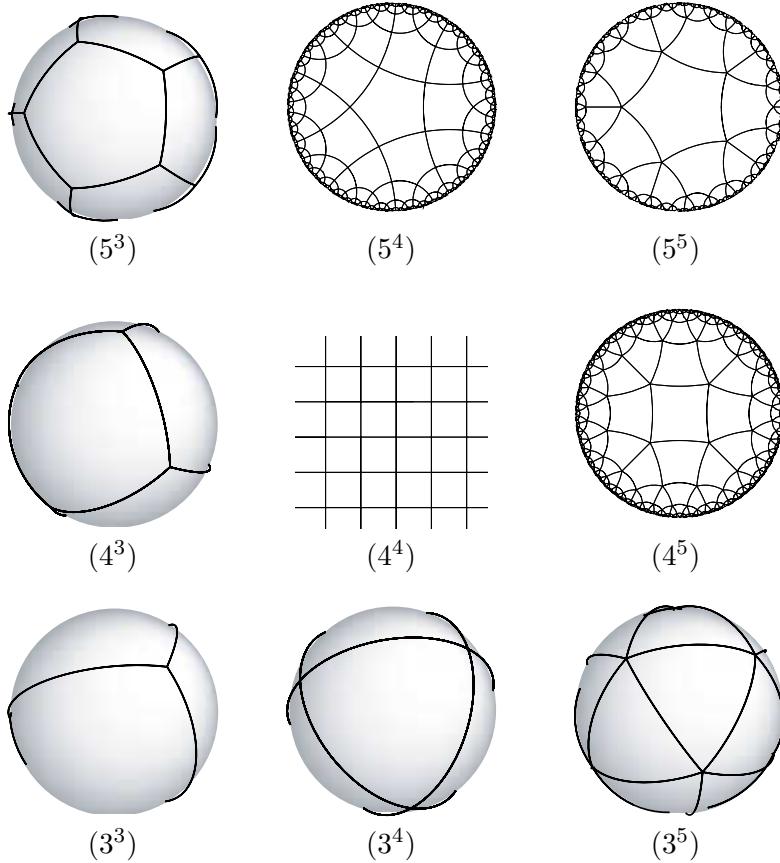
[3]Such tilings are usually called *isogonal*. In an isogonal tiling, the vertices form one transitivity class with respect to the tiling's symmetries. Isohedrality, a definition based on transitivity of whole tiles, will be encountered in Section 2.4.

Figure 2.6 The eleven uniform Euclidean tilings, also known as Archimedean tilings. The tiling $(3^4.6)$ occurs in two mirror-image forms.

the interior angles of the regular polygons around a vertex do not sum to 360 degrees. For example, five equilateral triangles leave a gap when arranged around a point. Yet there is a familiar shape with five equilateral triangles around every point: the icosahedron. Indeed, if we "inflate" the icosahedron so that all its edges become arcs of great circles on a unit sphere (line segments in spherical geometry), the result is a regular tiling of the sphere that can legitimately be called (3^5).

In general, a given vertex symbol $p_1.p_2 \ldots p_n$ may describe a Euclidean, spherical, or hyperbolic tiling (or no tiling at all). If there is a tiling associated with the symbol, the tiling will be Euclidean, spherical, or hyperbolic when $\sum_{i=1}^{n} \frac{1}{p_i}$ is respectively equal to, greater than, or less than $\frac{1}{2}$. In particular, for any $p \geq 3$ and $q \geq 3$ there is a regular tiling (p^q), consisting of regular p-gons meeting q around every vertex. We have already seen the three Euclidean cases. The five spherical

Figure 2.7 Some Euclidean and non-Euclidean regular tilings.

cases correspond to the Platonic solids. The remaining cases are all hyperbolic. Some examples of regular tilings are shown in Figure 2.7. As discussed in Section 2.2.2, each regular tiling (p^q) has symmetry group $[p, q]$, generated by reflections in a right triangle with interior angles π/p and π/q.

There is no simple mathematical formula for deciding whether an arbitrary vertex symbol can be realized as a uniform tiling. For the most part, one simply tries to build the tiling, either discovering that all roads lead to contradictions, or finding a pattern that allows the tiling to be continued forever. As in the Euclidean case, the uniform tilings of the sphere are completely enumerated[4] – they

---

[4]The uniform tilings of the sphere should not be confused with the so-called *uniform polyhedra* [77]. A uniform polyhedron is allowed to have non-convex faces and may pass through itself. These polyhedra follow from a natural extension of the notion of uniform tiling in which the vertex symbol is allowed to have rational entries. There are

correspond to the five Platonic and thirteen Archimedean solids [136].

The existence of infinitely many regular tilings of the hyperbolic plane implies that there are infinitely many uniform tilings. Enumerating all possible uniform hyperbolic tilings remains an active area of inquiry. One approach is to define functions that transform a regular tiling into a related uniform tiling. Many of these functions are hyperbolic equivalents of spherical versions used by Hart to create novel polyhedra [75] (these functions implement what Hart calls "Conway notation"). Mitchell uses a similar idea to prove the existence of many parameterized families of hyperbolic uniform tilings [112].

Throughout the foregoing discussion, we have avoided the question of whether a given vertex symbol uniquely determines a uniform tiling. As Grünbaum and Shephard point out, there is no reason why this should necessarily be true, but it happens that in the Euclidean plane the uniform tilings are "just barely" unique [68, Page 64]. The same could be said for the sphere. In the hyperbolic plane, there is no guarantee of uniqueness; combinatorially distinct uniform tilings exist with the same vertex symbol [107]. The lack of uniqueness further complicates the enumeration of uniform hyperbolic tilings.
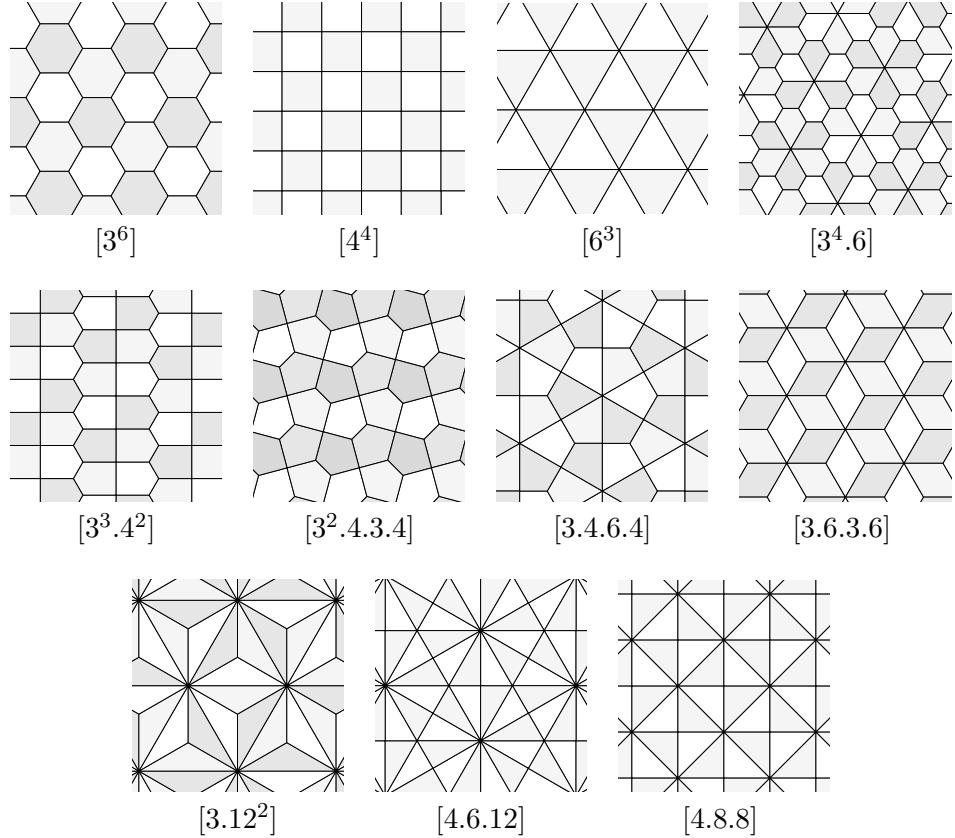
2.3.2 Laves tilings

Every uniform tiling has a well-defined geometric dual, obtained by replacing every n-sided face by an n-valent vertex and vice versa. These dual tilings are monohedral and have the property that every tiling vertex is regular: the edges leaving the vertex are evenly spaced around it. In the Euclidean plane, these duals are called the *Laves tilings*, and they are given names analogous to their Archimedean duals. They are depicted in Figure 2.8. The Laves tilings will prove useful in Section 4.5, where they serve as a set of "defaults" from which to develop more complex tilings.

2.3.3 Periodic and aperiodic tilings

A *periodic tiling* is a tiling of the Euclidean plane with periodic symmetry. That is, there exist two linearly independent directions of translational symmetry. In addition to a fundamental region, every periodic tiling has a *translational unit*, which is a fundamental region of the translational subgroup

---

seventy-five uniform polyhedra.
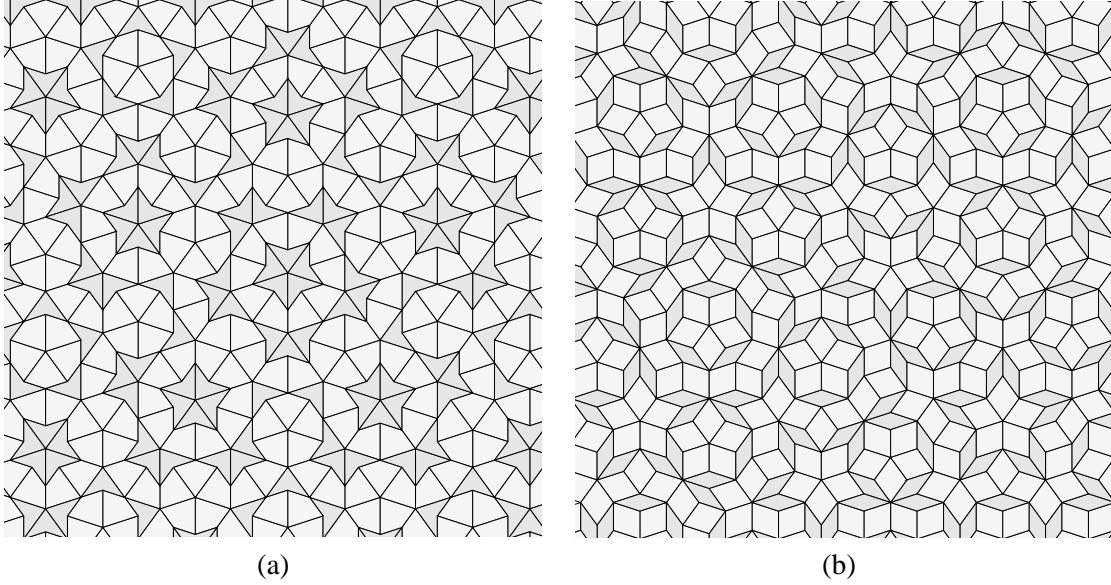
Figure 2.8 The eleven Laves tilings. Each one is the dual of a corresponding Archimedean tiling. The tiling $[3^4.6]$ occurs in two mirror-image forms.

of the tiling's symmetry group. The translational unit can always be chosen to be a parallelogram with sides equal to two of the tiling's shortest non-parallel translational vectors.

The behaviour of periodic tilings is much more predictable than that of tilings in general because this behaviour only varies over a single translational unit. In a periodic tiling, a translational unit can be assembled from a finite number of tiles, guaranteeing that there are only finitely many different tile shapes. Each prototile can only occur in finitely many orientations and reflected orientations. We refer to these orientations collectively as the prototile's *aspects*, and distinguish the *direct aspects* from the *reflected aspects* when necessary.

At the opposite end of the spectrum, *aperiodic tilings* have received a great deal of attention over the past few decades, both from tiling theorists and lay audiences. Especially popular are Penrose's

(a)   (b)
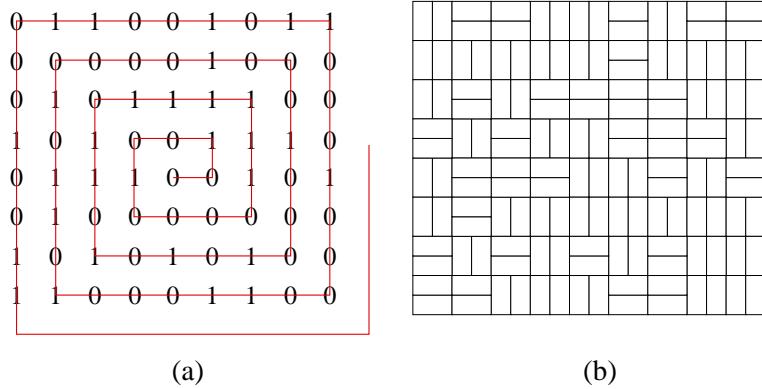
Figure 2.9 The two famous aperiodic tilings of Penrose. The "kite and dart" tiling is shown in (a) and thin and thick rhombs in (b).

aperiodic tilings, which have provided a steady stream of results and inspiration in physics [74, Section 9.8], algebraic geometry [23], and the popular mathematical press [119, Chapter 7]. Of the aperiodic tilings devised by Penrose, the most well known are the ones shown in Figure 2.9, made up either of "kites" and "darts" or thin and thick rhombs.

Like the Mandelbrot set, the Penrose tilings are ambassadors of the geometric aesthetic, bringing their message of mathematical beauty to a general audience. A quick survey of the Geometry Junkyard's web page on Penrose tilings [46] shows dozens of non-technical ways that they have been applied. They have been fashioned into puzzles, games, and fridge magnets; used to cover walls, floors, and windows; even sewn into lace doilies. Glassner has discussed the creation of graphical ornament based on Penrose tilings [54, 55].

A tiling that is not periodic is called *nonperiodic*. A frequent but incorrect assumption is that the definitions of aperiodic and nonperiodic coincide. In fact, the aperiodic tilings are a very special subset of the nonperiodic tilings. It is worthwhile to clarify the distinction between the two, showing why the aperiodic tilings are an active and exciting area of research.
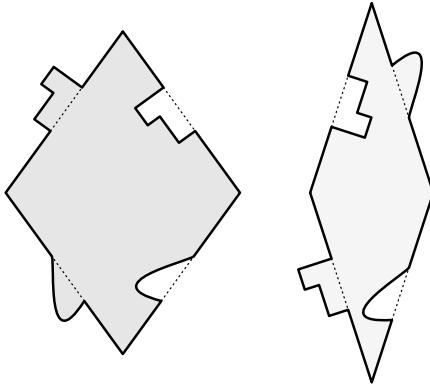
Figure 2.10 A contrived example of how even a very simple shape may yield aperiodic tilings. A spiral path is used to place digits from the binary expansion of π, as given by Wolfram [138, Page 137]. Each digit is then used to place a pair of bricks, oriented vertically to represent a 0 and horizontally to represent a 1. The resulting tiling, when extended to the whole plane, is clearly aperiodic, even though the brick prototile could easily be used to construct periodic tilings. There exist uncountably many aperiodic tilings based on this prototile.

Lack of periodicity is not in itself a very surprising property. One might argue that nonperiodicity is the common case in the universe of tilings, and that the periodic tilings represent small pockets of order in a sea of chaos. This situation is reminiscent of the disposition of the rational numbers, peppered among the much more common irrationals. This analogy can be interpreted quite literally: a number is rational precisely when its digit sequence is periodic. To move from this chaos to a definition of aperiodicity, we must must narrow our point of view significantly.

It is easy to construct tilings where every tile shape is unique – consider, for example, the Voronoi diagram induced by an infinite integer lattice whose points have been jittered randomly. Clearly there can be no hope of periodicity in such a case, but this fact seems unimpressive. In developing a definition of aperiodicity, we therefore consider only those nonperiodic tilings with a finite number of prototiles, *i.e.*, those that are k-hedral for some k.

Under this restricted definition, we can still construct very simple nonperiodic tilings. Even a 2-by-1 brick yields an infinite variety, as demonstrated in Figure 2.10. These tilings seem contrived, however, because the same brick can easily be made to tile periodically. Nonperiodic tilings become

Figure 2.11 Sample matching conditions on the rhombs of Penrose's aperiodic tile set
$P3$. The unmodified rhombs (indicated by dotted lines) can form many periodic tilings.
The puzzle-piece deformations on the tile edges guarantee that any tiling formed from
these new shapes will be aperiodic.

truly interesting when we take into account all possible alternative tilings that can be constructed
from the same set of prototiles. We call a set of prototiles an *aperiodic tile set* when every tiling
admitted by that set is nonperiodic. (We also ask that the set admit at least one tiling!) We can then
define an *aperiodic tiling* as a tiling whose prototiles are an aperiodic tile set. An aperiodic tiling
is one that is "essentially nonperiodic," in the sense that no rearrangement of its tiles will achieve
periodicity. When used to refer to a particular tiling, aperiocity is therefore a far reaching concept —
it encompasses all possible tilings that can be formed from the same prototiles.

The Penrose tilings highlight the special behaviour of aperiodic tilings. Consider the tiling by
Penrose rhombs shown in Figure 2.9(b). By themselves, the two rhombs do not form an aperiodic
tile set; they can be arranged into both periodic and nonperiodic tilings. What makes the rhombs
aperiodic are additional "matching conditions" that are imposed on them, limiting the ways that
two tiles may be adjacent to one another. These matching conditions can be expressed in a number
of ways. One possibility is to modify the shapes of the tiles by adding protrusions to the rhomb
edges, so that the tiles must snap together like pieces in a jigsaw puzzle [68, Section 10.3] (we will
exploit these geometric matching conditions in Section 4.6). A set of protusions that express the
matching conditions is shown in Figure 2.11. It is these two modified shapes that form an aperiodic
tile set, known as the Penrose tile set $P3$ (the modified kite and dart are known as $P2$). Many sets of

prototiles must be endowed with similar matching conditions to enforce aperiodicity. The matching conditions are typically not shown when the tilings are rendered, perhaps leading to the confusion between nonperiodicity and aperiodicity.
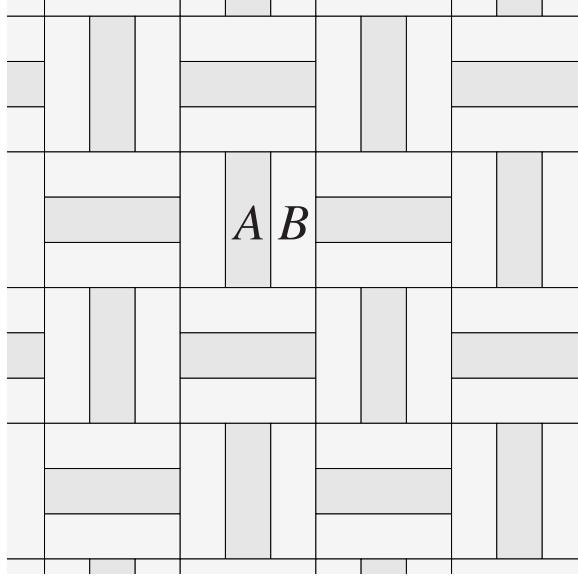
As it turns out, Penrose tilings have even more structure than that determined by the aperiodicity of their prototiles. A beautiful argument shows that even though there are infinitely many different combinatorially inequivalent Penrose tilings from a given set of prototiles, they are all indistinguishable on any bounded region of the plane. Given some Penrose tiling, if we let $\mathcal{P}$ denote the set of tiles that lie inside some bounded region of the plane, then $\mathcal{P}$ will occur infinitely often in every Penrose tiling by those same prototiles. Tilings like these seem to live right on the edge of periodicity, and receive the special designation *quasiperiodic*.

For many years, research in aperiodic tilings has sought to create ever smaller aperiodic tile sets. The first aperiodic sets contained thousands of prototiles [68, Chapter 11]. Penrose managed to bring the minimum number of prototiles needed down to two. Other small aperiodic tile sets have been discovered by Ammann [68, Section 10.4] and Goodman-Strauss [61]. However, nobody has been able to improve upon this result or to show that at least two tiles shapes are required. The question of whether there exists an *aperiodic tile*, a single shape that tiles only aperiodically, remains open [28, Section C18], and is one of the most beautiful unsolved problems in geometry.

2.4 Transitivity of tilings

For two congruent tiles T_1 and T_2 in a tiling, there will be some rigid motion of the plane that carries one onto the other (there may in fact be several). A somewhat special case occurs when the rigid motion is also a symmetry of the tiling. In this case, when T_1 and T_2 are brought into correspondence, the rest of the tiling will map onto itself as well. We then say that the two tiles are *transitively equivalent*.

Transitive equivalence is an equivalence relation that partitions the tiles into *transitivity classes*. When a tiling has only one transitivity class, we call the tiling *isohedral*. More generally, a k-isohedral tiling has k transitivity classes. An isohedral tiling is one in which a single prototile can cover the entire plane through repeated application of rigid motions from the tiling's symmetry group. In an isohedral tiling, there is effectively no way to tell any tile from any other.
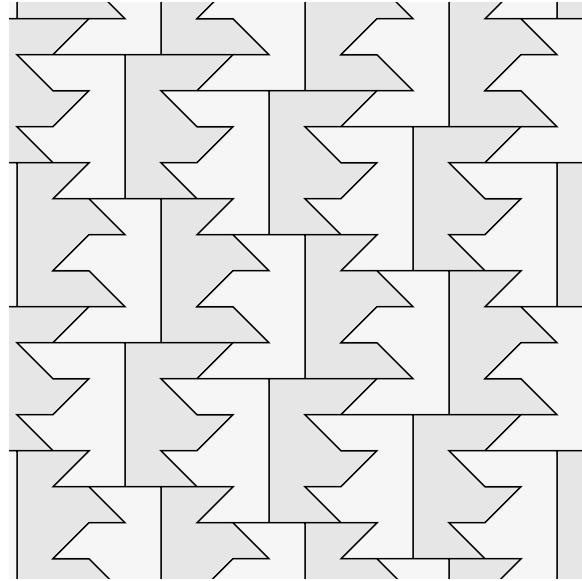
Figure 2.12 An example of a monohedral tiling that is not isohedral. The two tiles labeled A and B cannot be in the same transitivity class, which can be seen by the different ways each is surrounded by its neighbours. This tiling is 2-isohedral.

Two tiles in the same transitivity class must obviously be congruent, but the converse need not be true. Figure 2.12 shows a monohedral tiling with two transitivity classes. The two classes of tiles can be distinguished by the arrangement of a tile's neighbours around it.

As with the definition of aperiodicity, while the tiling given above is 2-isohedral, it seems that way only in a weak sense, because the same shape also tiles the plane isohedrally. We must ask, therefore, whether there exists an *anisohedral tile*, a single shape that tiles the plane, but never isohedrally. In 1900, Hilbert seemed to take it for granted that no such shape can exist [68, Section 9.6]. In 1935, however, Heesch demonstrated an anisohedral prototile [79], reproduced in Figure 2.13.

In Heesch's example, a single prototile generates tilings with two transitivity classes. We therefore specify that the given tile is *2-anisohedral*, and generalize the definition, calling a prototile *k-anisohedral* if every tiling it admits has at least k transitivity classes. Grünbaum and Shephard exhibit a collection 2- and 3-anisohedral pentagons, and ask for what values of k there exist k-anisohedral tiles [68, Section 9.3]. In 1993, Berglund found a 4-anisohedral prototile, and called for examples with $k \geq 5$ [10]. Since 1996, using computer searches over polyominoes, polyia-

Figure 2.13 Heesch's anisohedral prototile. No tiling that can be assembled from this shape will be isohedral.
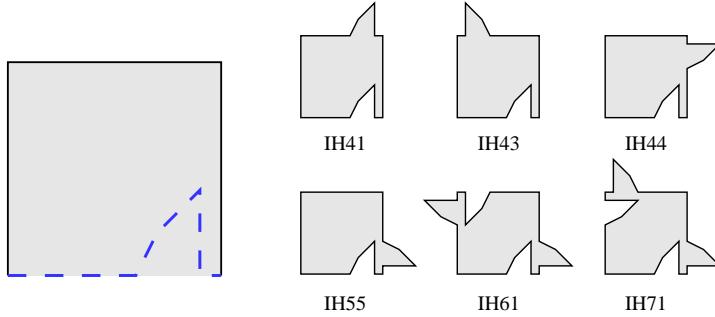
monds, and polyhexes,[5] Joseph Myers has demonstrated k-anisohedral tilings for all $k \leq 9$ [115]. There seems to be no reason to assume an upper bound on possible values of k, though one must imagine the search will become more difficult each time k increases. The search for k-anisohedral prototiles is related to the search for an aperiodic prototile, in the sense that an aperiodic prototile is ∞-anisohedral.

2.4.1 Isohedral tilings

By definition, an isohedral tiling is bound by a set of geometric constraints: congruences between tiles must be symmetries of the tiling. Grünbaum and Shephard show that those geometric constraints can be equated with a set of *combinatoric* constraints expressing the adjacency relationship a tile maintains across its edges with its neighbours. They prove that the constraints yield a divi-

---

[5]A *polyomino* is the union of a finite set of connected squares from the regular tiling by squares. *Polyiamonds* and *polyhexes* can be defined analogously from the regular tilings by triangles and hexagons. A general introduction to the field is given by Golomb [59], who coined the term; Grünbaum and Shephard discuss polyominoes as tilings of the plane [68, Section 9.4].
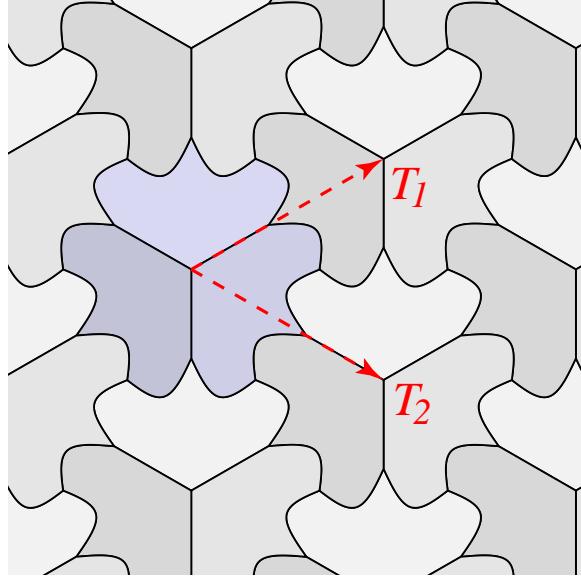
Figure 2.14 An isohedral tiling type imposes a set of adjacency constraints on the tiling edges of a tile. When the bottom edge of the square deforms into the dashed line, the other edges must respond in some way to allow the new shape to tile. The six resulting prototiles tiles here are from six different isohedral types, and show six of the possible responses to the deformation.

sion of the isohedral tilings into precisely 93 distinct *types* or *families*,[6] referred to individually as $IH1, \ldots, IH93$ and collectively as IH [68, Section 6.2]. Each family encodes information about how a tile's shape is constrained by the adjacencies it is forced to maintain with its neighbours. In 12 of these types, the adjacency relationship can only be realized by placing *markings* on tiles that indicate their orientations. We will primarily be concerned with the other 81 types, where the combinatoric structure of the tiling can be expressed geometrically through deformations to the tiling edges. A change to a tiling edge is counterbalanced by deformations in other edges; which edges respond and in what way is dependent on the tiling type, as shown in Figure 2.14. In what follows, we review the classification and notation used with the isohedral tilings.

The *combinatorial structure* of an isohedral tiling $\mathcal{T}$ is an infinite graph whose vertices are the tiling vertices of $\mathcal{T}$, and where two vertices are connected by an edge if the two corresponding tiling vertices are connected by a tiling edge. Two isohedral tilings can then be said to be *combinatorially equivalent* when their combinatorial structures are isomorphic. Combinatorial equivalence partitions the isohedral tilings into eleven classes, referred to as *combinatorial types*, or more commonly as *topological types*.[7] Each topological type has one of the eleven Laves tilings as a distinguished

---

[6]In tiling theory, seemingly arbitrary numbers like 93 are not uncommon; enumerations of families of tilings tend to have sets of constraints that collapse certain cases and fracture others.

[7]The use of the term "topological type" would seem to suggest that the two tilings are topologically, and not combina-

Figure 2.15 An example of an isohedral tiling of type IH16. A single translational unit of the tiling is shown through the two translation vectors T_1 and T_2 and the three coloured aspects.

representative, and we name the type using the vertex symbol of the corresponding Laves tiling. For example, Figure 2.15 shows an isohedral tiling of type IH16. We can see that every tile has six tiling vertices, all of valence three, meaning that IH16 is of topological type 3^6.
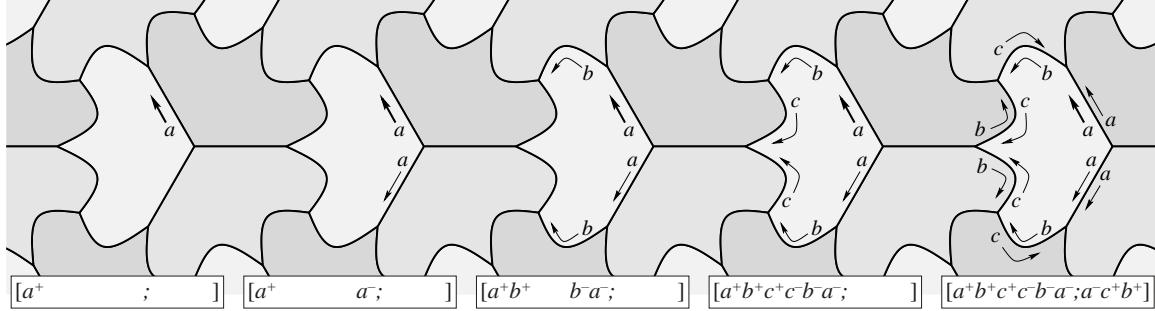
Every isohedral tiling is both monohedral and periodic, meaning that its behaviour over the entire plane can be summarized by specifying the aspects of the single prototile that make up a translational unit, and two linearly-independent translation vectors that replicate that unit over the plane. IH16 has three aspects, shown in varying shades of blue in Figure 2.15. These three tiles comprise one possible translational unit with translation vectors T_1 and T_2.

The adjacency constraints between the tiling edges of a tile are summarized by an *incidence symbol*. Given a rendering of an isohedral tiling, the incidence symbol can be derived in a straight-forward way.

Figure 2.16 shows five steps in the derivation of an incidence symbol for our sample tiling. To

torially equivalent. In fact, for normal tilings the two forms of equivalence are identical [68, Section 4.1] and so both terms are valid.

Figure 2.16 Five steps in the derivation of an isohedral tiling's incidence symbol.

obtain the first part of the incidence symbol, we pick an arbitrary tiling edge as a starting point, assign that edge a single-letter name, and draw an arrow pointing counterclockwise around the tile (step 1). Then, we copy the edge's label to all other edges of the tile related to it through a symmetry of the tiling (step 2). Should the edge get mapped to itself with a reversal of direction, it becomes undirected and is given a double-headed arrow. We then proceed counterclockwise around the tile to the next unlabeled edge (if there is one) and repeat the process (step 3). The first half of the symbol is obtained by reading off the assigned edge names (step 4). A directed edge is superscripted with a sign indicating the agreement of its arrow with the traversal direction. Here, a plus sign is used for a counterclockwise arrow and a minus sign for a clockwise arrow.

The second half of an incidence symbol records how, for each different label, a tiling edge with that label is related to the corresponding edge of the tile adjacent to it. To derive this part of the symbol, we copy the labeling of the tile to its neighbours (step 5). Then, for each unique edge letter assigned in the first step, we write down the edge letter adjacent to it in the tiling. If the original edge was directed, we also write down a plus or minus sign, depending on whether edge direction is respectively preserved or reversed across the edge. A plus sign is used if the arrows on the two sides of a tiling edge are pointing in opposite directions, and a minus sign is used otherwise. For the running example, the incidence symbol turns out to be $[a^+b^+c^+c^-b^-a^-; a^-c^+b^+]$. Note that the incidence symbol is not unique; edges can be renamed and a different starting point can be chosen. But it can easily be checked whether two incidence symbols refer to the same isohedral type.

Every isohedral type is fully described in terms of a topological type and an incidence symbol.

Enumerating all possible topological types and incidence symbols and eliminating the ones that do not result in valid tilings or that are trivial renamings of other symbols leads to the classification given by Grünbaum and Shephard.

2.4.2   Beyond isohedral tilings

Since the work of Grünbaum and Shephard on the classification of isohedral tilings of the Euclidean plane, other tiling theorists have gone on to search for generalizations to related tilings. In particular, a group led by Dress, Delgado-Friedrichs, and Huson pioneered the use of *Delaney symbols* in the study of what they call *combinatorial tiling theory* [33, 34, 86]. A Delaney symbol completely summarizes the combinatorial structure of a k-isohedral tiling of the Euclidean plane, the hyperbolic plane, or the sphere. They can also be generalized to tilings in spaces of dimension three and higher. Delaney symbols form the basis for an efficient software implementation, and Delgado-Friedrichs and Huson have created **2dtiler**, a powerful tool for exploring, rendering, and editing tilings from their combinatorial descriptions.

Combinatorial tiling theory does not play a direct role in the present work (although it is used in a classification by Dress that motivates the technique of Section 4.6.2). Nevertheless, Delaney symbols have helped to advance tiling theory beyond the material of *Tilings and Patterns*, and it seems likely that adopting them as a standard description of all-over tilings could lead to principled k-isohedral and non-Euclidean generalizations of the algorithms and data structures presented in Chapter 4.

2.5   Coloured tilings

Up to now, we have ignored the possibility of colouring tiles in a tiling. When analyzing the symmetries of a tiling, we have treated colour as superficial, to be disregarded when deciding whether two tiles are "the same." It is also possible to take colour into account, adding a layer of richness and complexity to a tiling. The colouring can have a great deal of structure, particularly when it acts compatibly with the symmetries of the tiling. Coxeter gives a group-theoretic presentation of colouring, using Escher's tilings as motivating examples [24]. Grünbaum and Shephard provide an extensive account of the relationship between colouring and tilings [68, Chapter 8]. We restate two

important definitions here.

A *k-colouring* of a tiling is a function from tiles to the set $\{1, \ldots, k\}$ that assigns an abstract "colour" to each tile.[8] That colouring is a *perfect colouring* if every symmetry of the tiling acts as a permutation of the colours. Symbolically, let σ be a symmetry of a tiling $\mathcal{T}$ with colouring $c : \mathcal{T} \to \{1, \ldots, k\}$. The rigid motion σ maps every tile $T \in \mathcal{T}$ to some tile $\sigma(T)$. Then the colouring c is a perfect colouring if for any symmetry σ, there exists a permutation ρ of the set $\{1, \ldots, k\}$ such that $c(\sigma(T)) = \rho(c(T))$ for all $T \in \mathcal{T}$.

Escher studied colourings of tilings in depth while preparing his notebook drawings. He paid great attention to the question of colouring, expressing as a clear objective that adjacent tiles should have contrasting colours to better distinguish them from each other [50]. In general, he aimed to achieve this contrast with a minimal number of colours. Yet his intuition seems to have guided him to the perfect colourings, in some cases choosing a perfect colouring with more colours over a non-perfect one with fewer. A clear example is symmetry drawing 20 [124, Page 131], where a tiling coloured perfectly by four colours is accompanied by a note mentioning that three would have sufficed to distinguish adjacent tiles. Shephard points out that for this tiling, no perfect colouring is possible with only three colours [126]. Escher intuited that a fourth colour allowed for a more regular colouring. Four colours suffice to perfectly colour any isohedral tiling in such a way that adjacent tiles have contrasting colours.

Escher's understanding of the compatibility between a tiling's symmetries and its colouring predated the development of a formal theory of colour symmetry, and to some extent set that development in motion [123]. For while a small amount of mathematical work had been done on the subject previously, it was when the crystallography community became aware of Escher's tesselations that they understood how much of a theory there was to be had, and they were provided with a rich library of illustrations from which to build that theory.

---

[8]This definition of k-colouring should be distinguished from its use in problems of graph colouring and map colouring, where there is an additional restriction that adjacent vertices or regions have different colours (as in the four colour theorem). A k-colouring does not require adjacent tiles to have different colours, although we will discuss this additional restriction as well.

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 vari-

ety 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 math-

ematically, 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 re-weaving 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.
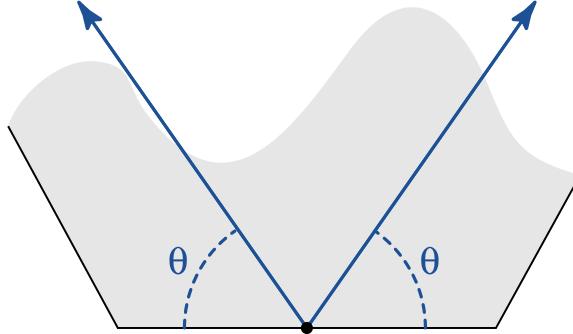
I have developed an interactive Java program that implements what I call "Hankin's Method." At its core is an "inference algorithm": a subroutine that takes a polygonal tile and a contact angle as input and returns a motif for that tile. The inference algorithm runs quickly enough that the contact angle 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
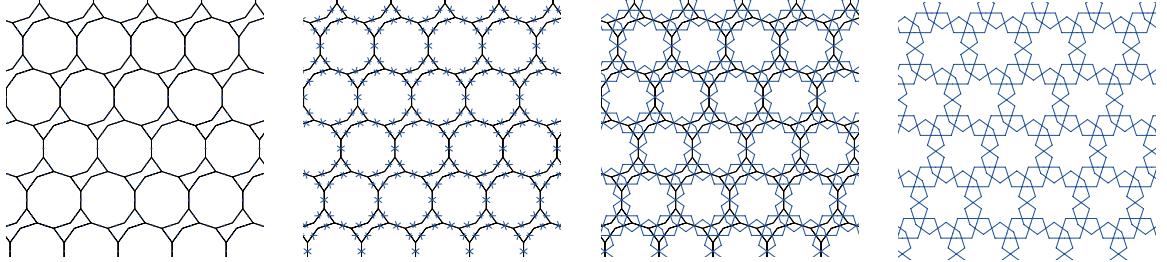
Figure 3.1 In the first step of Hankin's method, a pair of rays is associated with every contact position on every tile. Here, a single contact position gets its two rays, each of which forms the contact angle θ with the contact edge.

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.
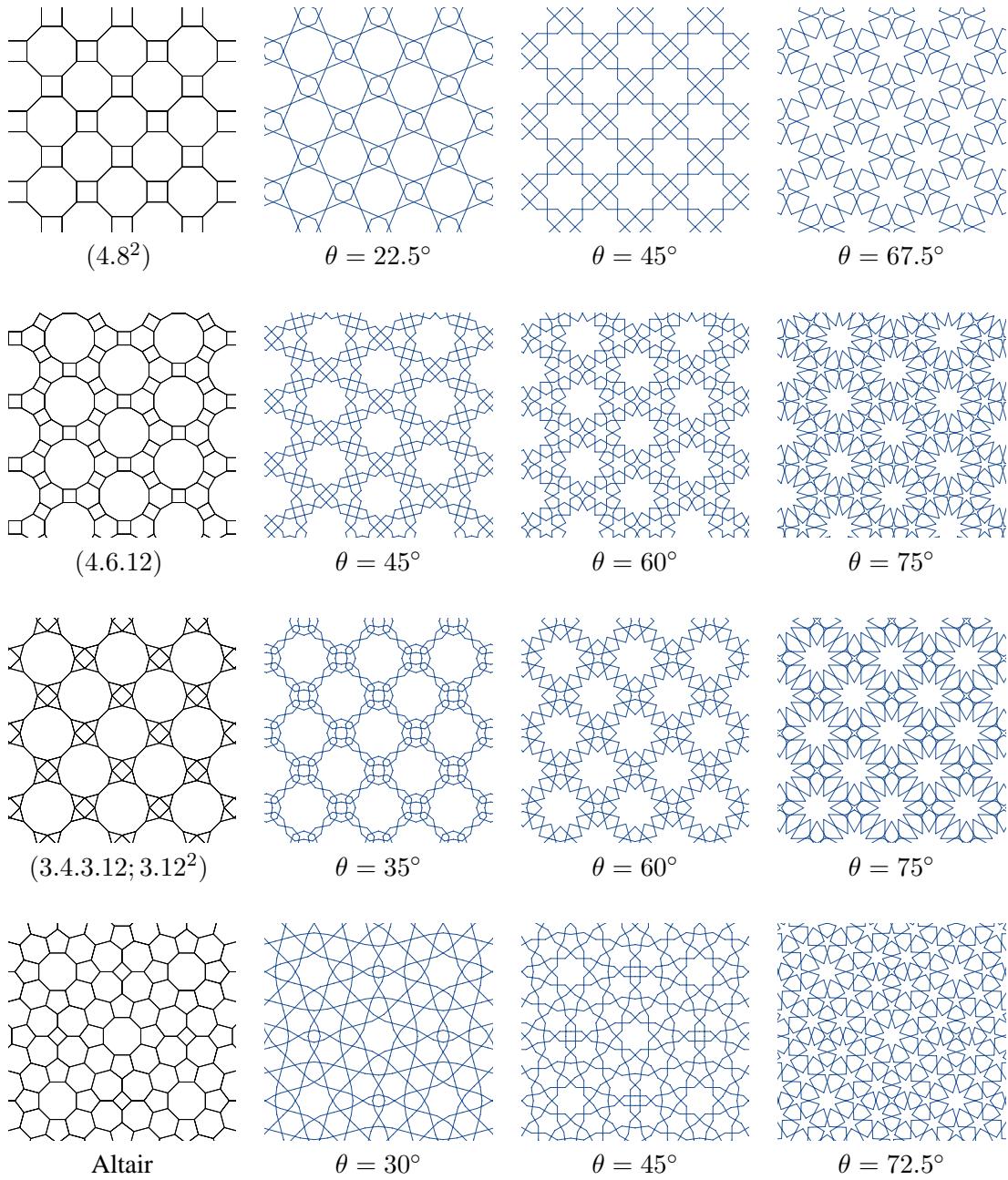
As a first attempt at inference, we might then proceed as follows. Iterate over all possible pairings, evaluating the total cost of each. From among all pairings that use the most possible rays, choose the one with the lowest total cost. Unfortunately, this algorithm is not practical. In an n-sided tile, 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 $\frac{n!}{(n/2)!2^n} = 1 \cdot 3 \cdot 5 \cdot 7 \dots (n-1)$. 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.
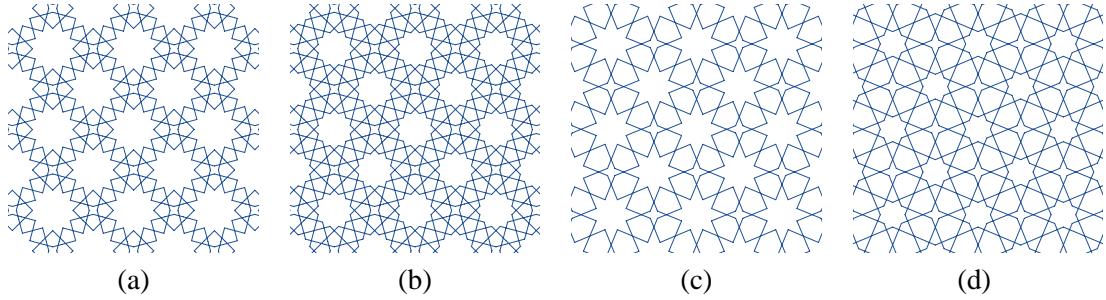
In practice, this algorithm performs very well on many of the typical tile shapes that arise in template tilings. It certainly performs perfectly on regular polygons, where it constructs star-shaped motifs. In the cases where it fails, it usually does so not because it is greedy, but because the underlying 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	$\theta = 30°$	$\theta = 45°$	$\theta = 72.5°$
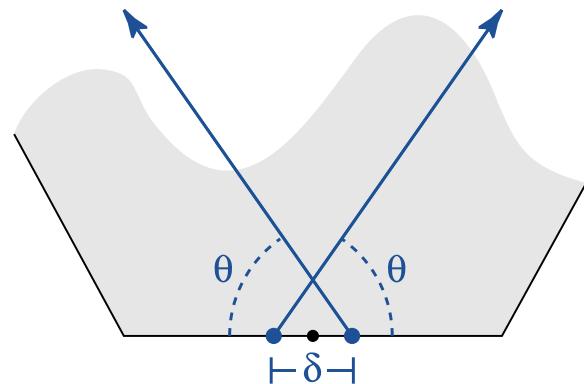
Figure 3.3 Examples of star patterns constructed using Hankin's method. Each row shows a template tiling together with three designs that can be derived from it using three different contact angles. The bottom row features an amusing tiling by nearly regular polygons. It is reproduced from Grünbaum and 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].
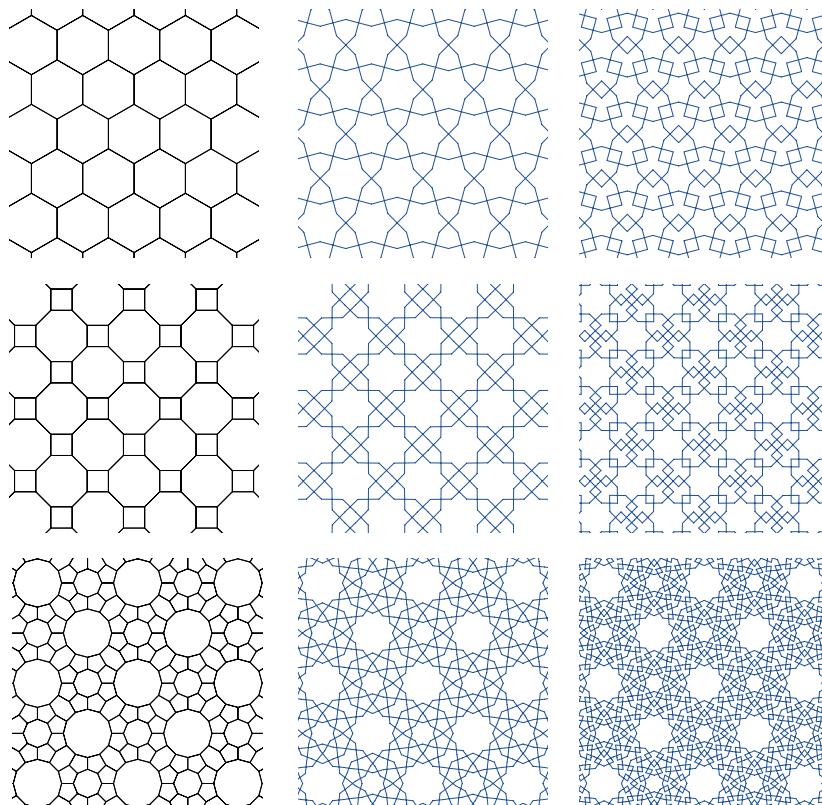
Figure 3.4 A demonstration of two cases where an extension to the inference algorithm can produce a slightly more attractive motif. In (a), a star pattern is shown with large unfilled areas that were the centers of regular dodecagons in the template tiling. Adding an additional layer of inferred geometry 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.
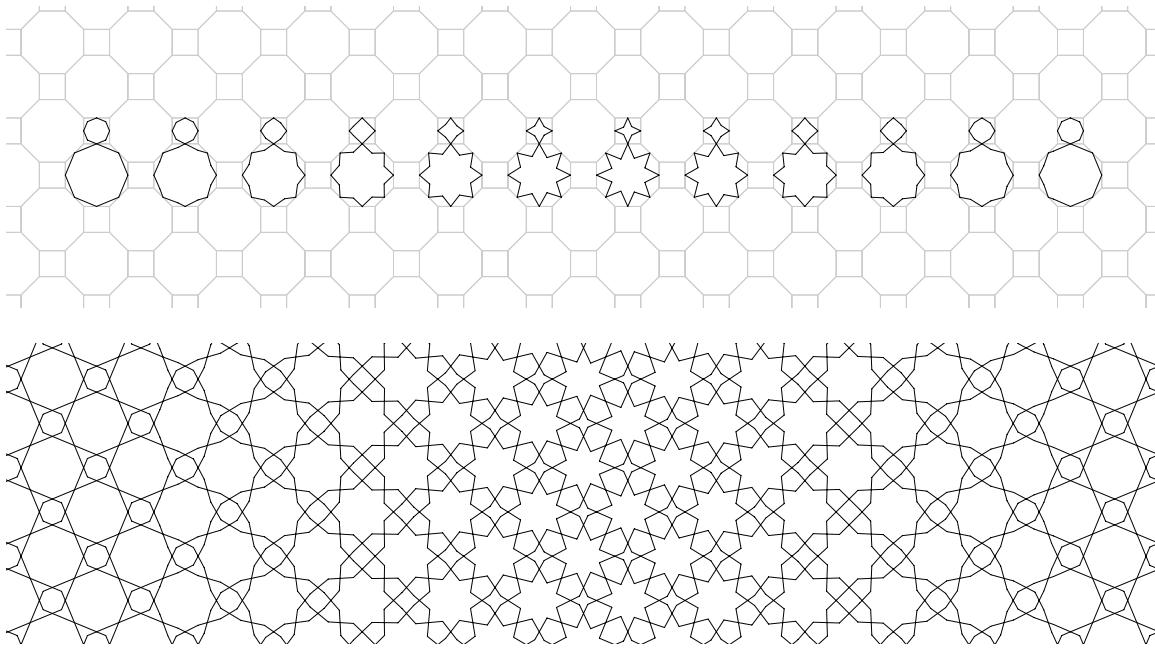
Figure 3.6 Examples of two-point star patterns constructed using Hankin's method. Each row shows a template tiling, a star pattern with $\delta = 0$, and a related two-point pattern with non-zero δ. The structure of the tiling in the bottom row will be explained in Section 3.8.

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.
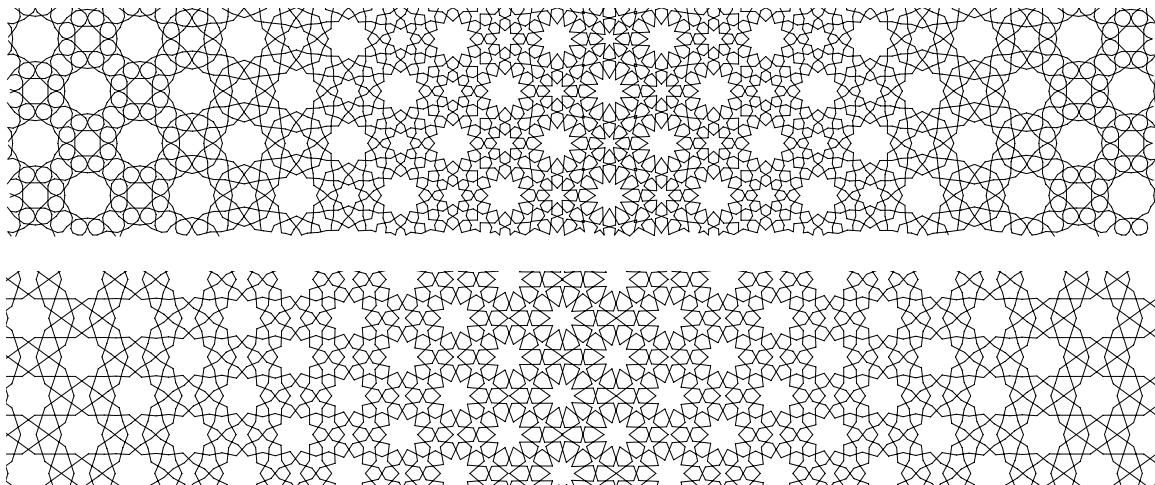
3.4.1   Islamic parquet deformations

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.
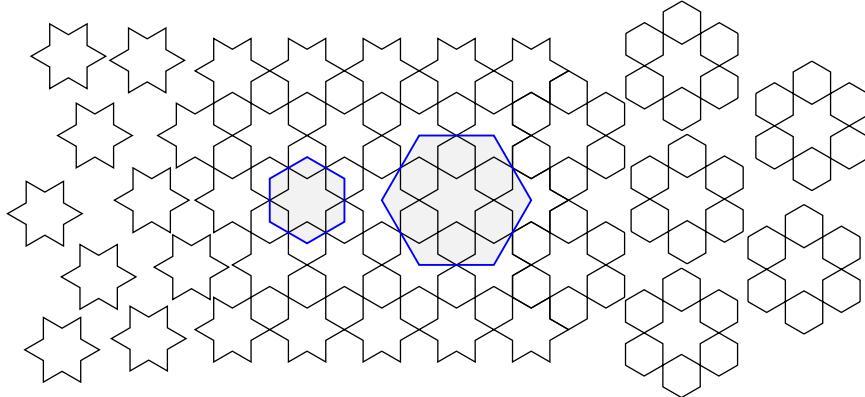
Hankin's method can be used as the basis for a simple but highly effective method of constructing Islamic patterns in the spirit of parquet deformations. I lay out a strip of the template tiling and then run a modified inference algorithm where the contact angle at every contact point is determined by 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.
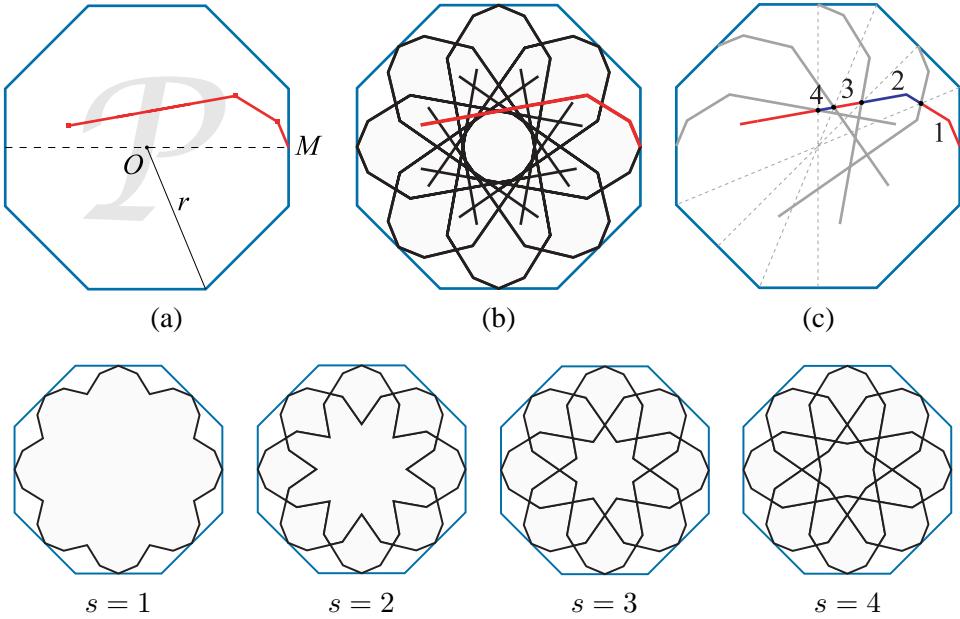
The design elements can take the place of the default inference algorithm in the construction of motifs for regular polygons. I combine the design elements and the inference algorithm in an interactive tool called **Taprats** that has been used successfully to design, render, and execute many traditional 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

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 imple- menation 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.
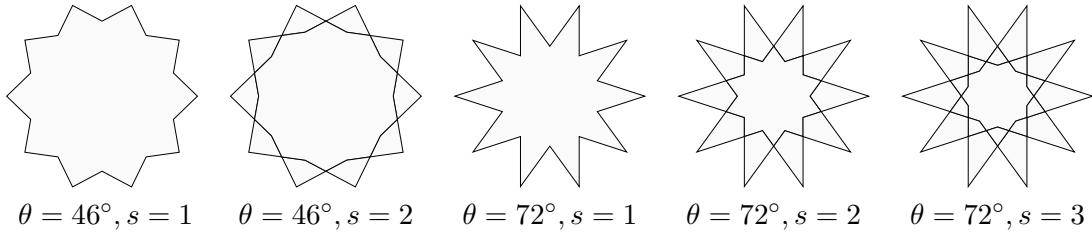
Grünbaum and Shephard [68, Section 2.5] show how star polygons may be specified using the notation $\{n/d\}$, where n and d are integers, $n \geq 3$, and $1 \leq d < \lceil n/2 \rceil$. The star is constructed by placing points $v_1, \ldots, v_n$ at the vertices of a regular n-gon, and joining every 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.
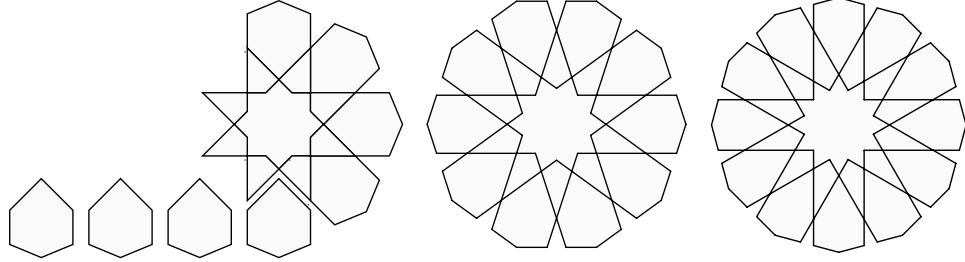
In Taprats, a star is constructed from a path consisting of a single line segment that effectively acts as a ray. The segment begins at M and has length $2r$. It is parameterized by a single degree of freedom, the contact angle θ. Some examples of stars constructed this way are given in Figure 3.11.

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.

$$\theta = 46°, s = 1 \qquad \theta = 46°, s = 2 \qquad \theta = 72°, s = 1 \qquad \theta = 72°, s = 2 \qquad \theta = 72°, s = 3$$

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 $\angle ACG = \angle AMG$.
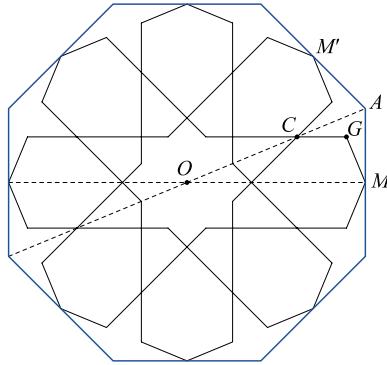
Before generalizing Lee's construction, let α be the "natural" height of point G in the ideal rosette, as a fraction of the surrounding polygon's side length. Referring to Figure 3.14(a), let point H be the projection of G onto segment $\overline{OM}$. Then $\alpha = GH/AM$. The value α provides a scale-independent 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.
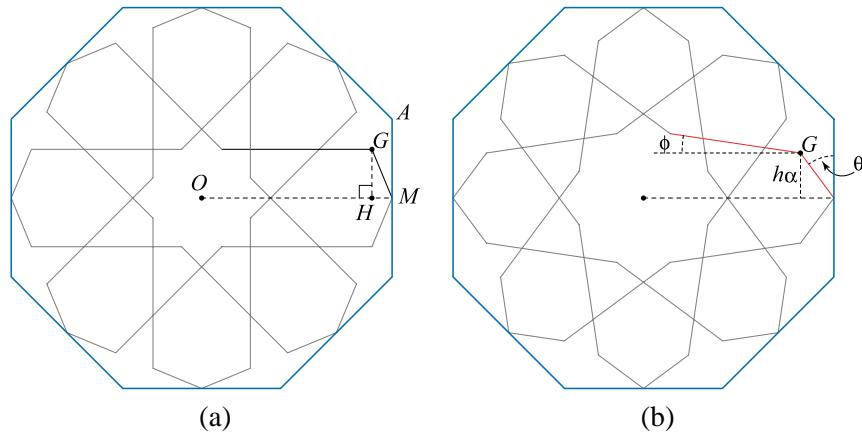
Every rosette contains a central star. As was mentioned, in constructing stars there is usually a natural value for s, determined from n and θ. The same is true here: the choice of s can be obtained by determining the correct value for the central star and adding 1 to account for the additional geometry 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
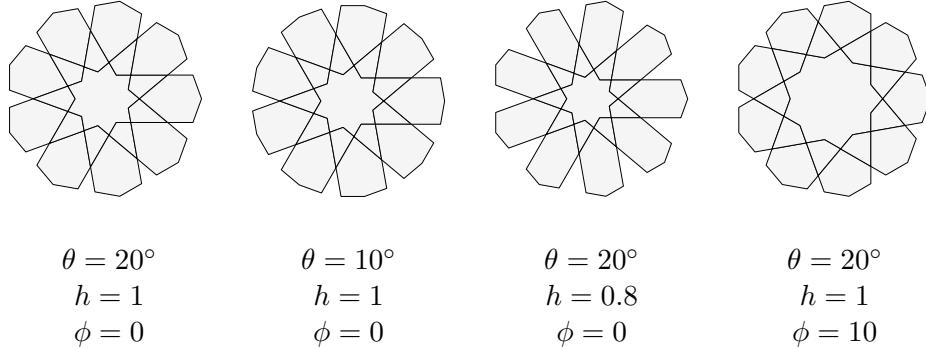
Figure 3.13 A diagram used to explain the construction of Lee's ideal rosette [99]. The construction is explained in Section 3.5.3.

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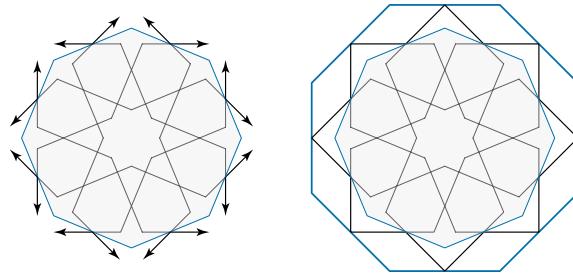
Figure 3.14 Two diagrams used to explain the construction of generalized rosettes as presented in Section 3.5.3.

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

$$\theta = 20° \qquad \theta = 10° \qquad \theta = 20° \qquad \theta = 20°$$
$$h = 1 \qquad h = 1 \qquad h = 0.8 \qquad h = 1$$
$$\phi = 0 \qquad \phi = 0 \qquad \phi = 0 \qquad \phi = 10$$

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.
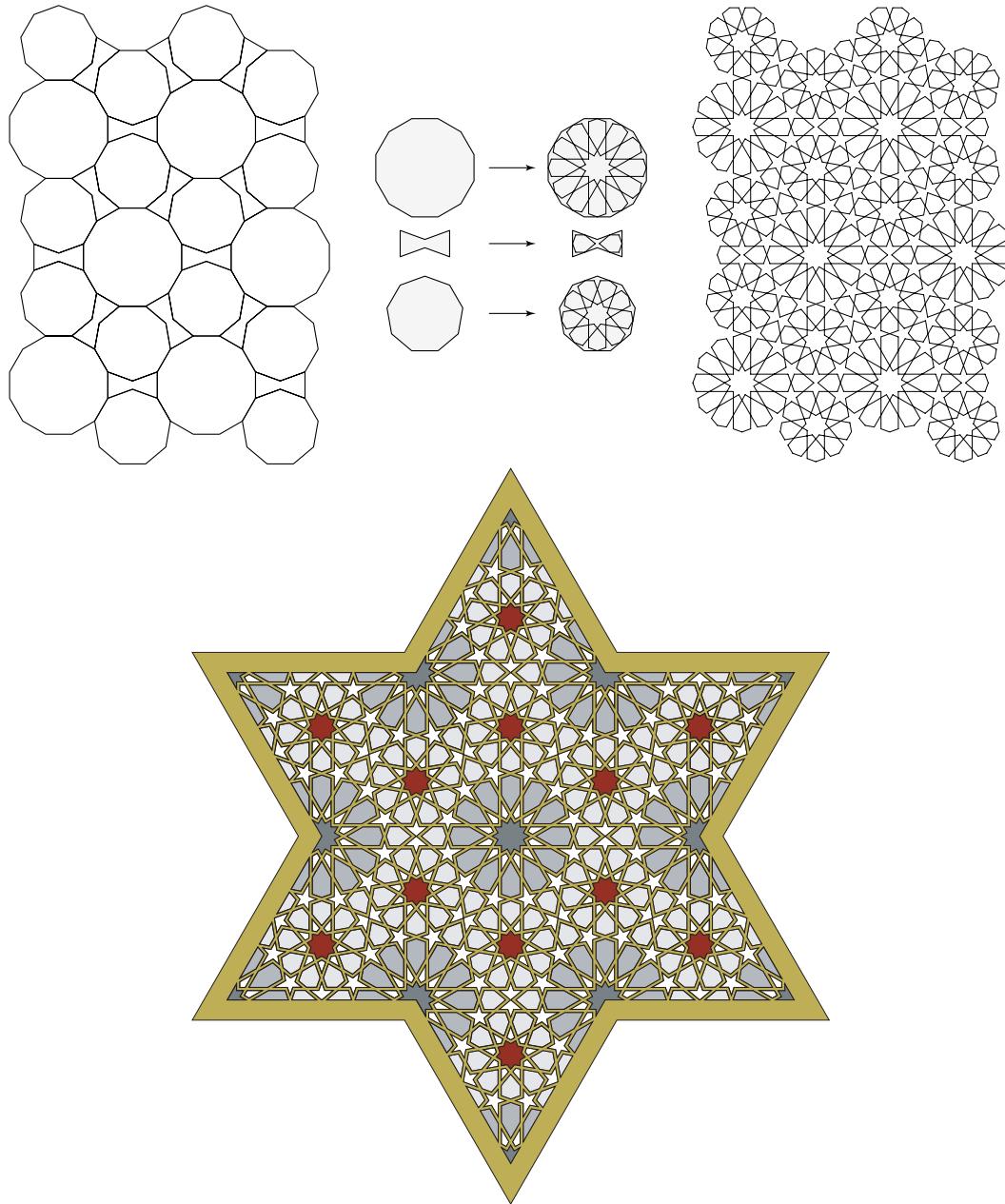
3.5.5 *Taprats*

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.
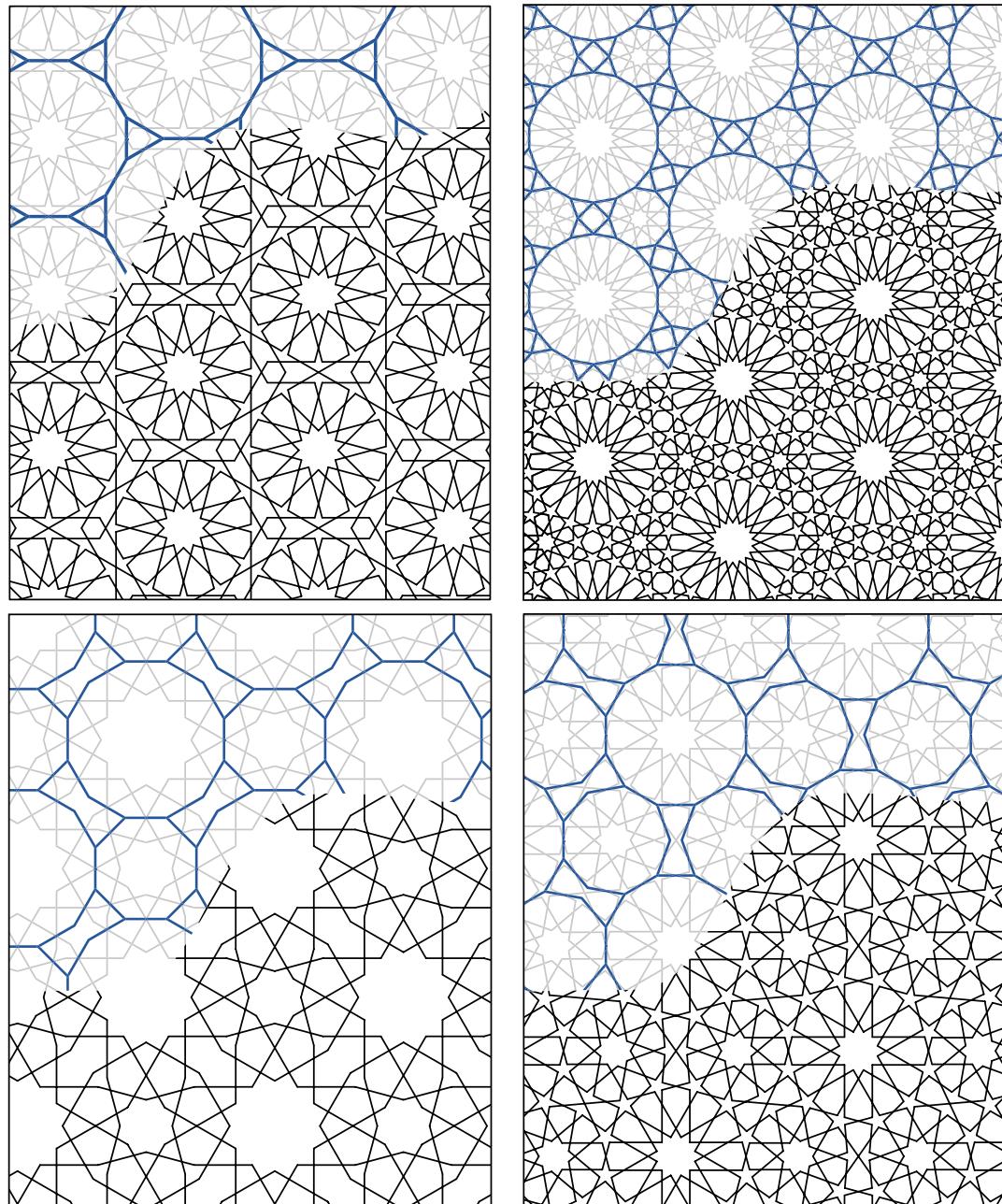
Given a periodic tiling, Taprats presents a user inferface featuring an editor for the motifs in each tile shape. For regular polygons, that editor allows the user to choose from among stars, rosettes, and extended rosettes, and to edit the degrees of freedom in the design elements directly. For the 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).
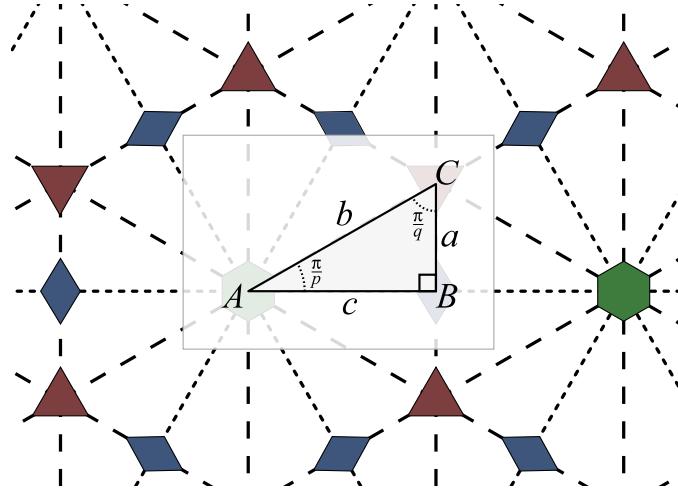
In this section, I present my family of tilings using only facts from absolute geometry. I then give revised design elements that have all reliance on parallelism removed. This generalized technique is then implemented in a C++ program called **Najm** (pronounced **Nazhm**, arabic for "star"). I discuss 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.
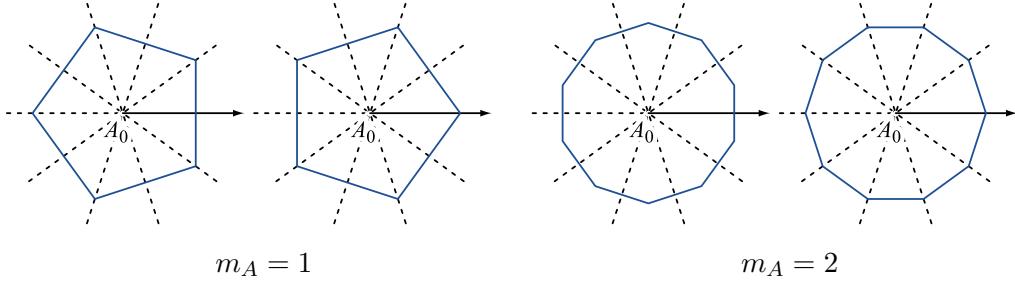
Figure 3.19 The canonical triangle used in the construction of Najm tilings.

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
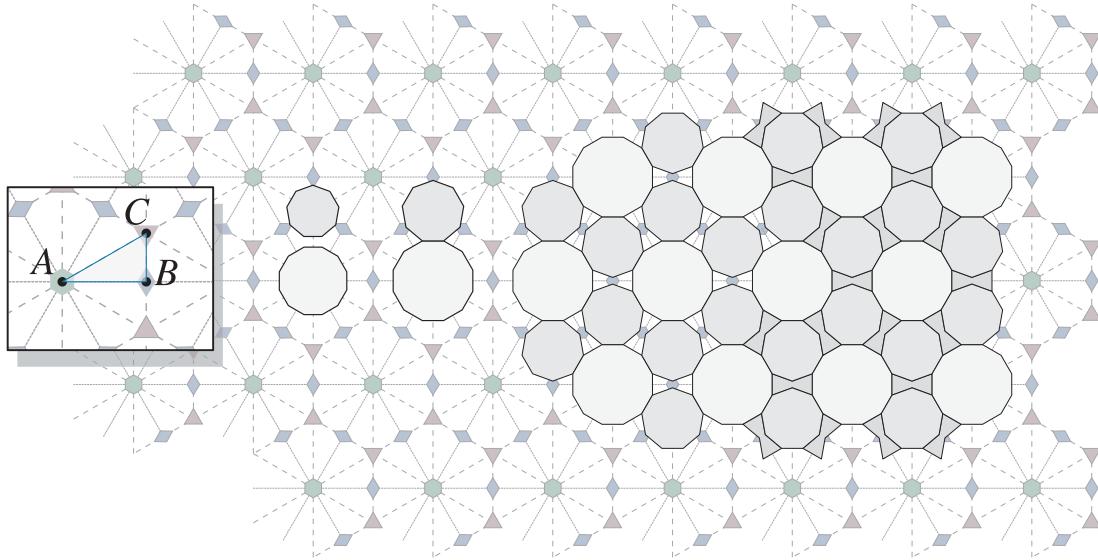
Figure 3.20 Examples of valid orientations for on-axis polygons around a fivefold rotational axis. The first and third examples have edge midpoints lying on the designated ray (marked by an arrow). The second and fourth have vertices on the ray. The notation $o_A = \mathsf{e}$ and $o_A = \mathsf{v}$ 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 $([6, 3]; 2\mathsf{e}, 0, 3\mathsf{e}; AC)$, 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 $o_A = \mathsf{e}$, these dodecagons will be oriented so that an edge midpoint lies on the ray $\overrightarrow{AB}$. We also have $m_C = 3$ and $o_C = \mathsf{e}$, 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 $[p, q]$.
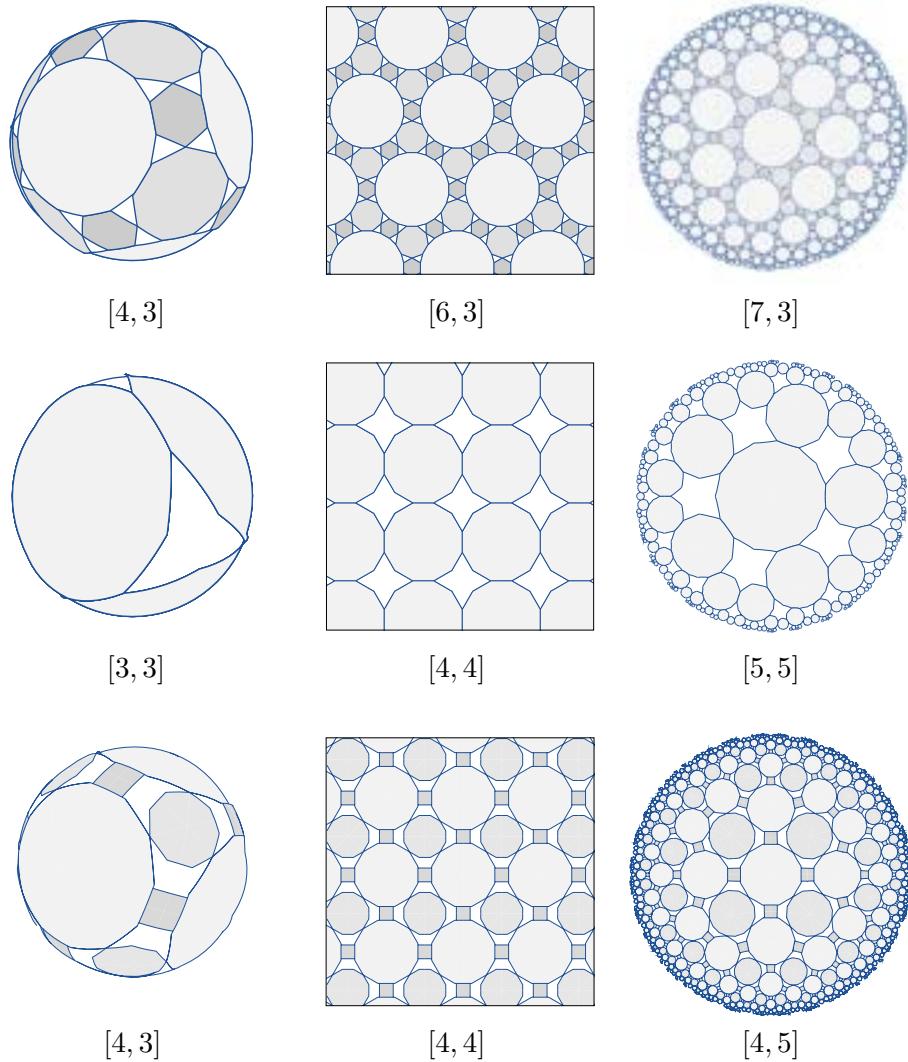
At the start of this section, I pointed out the relationship between polyhedra and patterns in spherical geometry. When p and q are chosen to yield a spherical symmetry group in the notation above, many of the resulting tilings can easily be converted into polyhedra simply by taking the convex hull of 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*
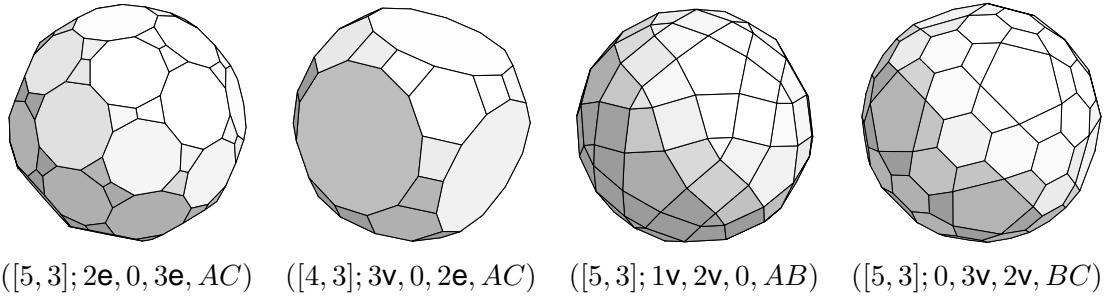
Now that I have a geometry-neutral source of tilings to use in the creation of Islamic star patterns, we must reexamine the way motifs are generated to make sure that the same algorithms apply in absolute geometry. Certainly there is no problem expressing planar maps in absolute geometry; points and 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.

$([5,3]; 2\mathsf{e}, 0, 3\mathsf{e}, AC)$ $([4,3]; 3\mathsf{v}, 0, 2\mathsf{e}, AC)$ $([5,3]; 1\mathsf{v}, 2\mathsf{v}, 0, AB)$ $([5,3]; 0, 3\mathsf{v}, 2\mathsf{v}, BC)$
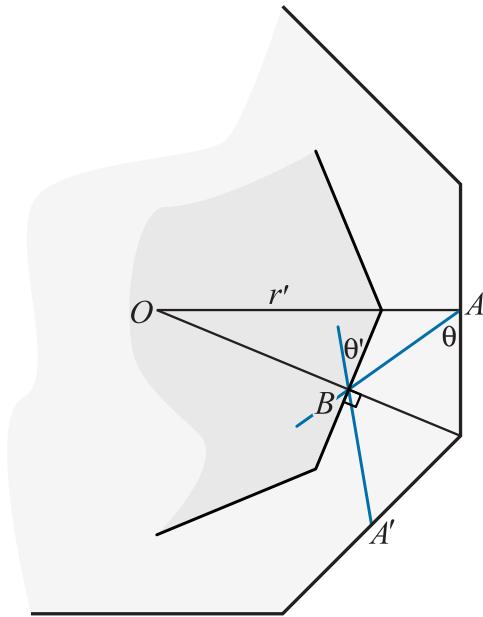
Figure 3.23 Examples of symmetrohedra, symmetric polyhedra based on the tilings described in Section 3.6.1. Each solid is derived from its given tiling symbol by building the spherical tiling and taking the convex hull of the tiling vertices.

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;
}
```
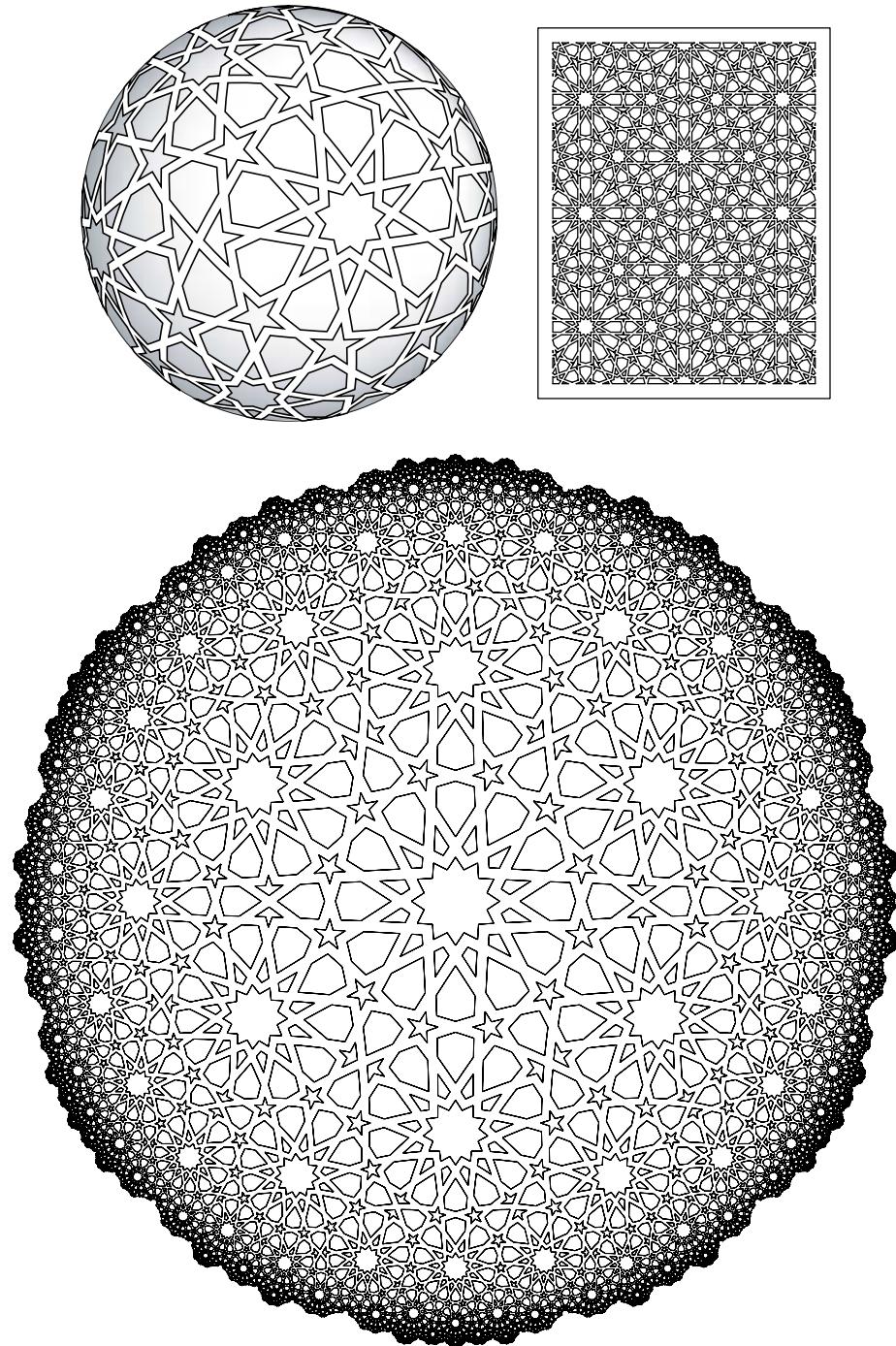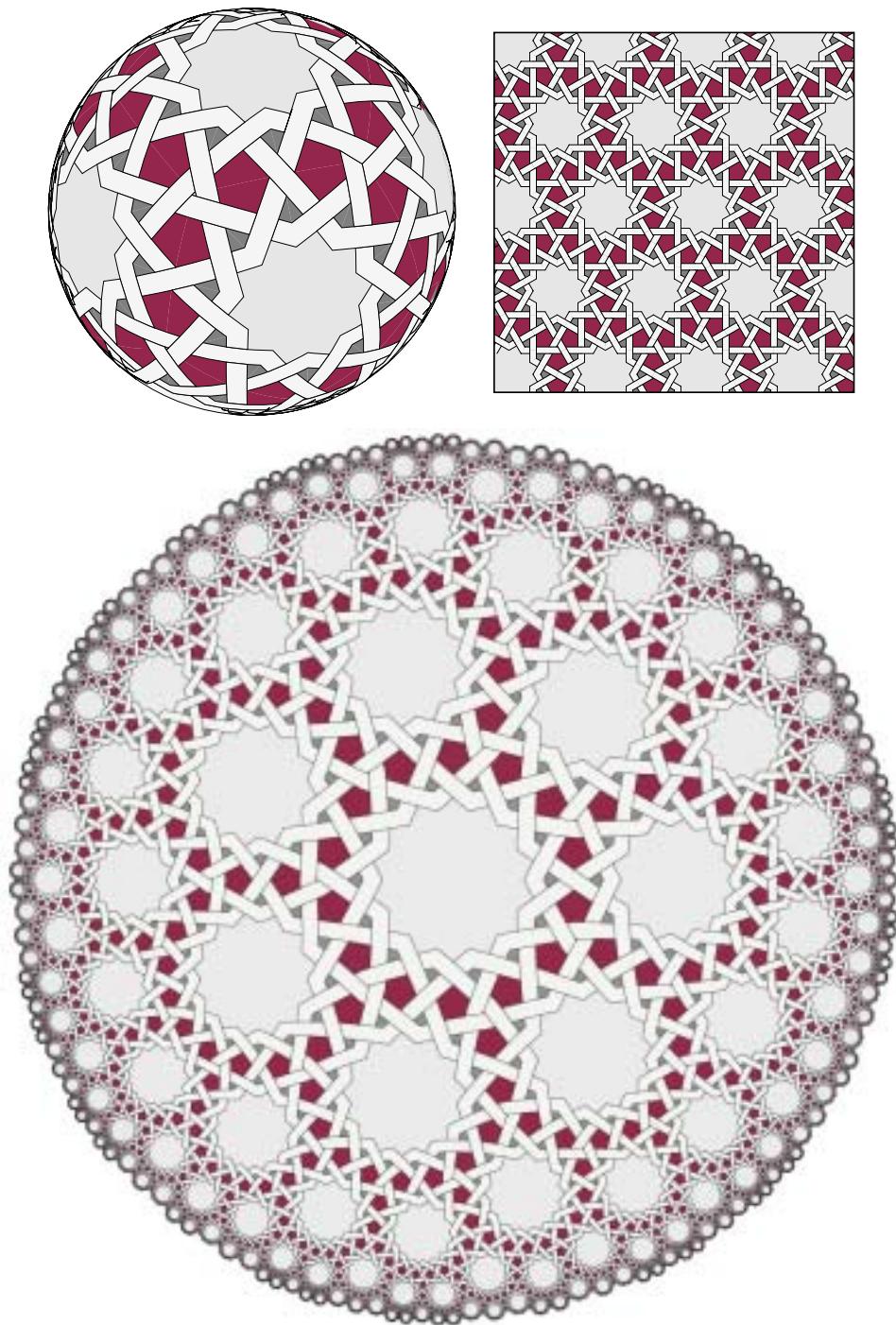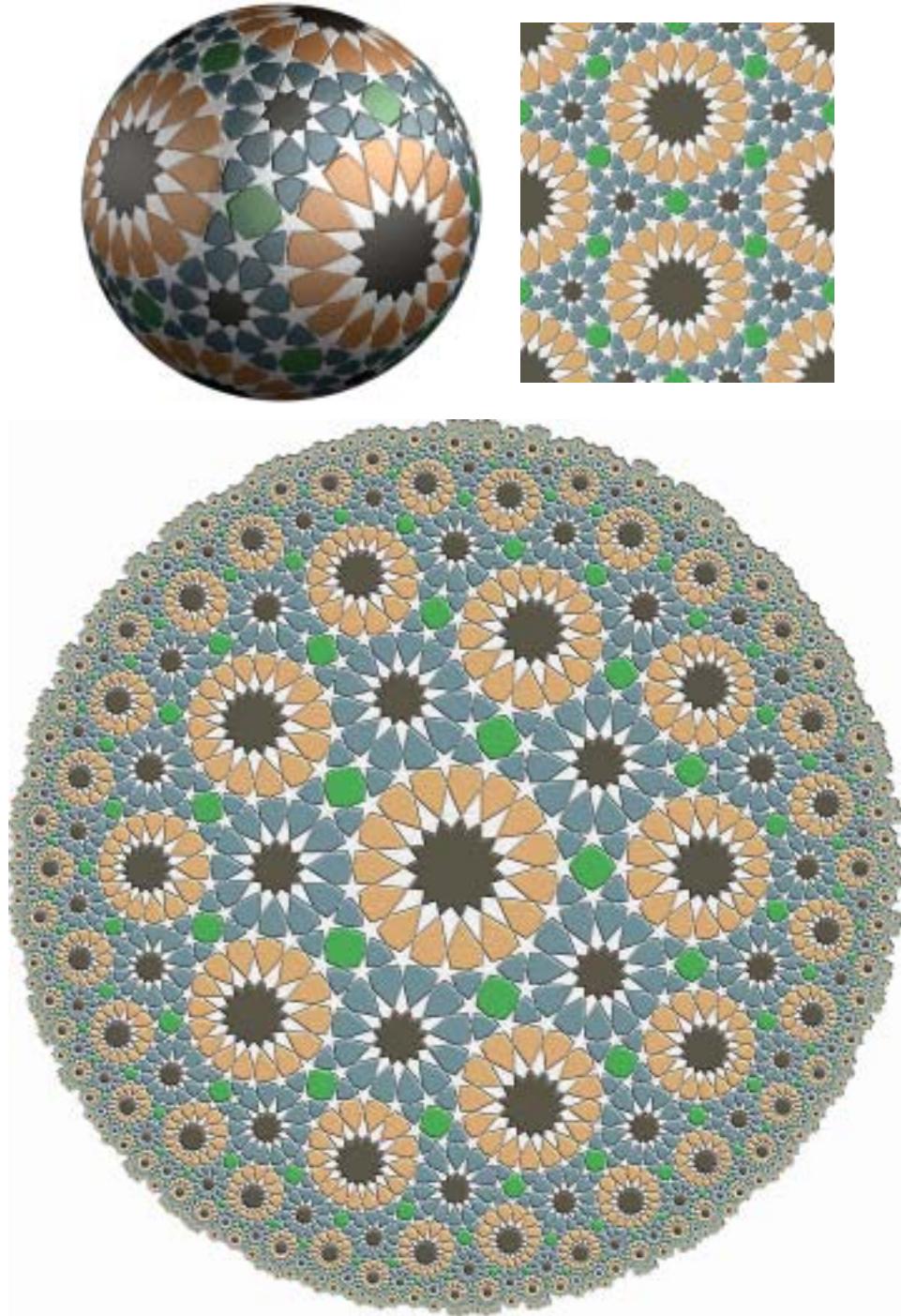
Figure 3.25 An excerpt from the absolute geometry library underlying Najm, showing the class specialization technique. C++ templates allow code to be parameterized over a choice of geometry without incurring the runtime overhead of a class hierarchy. Here, I demonstrate the declaration of a simple point class, together with its specializations and a sample client function.
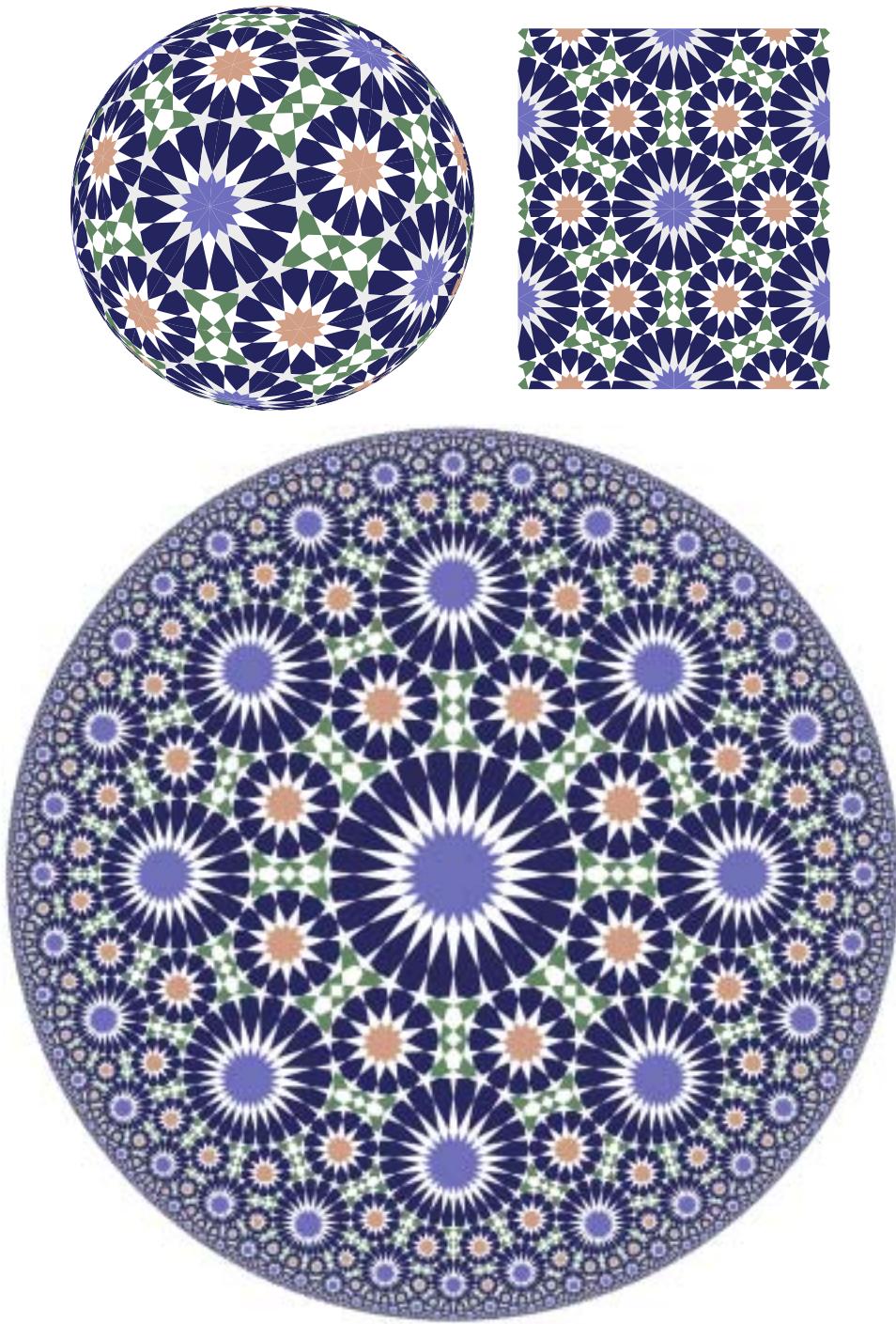
Figure 3.26 Samples of Islamic star patterns that can be produced using Najm. To provide a basis for comparing patterns across geometries, each page presents a single conceptual design interpreted in each of the three different geometries.

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 by-product 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
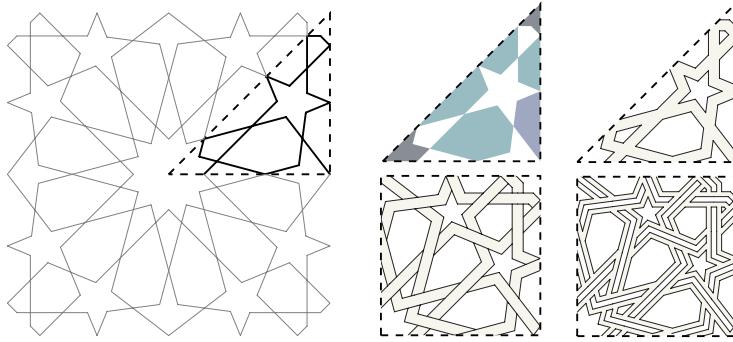
Examples of rendered star patterns in the three planar geometries are given in Figure 3.26. In each case, a single choice of motifs and rendering style is applied across related Euclidean, hyperbolic, and spherical symmetry groups. This consistency makes clear the similarity between related symmetry 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.
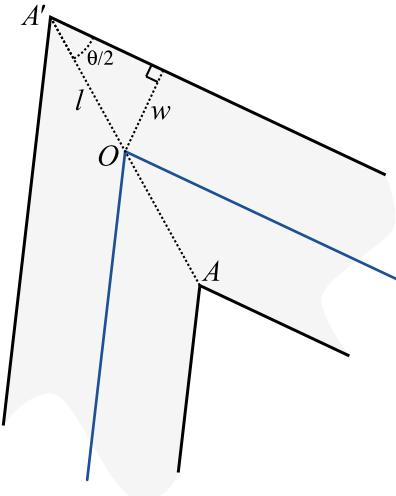
3.7.1 Filling

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
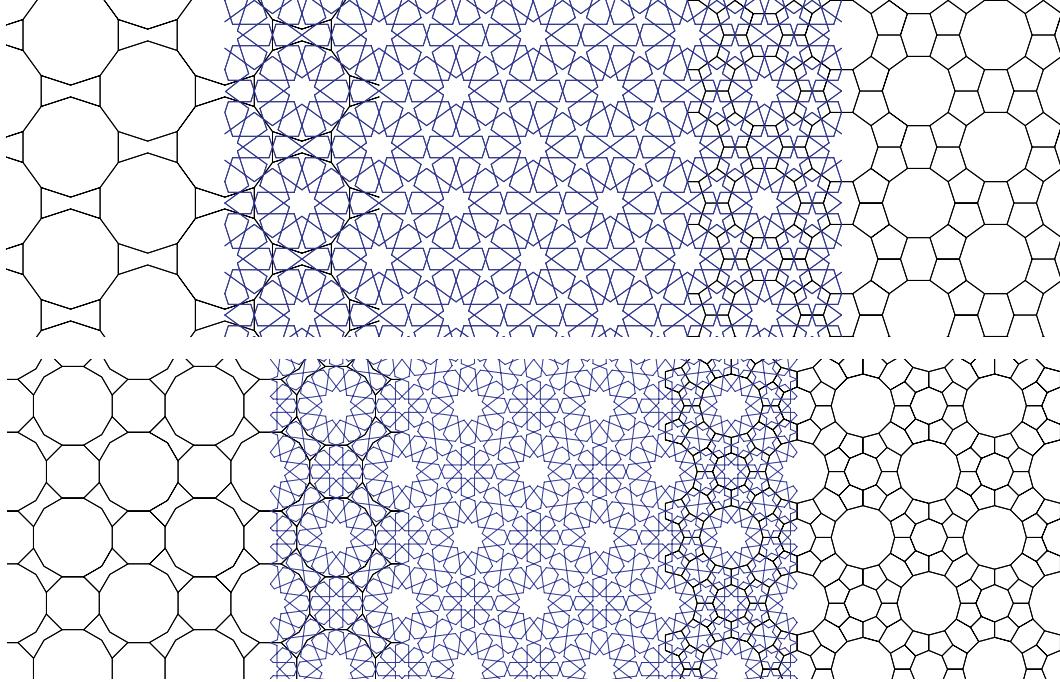
In practice, designs are rendered using some combination of the styles above. The most common combinations are the superposition of an outlined or interlaced rendering over a filled rendering.

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
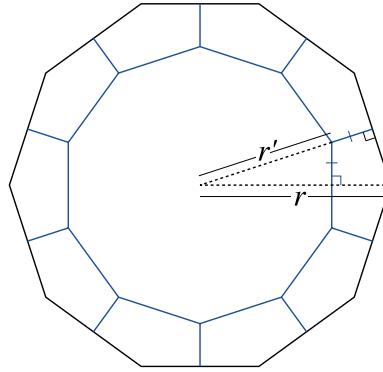
Figure 3.29 Examples of distinct tilings that can produce the same Islamic design. In each case, the tilings on the left is filled in using a combination of design elements and inference, and the tiling on the right uses inference alone. They meet in the shared design in the center.
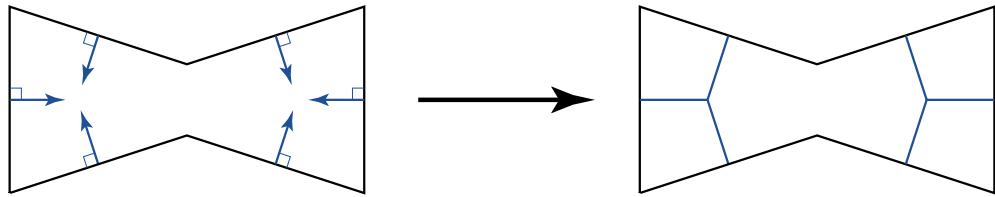
may originate from two very different tilings (he makes this observation in the context of the tilings in the top row of Figure 3.29). In fact, there is a deep connection between the two tilings, one that should not be dismissed as coincidence. In this section, I explain the relationship between Hankin 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.
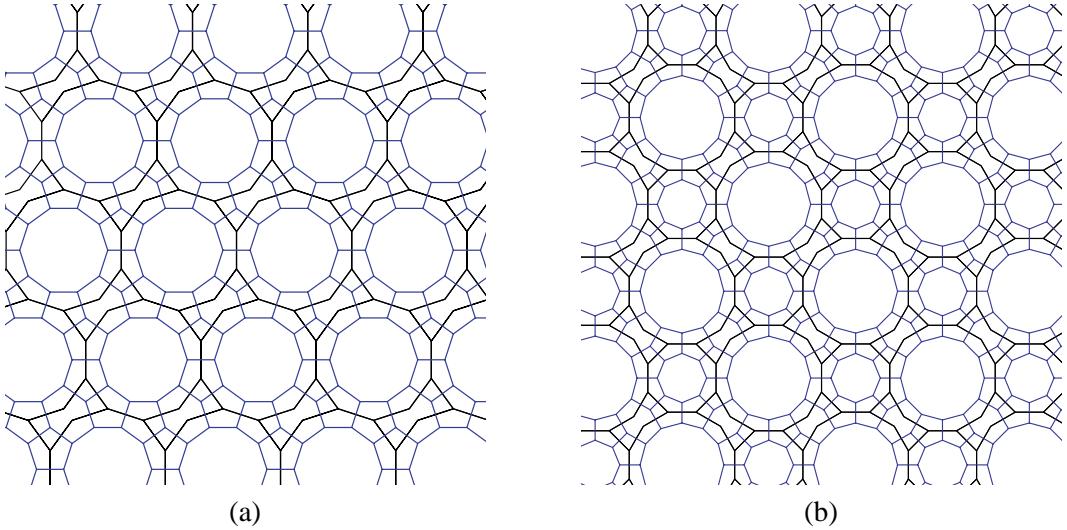
Figure 3.31 The rosette transform applied to an irregular polygon. On the left, a perpendicular bisector is drawn for every tile edge as a ray pointing to the interior of the tile. The rays are cut off when they meet each other, as with the inference algorithm.

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
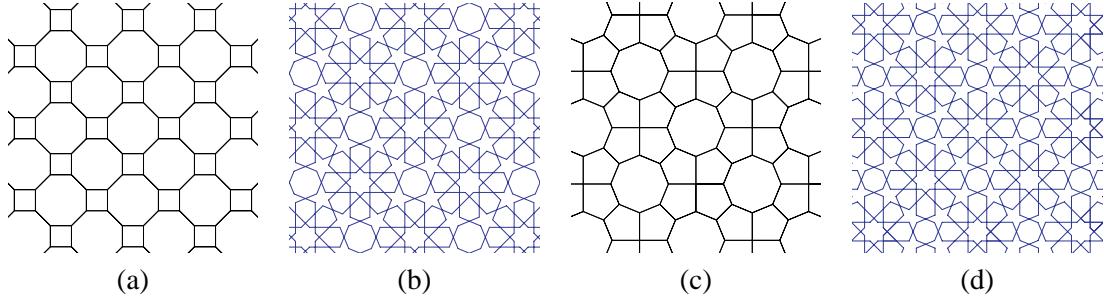
transform of the bottom row of Figure 3.29 in (b).

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

(a)   (b)   (c)   (d)
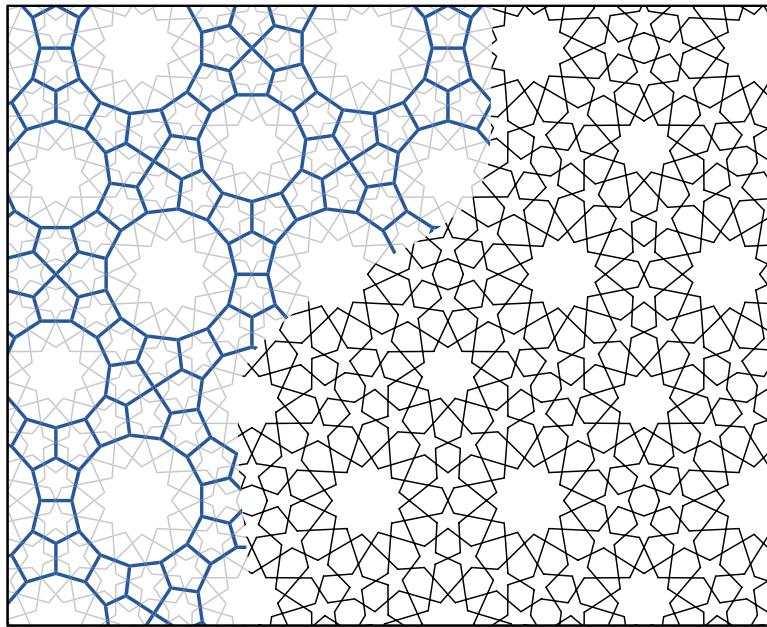
Figure 3.33 An example where the simpler tilings of the Taprats method, combined with more complex design elements, can produce a better overall design than the more complex tilings of Hankin's method. The tiling in (a) produces the design in (b) by placing rosettes in the octagons. The corresponding 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
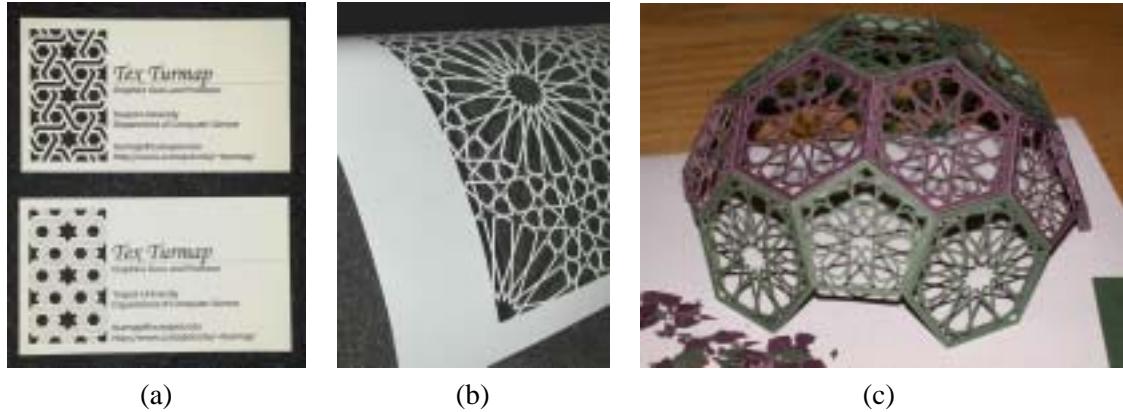
Figure 3.34 An unusual star pattern, reproduced from Bonner's manuscript, featuring 11-
and 13-pointed stars.

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

(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.
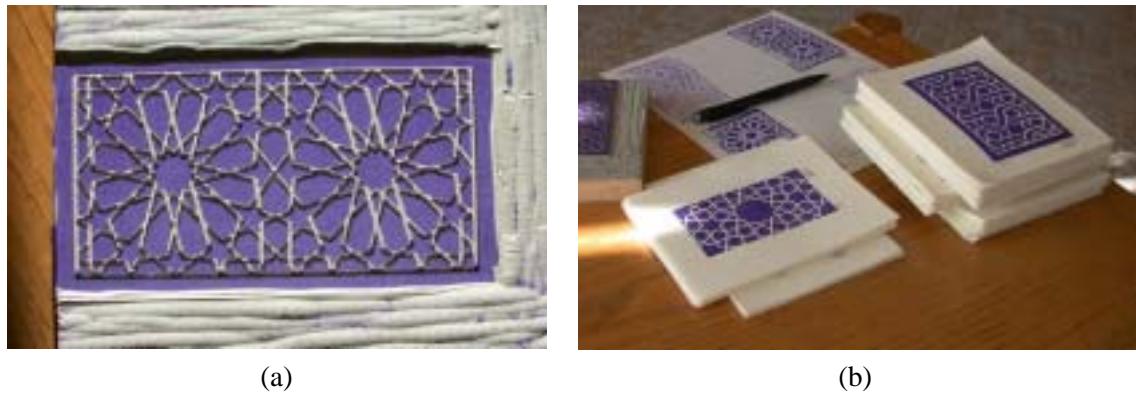
CNC milling machines have been around for some time. Here, the cutting tool is a drill bit, allowing the milling machine to automate many of the tasks performed by hand-operated power tools. Figure 3.36 shows an example of the use of the milling machine. Using a router bit, I cut star patterns into blocks 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.
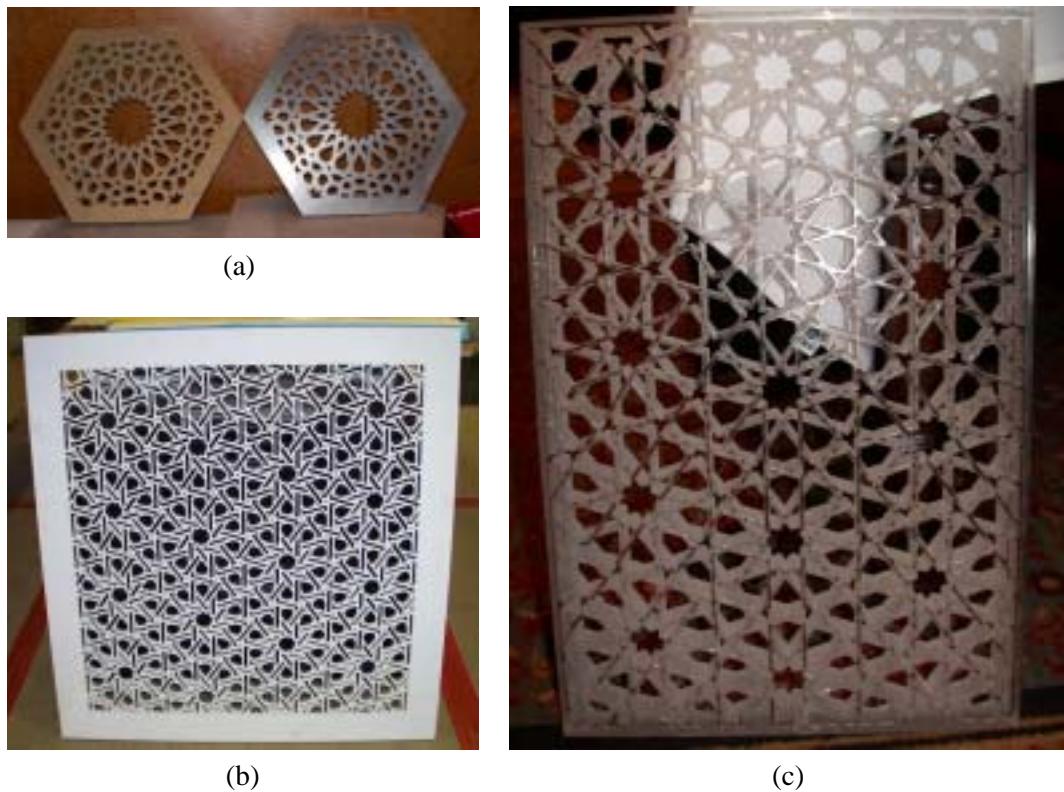
At the high end of the plotting devices is the waterjet cutter. These machines cut by directing a high-pressure stream of grit-impregnated water at the material. They have an incredible range, able to cut through materials ranging in toughness from foam to titanium, even able to handle delicate materials 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.
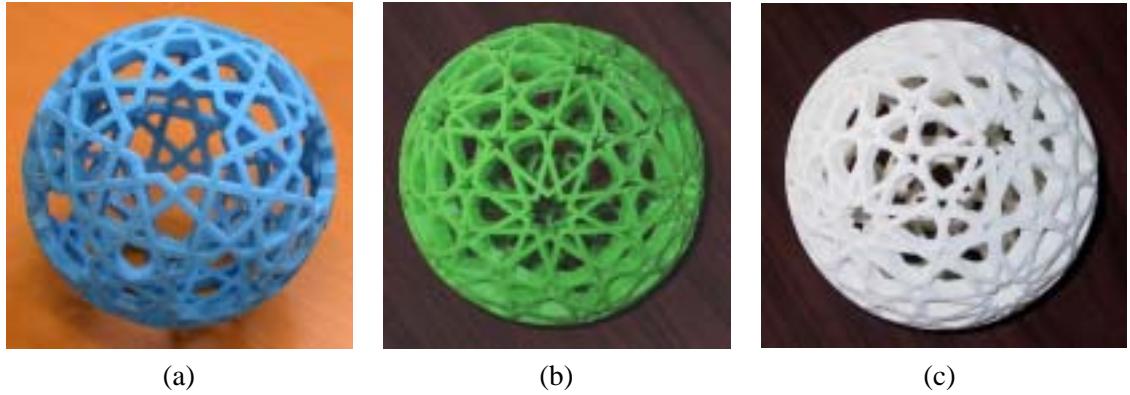
Rapid prototyping machines are not especially practical for executing Euclidean patterns; these are much more effectively manufactured using any of the plotting machines described above. But they are excellent at fabricating spherical patterns, which otherwise would be very difficult to machine or carve 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.
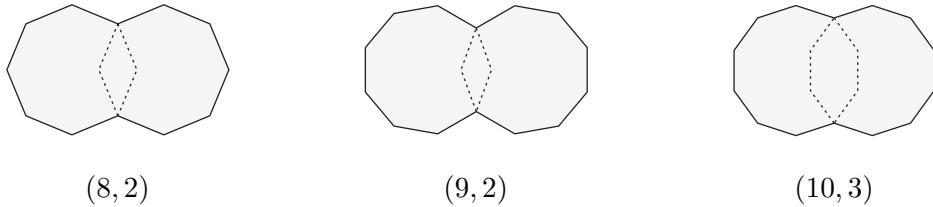
3.10.1   Lattice projection tilings

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.

$(8, 2)$ $(9, 2)$ $(10, 3)$
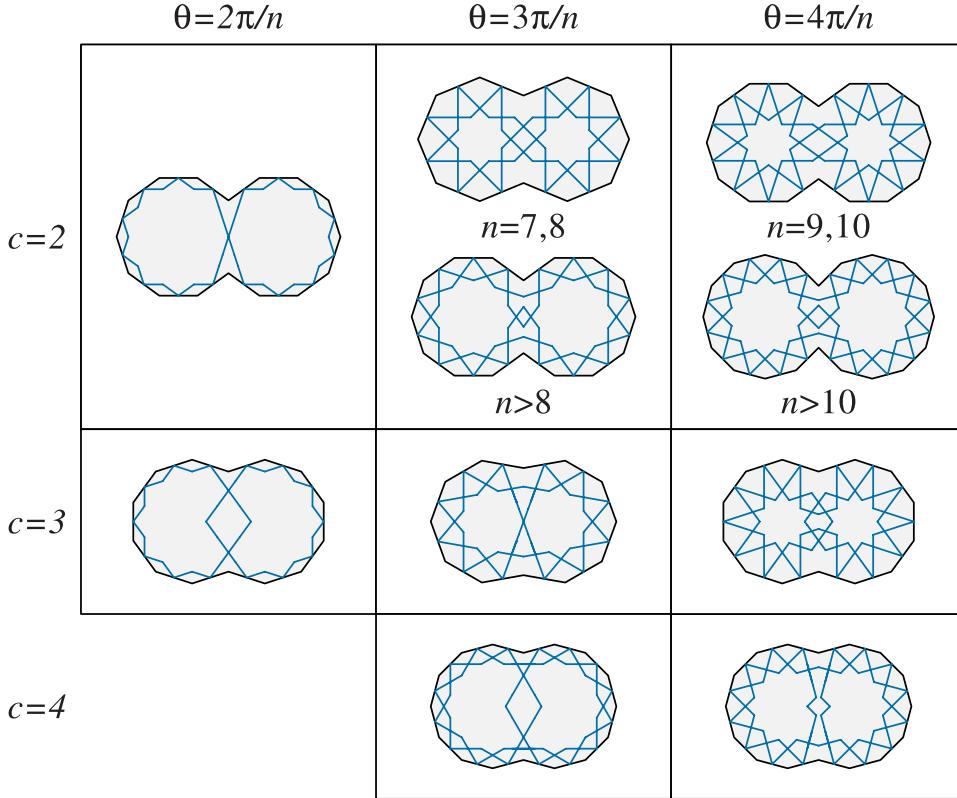
Figure 3.39 Examples of (n, c)-monsters for $n = 8$, $n = 9$, and $n = 10$.

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.
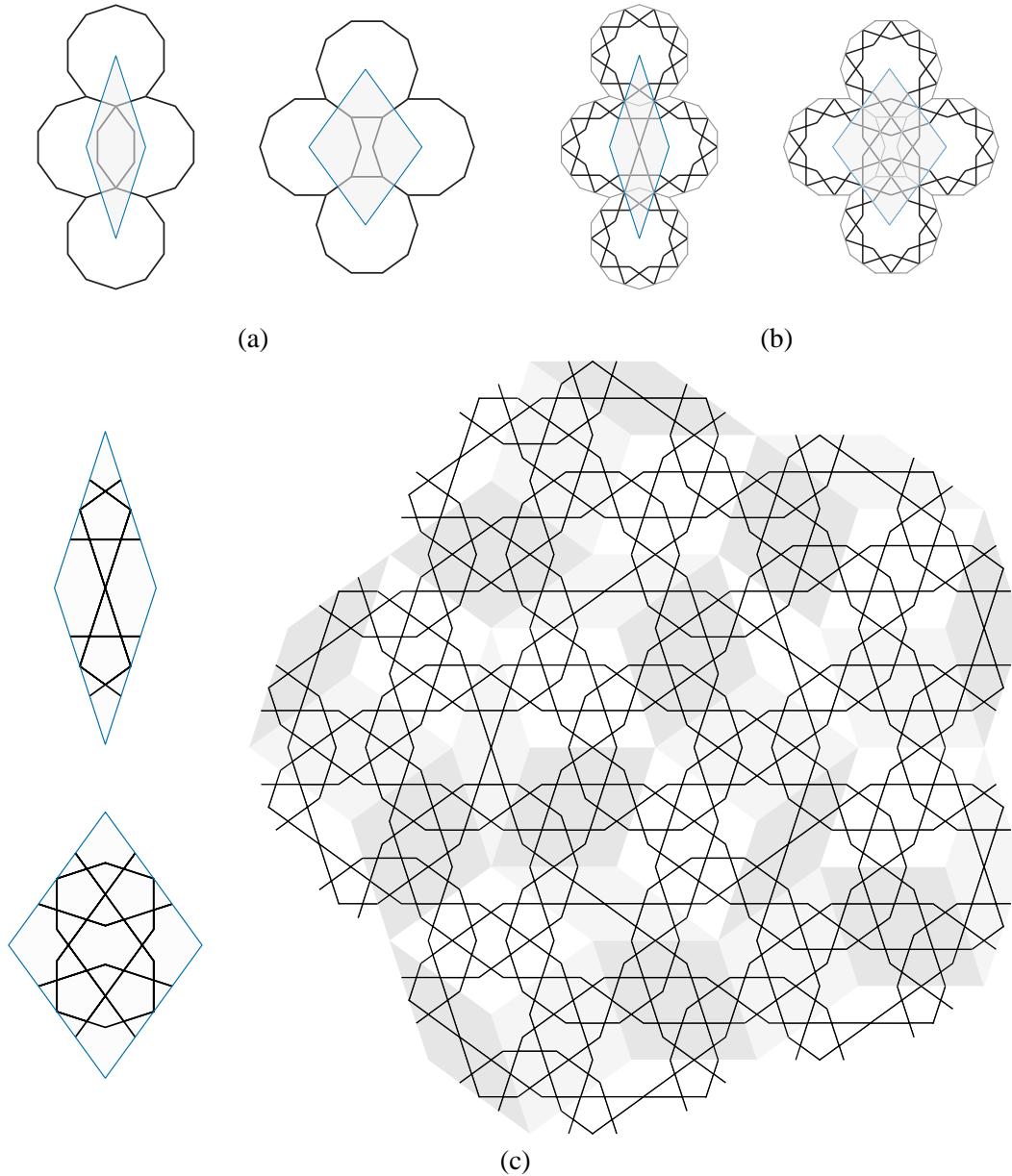
Monsters are somewhat special, in that the motif chosen to deal with the region where the polygons overlap is usually specially tailored to that situation. In some cases, a solution will generalize to other values of n, c, and the desired contact angle. In his manuscript, Bonner provides motifs for the $(10, 2)$-monster under various choices of contact angle. For the contact angle $2\pi/5$, Castera
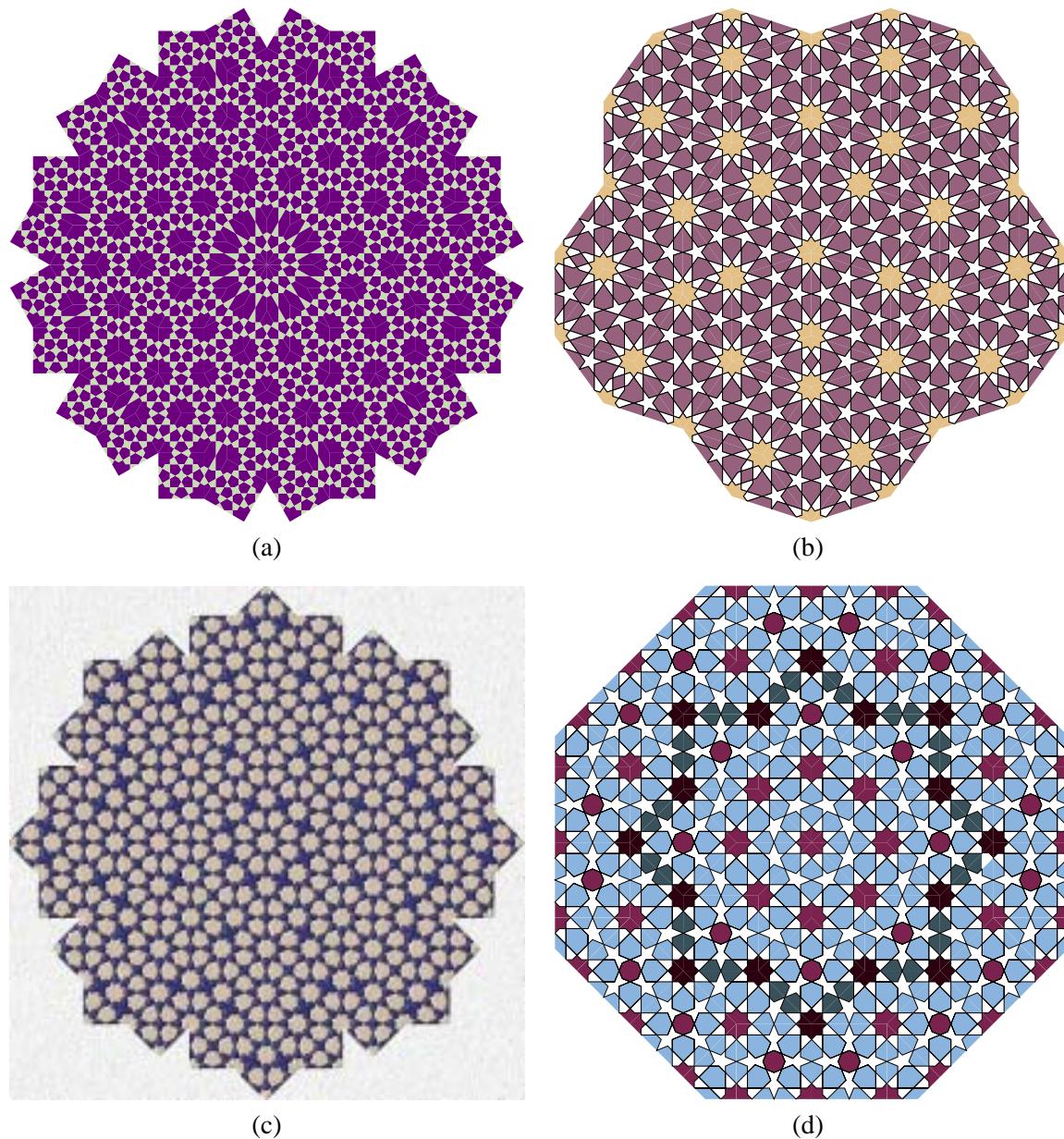
Figure 3.40 A menagerie of monsters and their motifs. This table shows a collection of (n, c)-monster motifs useful for $n \leq 12$. Each row depicts monsters with a different value of c. The columns depict varying contact angles (I consider only multiples of π/n). Except where otherwise noted, the entry 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

(a)　　　　　　　　　　　　　　(b)

(c)

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.
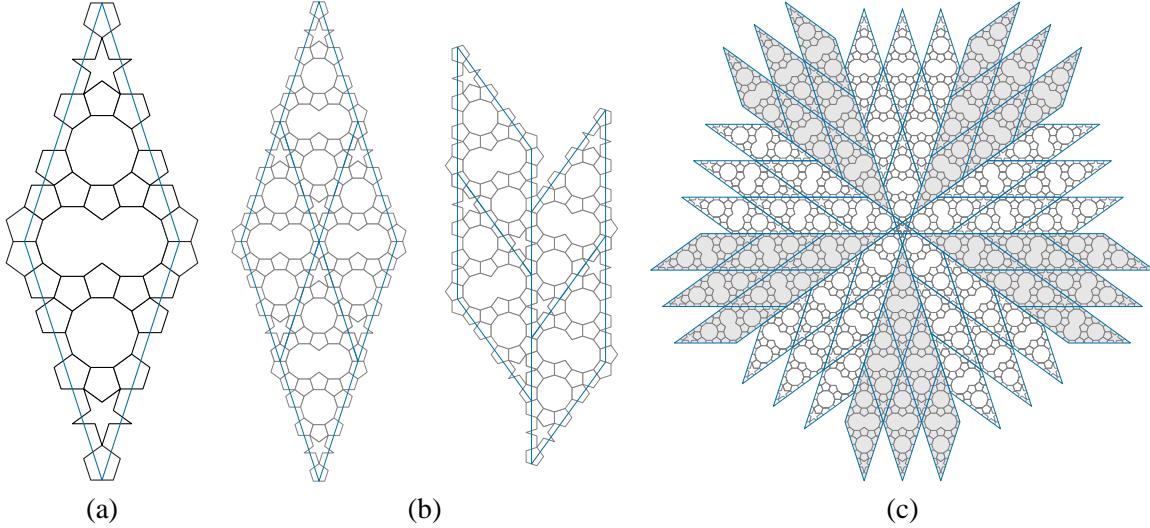
3.10.2   Kepler's Aa tiling

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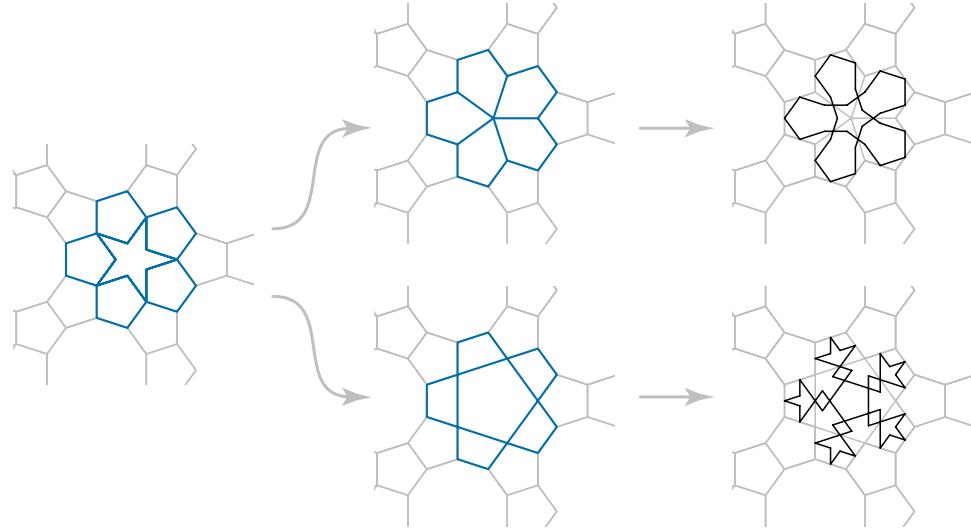
Kepler's tiling is made up of regular pentagons, regular decagons, $(10, 2)$-monsters, and pentacles (stars of the form $\{5/2\}1$). Except for the pentacles, all of these tiles are now familiar ground, and motifs can be given for them easily. The one remaining obstacle in building a star pattern based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

(a)　　　　　　　　　　(b)　　　　　　　　　　(c)

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.
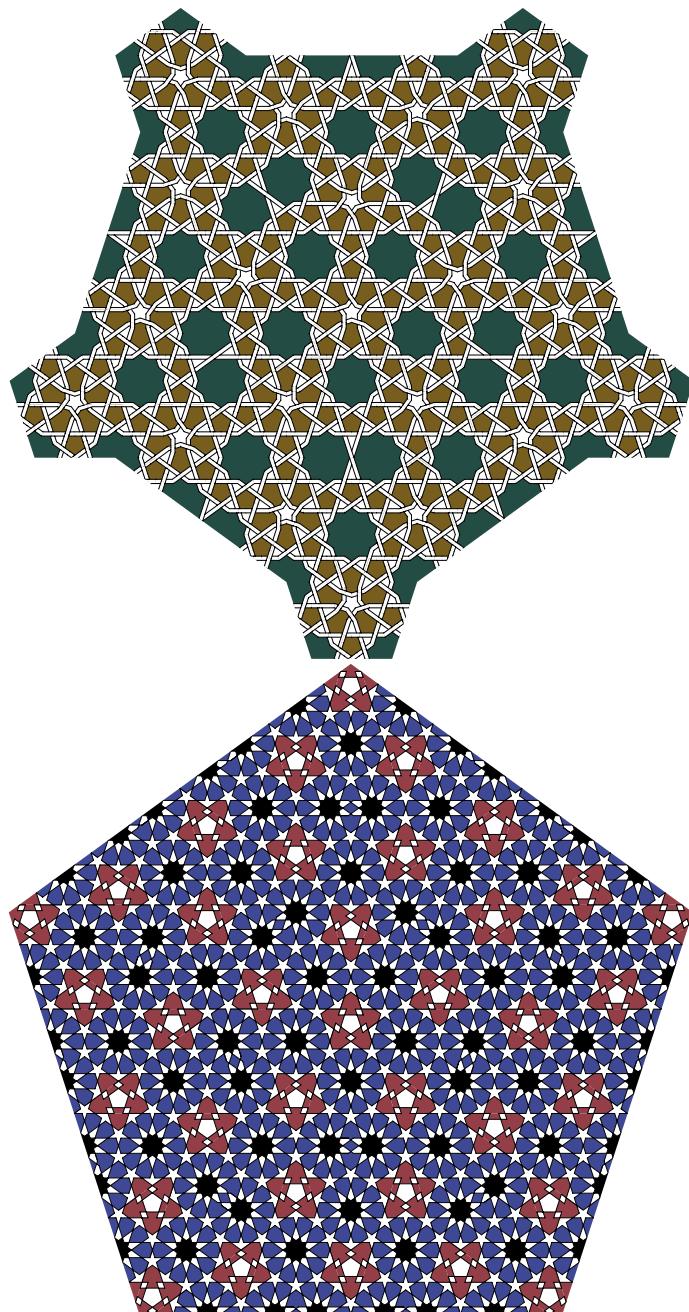
3.11   Future work

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.

Chapter 4

ESCHER'S TILINGS

4.1 Introduction

M. C. Escher, more than any other individual through-out history, brought forth the great beauty that lies at at the intersection of art and mathematics. The im-mense popularity of his graphic art shows no signs of waning, and his imagery continues to appear regularly everywhere people casually place eye-catching art. Al-though he downplayed his abilities as artist and as math-ematician, his work has always been a source of pro-found inspiration to both. It drives the artist to seek out the aesthetic possibilities of geometry and pushes the mathematician to invent new machinery to explain the designs he created through what he considered non-technical means.

Figure 4.1 M.C. Escher in a self-portrait. *M.C. Escher's "Self-portrait"* ©*2000 Cordon Art B.V. – Baarn – Holland. All rights reserved.*

Bruno Ernst provides a classification of the many ways that mathematics played a role in Escher's work. Any one of these suggests a fruitful avenue for exploration of the geometric aesthetic. I have chosen to focus on the regular division of the plane, on the many tesselations he created.

Escher had a lifelong obsession with the regular division of the plane, an obsession that in part can trace its lineage back through the Islamic star patterns of the previous chapter. Escher made several visits to the Alhambra and other monuments in Spain. He was deeply inspired by the interlocking geometric forms of the Moors but felt it a pity that they were forbidden by their

religion from depicting real-world objects in their art [50].[1] He undertook as a personal quest the reinvention of geometric art, substituting easily-recognized motifs such as animal forms for the purity of the Moorish rosettes and polygons.

By the time Escher began studying the regular division of the plane in the first half of the twentieth century, tiling as an art form had passed mostly into history, to be replaced by the growing development of a systematic mathematical theory. Escher came along at the right time. He was born just as physicists and mathematicians were beginning to categorize the sorts of symmetries he used. One of his first exposures to the mathematics of symmetry was when his half brother, a geologist, pointed him at a paper by Polyà.

Escher arrived at each of his interlocking animal forms after a great deal of tinkering and manipulation. Over the years, he became more proficient at inventing new arrangements of motifs, developing his own "layman's theory" of tilings to track the ground he had covered and suggest new directions for exploration. He managed over his career to produce a notebook with more than a hundred of these ingenious, playful designs. His notebooks are lovingly reproduced by Schattschneider [124].

Most of Escher's work is not ornamental in the traditional sense. The regular division drawings in his notebook were either intended purely as exercises, or as studies for future prints. Later in his life, when Escher attained a certain measure of celebrity, he did receive several commissions for ornamental installations around Holland, and he covered architectural surfaces with patterns in paint, inlaid wood, and ceramic tile (see the notes in Schattschneider [124] for photographs of some of these installations). Often he took a periodic pattern and wrapped one of the directions of periodicity into a circle, resulting in a cylindrical pattern that could be used to cover a column. Schattschneider also points out that even if Escher did not focus his efforts on ornamental design, his work has had a great influence in the decorative arts [124, Page 281].

In this chapter, I develop a suite of mathematical and software tools that enable the construction and manipulation of Escher-style tilings of the plane. The mathematical tools are based on the theory of isohedral tilings, as developed by Grünbaum and Shephard and presented in Section 2.4.

---

[1]As Section 3.1 points out, the prevalence of geometric patterns in Islamic art is not fully explained by a simple religious prohibition. However, this misunderstanding was common in Europe at the time that Escher was exposed to Islamic art.

The main focus of the chapter is *Escherization*: an algorithm that, given an arbitrary polygonal shape, attempts to find a tiling of the plane using a prototile that resembles the shape. Escherization takes over the task of searching through the complex space of tilings, providing an outline that an artist can manipulate and decorate. I then adapt Escherization to handle more complex cases, namely where there are two goal shapes and the objective is to create a dihedral tiling of the plane. I also consider the problem of creating Escher-like tilings using Penrose's aperiodic tile sets, tiles that Escher might very well have enjoyed working with had they been discovered during his lifetime.

4.2   Related work

Software specifically geared towards the construction of tilings of the plane has been around for at least twenty years. Chow had a very successful FORTRAN program [21] that let the user input the portion of a tile's boundary that is "independent," *i.e.*, not constrained to some other part of the boundary through a symmetry of the tiling. The program then filled in the remaining part of the tile and replicated it in the plane.

For many years, Kevin Lee has offered a commercial software package called TesselMania! that makes it easy to draw and decorate Escher-like tilings. His system is geared towards the use of tilings as a tool for mathematics education, and the most recent version of TesselMania! includes tutorials, games and puzzles designed for teaching concepts of geometry.

Tupper's Tess [87] has traditionally allowed the user to create drawings belonging to the frieze and wallpaper groups, in a manner similar to the Geometry Center's Kali [5]. Recently, he modified Tess to support a set of tilings directly. Like TesselMania!, Tess is geared towards pedagogical use.

The three programs above are all based on the 28 *Heesch types* [80], which are in some sense precursors to the isohedral tilings that do not take into account the additional internal symmetries that a prototile may be forced to have. Roughly speaking, each Heesch tiling represents a set of isohedral types related through a hierarchy of increasingly symmetric tile shapes, and so the Heesch tilings are just a coarse-grained view of the isohedral tilings. Schattschneider reproduces a table of the 28 Heesch types in her book [124, Page 326]. In principle, much of the work in this chapter could have been based on the Heesch types rather than the isohedral types. However, the availability, thoroughness, and clarity of Grünbaum and Shephard's presentation of the isohedral tilings make

them the preferable choice for this research.
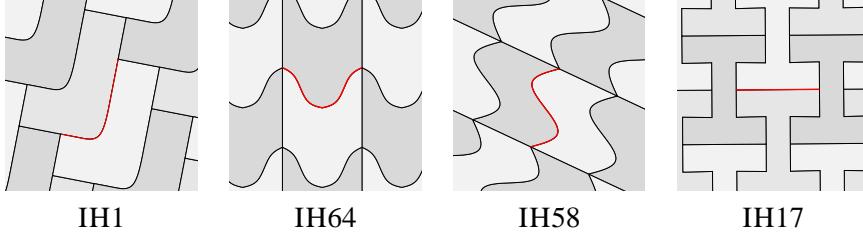
Escher created a small number of carved wooden sculptures featuring spherical interpretations of his tesselations. Using these as a starting point, Yen and Séquin [140] created an "Escher Sphere Construction Kit," a system that allows the user to design ornamental spherical tilings much as one could create Euclidean tilings in the systems above. As an added feature, the tilings they create can be exported to rapid prototyping hardware and constructed as real artifacts, as was done with spherical Islamic star patterns in Section 3.9. They have created many attractive spherical patterns. Some are fabricated in one piece, others tile by tile, assembled by gluing each tile to the surface of a ball.

4.3 *Parameterizing the isohedral tilings*

We now return to the subject of the isohedral tilings first described in Section 2.4.1. The isohedral tilings turn out to have a balance of mathematical, algorithmic, and aesthetic properties that make them an excellent choice in the search to create new designs in the spirit of Escher. We use them as the basis for almost all of the work in this chapter.

Escher rarely strayed from the isohedral tilings when developing his tesselations. Mathematically, every monohedral tiling produced by his "layman's theory" is isohedral. Only one of his roughly 140 notebook drawings is anisohedral, and that drawing was based on a tiling that Penrose showed to Escher specifically to demonstrate the existence of an anisohedral prototile [118]. Even his periodic tesselations with multiple motifs are ultimately based on the isohedral tilings, for he constructed such tilings by splitting an isohedral prototile into multiple pieces. As a further justification for our decision to use IH, we may appeal to Grünbaum's argument that in building a pattern, artists and designers focus on the relationship between a motif and its neighbours, rather than that between a motif and the whole [64]. The isohedral tilings are precisely those monohedral tilings in which a tile's relationship to its neighbours completely determines the complete structure of the tiling.

On the practical side, the combinatorial structure of the isohedral tilings allows us to construct simple data structures to represent them and efficient algorithms to manipulate them. Building upon the mathematical treatment of the isohedral tilings given in Section 2.4.1, this section and the next

Figure 4.2 Examples (from left to right) of **J**, **U**, **S** and **I** edges. In each case, the tiling edge with the given shape is highlighted in red.

one develop the computational tools needed to work with IH, the space of isohedral tilings. I begin here by providing a parameterization of the space of shapes of isohedral prototiles. In the next section, I will use that parameterization to construct a software library.

Within the tilings of a single isohedral type, prototiles are distinguished from each other by their shapes, determined by the positions of the tiling vertices and the shapes of the curves that join them. In order to move from the combinatorial description of isohedral tilings to a geometric one, we must understand how incidence symbols dictate the range of possible prototile shapes for a given isohedral type. We parameterize the space of isohedral tilings by giving, for each type, an *edge shape parameterization* and a *tiling vertex parameterization*. The former encodes the minimal non-redundant geometric information sufficient to reconstruct the tiling edges. The latter determines the legal configurations of tiling vertices.

4.3.1 Edge shape parameterization

The constraints on the shapes of tiling edges in an isohedral tiling are simple to describe. Although the underlying choice of how to represent a curve is left open, the tiling's symmetries imply a great reduction in the tiling edges' degrees of freedom. These constraints can be extracted directly from the tiling's incidence symbol. We enumerate the four cases for the structure of a tiling edge. For each case, Figure 4.2 shows a tiling with such an edge.

If some directed edge is adjacent to itself without a flip, then a tile's neighbour across that edge is adjacent through a half-turn. This rotation forces the edge shape to itself be symmetric through a half-turn about its centre. We call such an edge an **S** edge as a visual mnemonic. Only half of an **S**

edge is free; the other half must complete the rotational symmetry.

An undirected edge must look the same starting from either end, meaning it must have a line of mirror symmetry through its midpoint. If the edge is adjacent to any edge other than itself, it is free to take on any curve with this bilateral symmetry. We call it a **U** edge. Again, only half of a **U** edge is free.

If an undirected edge is adjacent to itself, or if a directed edge is adjacent to itself with a change in sign, that edge must have both **S** symmetry and **U** symmetry. The only shape that has both is a straight line, leading us to call such an edge an **I** edge.

The remaining case is when a directed edge is adjacent to some other directed edge. Such an edge is free to take on any shape, and we call it a **J** edge.
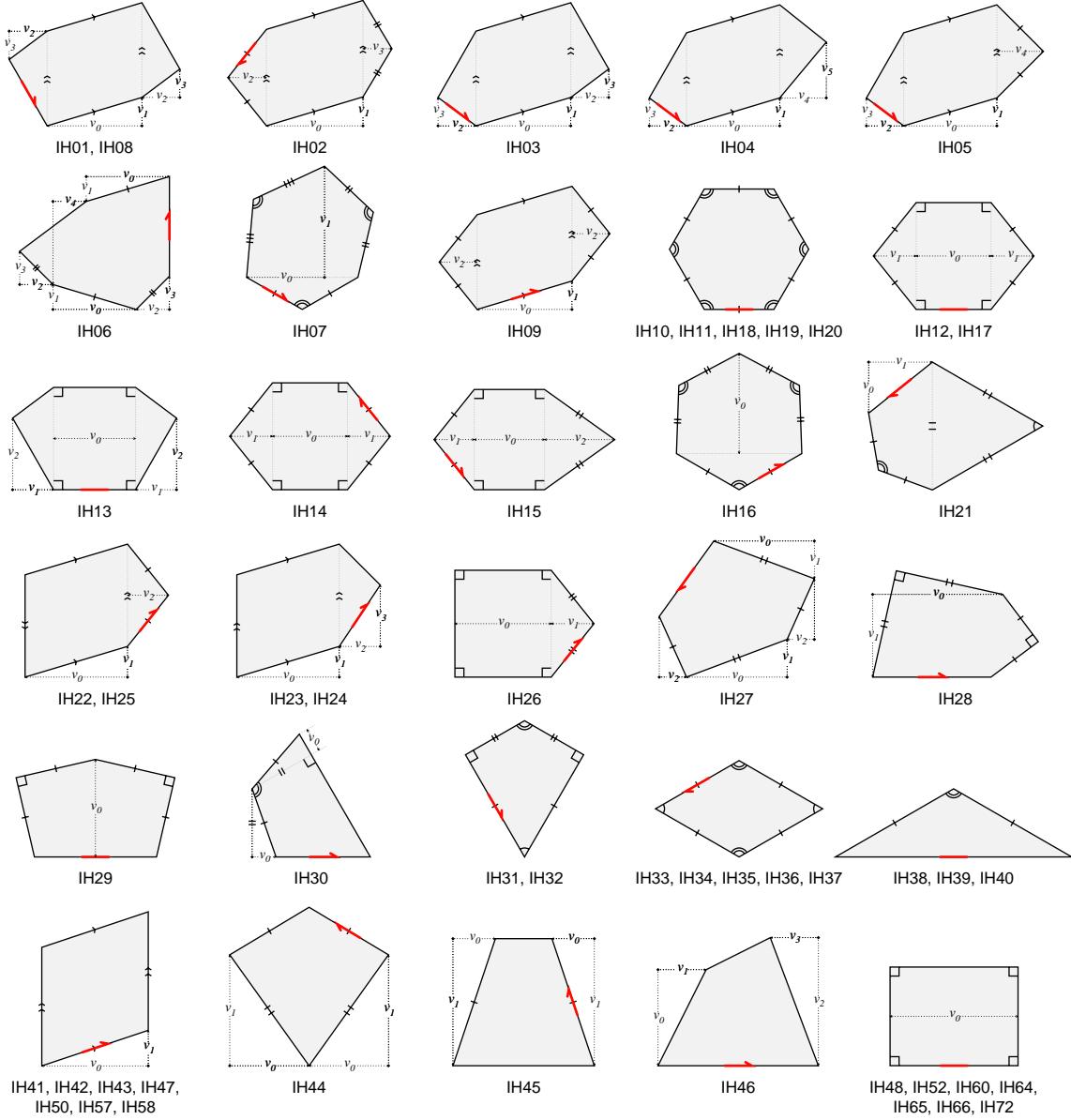
Note also that if an edge x is adjacent to an edge y, then x and y have the same shape (even though they have different names). In this case, we need only represent one tiling edge, since the other is entirely constrained to it. Thus, referring back to the derivation presented in Figure 2.16, the tiling edges of IH16 can be summarized by one curve: the shape of the edge labeled b. Edges labeled a are **I** edges and have no degrees of freedom, and edges labeled c are constrained to b.

4.3.2   Tiling vertex parameterization
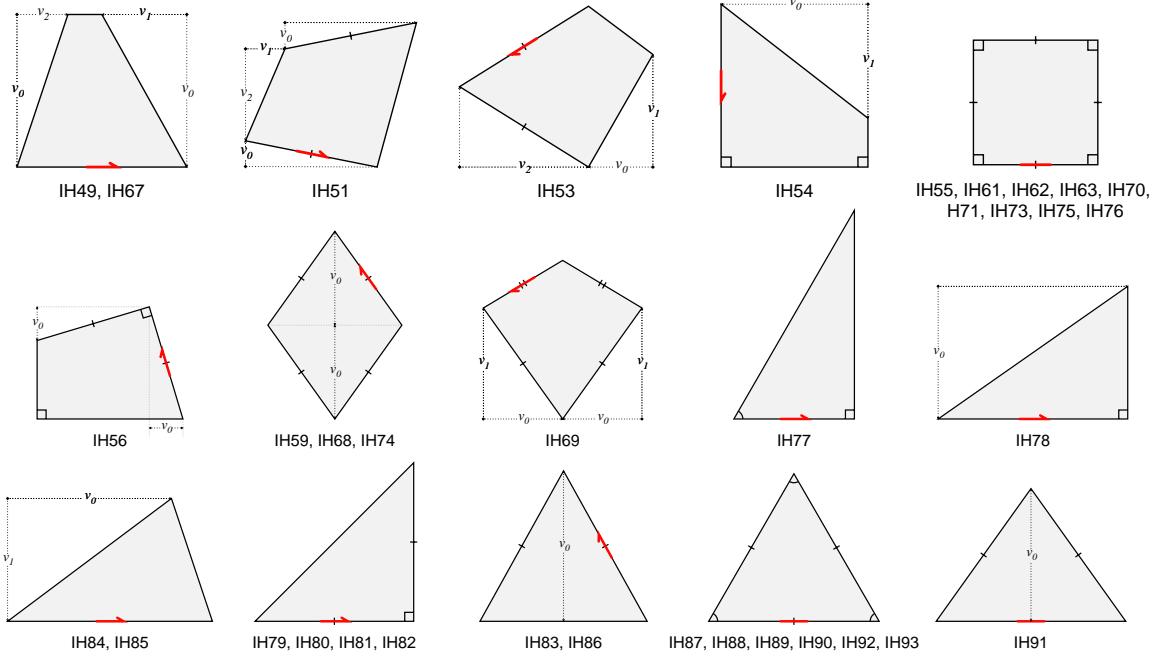
Like the shape vertices, tiling vertices cannot move entirely independently of each other. Moving one tiling vertex forces the others to move to preserve the tiling. The exact nature of this movement depends on the tiling type in question. The incidence symbol for a tiling type implies a set of constraints on the tiling polygon's edge lengths and interior angles. Any tile of that type will have a tiling polygon that obeys those constraints.

If we hope to build a generative model of isohedral tilings, it is not sufficient merely to recognize the constraints on the shape vertices: we need a way to explicitly navigate the space of legal tiling polygons. For each isohedral type we need a parameterization of the tiling vertices for tilings of that type. The parameterization should be *complete*, in the sense that for every legal configuration of tiling vertices, there is a set of parameters that generates that configuration. We also require it to be *consistent*, in the sense that every set of parameters generates legal tiling vertices.

We have developed a set of parameterizations for the isohedral types that we believe to be
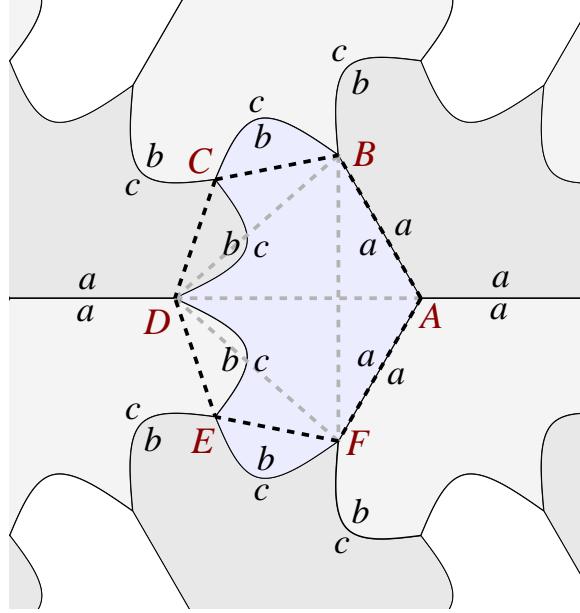
Figure 4.3 The complete set of tiling vertex parameterizations for the isohedral tilings. In each tile, the edge marked with a red line is the first edge in the tiling type's incidence symbol. When that first edge is directed, the red line has an arrowhead. Labelled dotted lines represent parameter values, and are horizontal or vertical (with the exception of one guide line in the diagram for IH30). Since the diagrams are scale independent, distances that do not depend on parameters can be taken to have unit length. Tile edges cut with the same number of short lines have the same length, and edges cut with chevrons are additionally parallel. A single arc, a small square, and a double arc at vertices represent 60°, 90°, and 120° angles, respectively.

Figure 4.3 The complete set of tiling vertex parameterizations for the isohedral tilings (continued).

consistent and complete, though the history of IH has certainly experienced its share of flawed analyses [68, Section 6.6]. The parameterizations were derived by determining angle and length constraints from the incidence symbols and parameterizing the unconstrained degrees of freedom. In some cases, parameterizations are shared between tiling types: nine tiling types have squares as tiling polygons (implying a parameterization with zero parameters), and seven have parallelograms (implying two parameters). These easy parameterizations are balanced by tiling types with one-of-a-kind structure that can take some thought to derive. In all, the 93 isohedral types require 45 different parameterizations. Diagrams of the parameterizations appear in full in Figure 4.3.

To our knowledge, no tiling vertex parameterizations have ever been given specifically for IH. They represent a nontrivial extension to the table of information about IH found in Grünbaum and Shephard. Previously, Heesch and Kienzle provided tiling vertex parameterizations for the 28 Heesch tiling types [80]. Our parameterizations can be seen as an elaboration of theirs; each Heesch type is identical to one of the isohedral types, and for those types the parameterizations coincide.
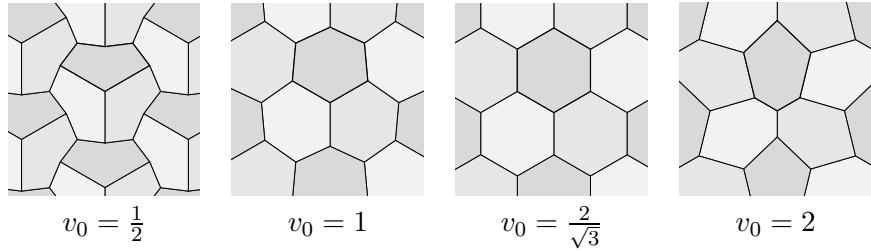
Figure 4.4 The diagram used to establish a tiling vertex parameterization for IH16. For simplicity, the arrows indicating edge direction have been left out of the diagram.

The remaining isohedral types have parameterizations where degrees of freedom are coalesced to yield more symmetric tiling polygons. Because our parameterizations provide information directly tied to each individual isohedral type, they are still useful in applications of tiling theory.

Often, determining the tiling vertex parameterization for an isohedral type is easy. Take, for example, IH41. The table provided by Grünbaum and Shephard tells us that every tiling of this type will have symmetry group $p1$, consisting simply of translations in two directions. Furthermore, each translational unit contains a single aspect. These two facts imply that the tiling polygon must be able to function as a translational unit of the tiling, meaning that it must be a parallelogram.

To give the flavour of a more complicated parameterization, here is a sketch of the derivation for IH16 (see Figure 4.4). We begin by placing at least enough tiles to completely surround one central tile, and marking up the tiles with the labels from the tiling's incidence symbol. Now consider the situation at tiling vertex A. This vertex is surrounded by three copies of the same angle from three different tiles, namely $\angle FAB$, the angle between the a edges. It follows that the tiling polygon must have a $120°$ angle at that vertex. The same observation applies to vertices C and E. Thus,

$$v_0 = \tfrac{1}{2} \qquad v_0 = 1 \qquad v_0 = \tfrac{2}{\sqrt{3}} \qquad v_0 = 2$$

Figure 4.5 Some examples of IH16 with different values for the single parameter in its tiling vertex parameterization.

$\triangle FAB$, $\triangle BCD$, and $\triangle DEF$ are all $120°$ isosceles triangles. Because these isosceles triangles can be constructed given only the edge opposite the $120°$ angle, the tiling polygon depends entirely on the "skeleton" triangle $\triangle BDF$. Furthermore, the incidence symbol reveals a line of bilateral symmetry in the tile across $\overline{AD}$, forcing $\triangle BDF$ to be isosceles. The only degrees of freedom left in the tiling polygon are the lengths of $\overline{AD}$ and $\overline{BF}$. However, we are not interested in capturing the absolute size of the tiling polygon, merely its shape up to similarity. We can factor out the dependence on scale by fixing $BF = 1$ and keeping just a single parameter: $v_0 \equiv AD$. Figure 4.5 shows tilings of type IH16 that can result from different values of this single parameter.

4.4 Data structures and algorithms for IH

Using the parameterization given in the previous Section, I have developed a computer representation of isohedral tilings. The representation is expressed as a self-contained C++ library. I have written a number of large applications and small utilities that use this library to generate tilings, edit them, render them in various ways, and extract information from them.

At the top level, the library provides two classes: `IsohedralTemplate`, an abstraction of an isohedral tiling type, and `IsohedralTile`, an abstraction of a specific prototile. The template contains information about a tiling type in general, information that doesn't change from instance to instance. The prototile refers to a template and contains the information needed to determine the locations, shapes, and colours of tiles. I describe each of these components in detail, and then show how they can be used to support efficient editing and viewing.

```
        template IH16 {
   1        topology 3^6
   2        symbol [a+b+c+c-b-a-;a-c+b+]
   3        colouring (1 2 3) (1 2 3) (1 2 3)
   4        aspects 3
   5        rules
   6            aspect 2 1
   7            aspect 3 6
   8            translate T1 1,5
   9            translate T2 1,3
        }
```
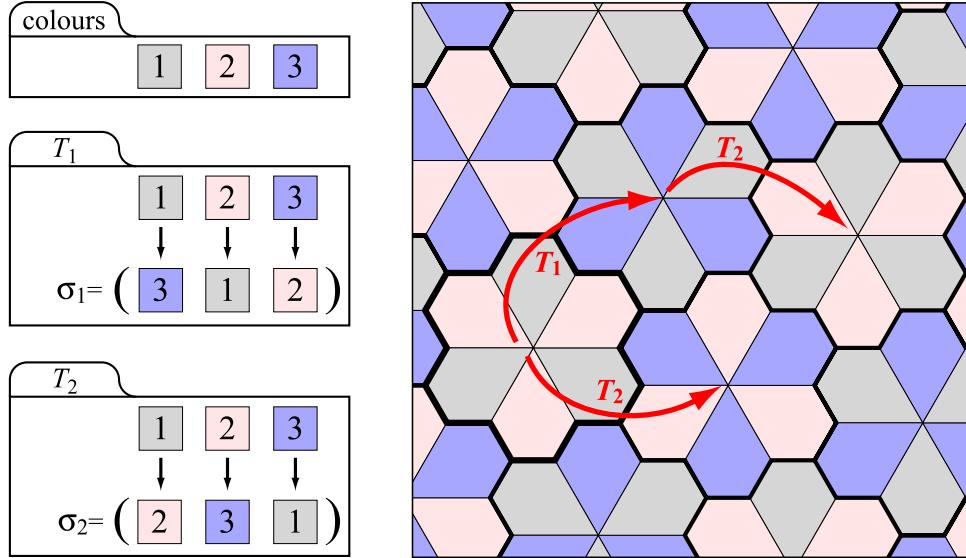
Figure 4.6 The tiling type information stored for IH16

4.4.1 Isohedral templates

The templates are computed once ahead of time, and stored in a master file (`isohedral.ih`) designed to be computer-readable. This file has been publicly available [92] on the World Wide Web for about two years, and has been extensively debugged in that time. When the tile library is initialized, the file is parsed and an `IsohedralTemplate` instance is created to hold each template. Other programmers using `isohedral.ih` have written software to generate code directly from the file, potentially leading to better performance, at the expense of the ease of debugging that was necessary as the templates were first being developed.

The template file contains one record for each isohedral type. A sample of such a record appears in Figure 4.6. It reproduces some of the information tabulated by Grünbaum and Shephard, such as the topological type (line 1), the incidence symbol (line 2), and the number of aspects (line 4). It also gives a default colouring (line 3). The remaining information, the `rules` section (lines 5–9), is symbolic description of how to compute the tiling's translation vectors and transform matrices for its aspects. We execute the symbolic description and cache the resulting transforms in an `IsohedralTile` to permit efficient rendering of tilings. In what follows, I provide more details about the `colouring` field and `rules` section.

The `colouring` field provides a default rule for assigning colours to tiles (colourings of tilings are described in Section 2.5). An `IsohedralTile` may override this default with its own colour-
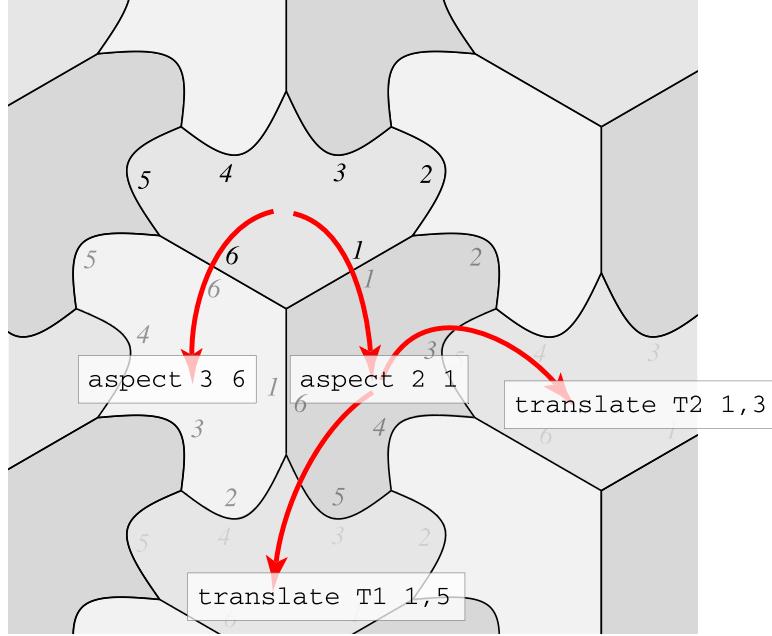
Figure 4.7 A demonstration of how the colouring information in the isohedral template (for IH21 in this case) is used to apply colours to tiles. The translational units (each containing six aspects) are outlined in bold. There are three symbolic colours, $\{1, 2, 3\}$, and they are associated respectively with gray, pink, and blue. On the left, the permutations for the two translation vectors are indicated by showing with arrows the mapping from original to permuted colours; the permutation's textual desciption can be read off of the bottom row of this mapping. On the right, the permutations are applied when moving between translational units. The colouring for this tiling can be read from the diagram as
```
colouring 3 (1 2 1 2 1 2) (3 1 2) (2 3 1).
```

ing. Here we follow Escher's lead and aim to provide perfect colourings. Recall that in a perfect colouring, every symmetry of the tiling is a permutation of the set of colours.

The actions of all the symmetries can be summarized by giving the permutations associated with the two translation vectors of the tiling and a default assignment of colours to the aspects in a single translational unit. Successive translations will permute this default assignment appropriately. The colouring field in the template gives, in order, the number of colours, the assignment of colours to aspects in a translational unit, and the permutations of the assignment associated with the two translation vectors. A permutation ρ of the numbers $\{1, \ldots, n\}$ is very simply represented as a sequence $(s_1 \ldots s_n)$, with $\rho(k) = s_k$.

In particular, consider a tiling with translation vectors $\overrightarrow{T_1}$ and $\overrightarrow{T_2}$ and their associated colour

Figure 4.8 A visualization of how aspect transforms and translation vectors are computed for IH16, using the information in the `rules` section of the isohedral template (see Figure 4.6). In the order that they are referenced in the template, the aspects are coloured blue, pink, and gray. The edges are numbered as they are given in the incidence symbol. Each red arrow represents a single hop, a rigid motion that brings a tile into coincidence with one of its neighbours. The end of every sequence of hops is labeled with the corresponding rule from the template for IH16 (see Figure 4.6).

permutations ρ_1 and ρ_2. Let the tiling have n aspects, with the default colours in a translational unit given as $(c_1, \ldots, c_n)$. Then aspect k in the translational unit located at $a\overrightarrow{T_1} + b\overrightarrow{T_2}$ will have the colour $\rho_2^b(\rho_1^a(c_k))$. This encoding can in fact express a superset of the perfect colourings, but it is easy to check empirically whether a given colouring is perfect.

The `rules` section gives a collection of rules that, when applied to a tiling polygon, yield rigid motions (in the form of transformation matrices) for all the aspects of a translational unit, as well as for the two translation vectors. These transforms cannot be computed ahead of time, as they depend on the tiling polygon. We speed up the drawing of the tiling by storing these transform matrices in the `IsohedralTile` instance, and recomputing them only when the tiling vertices move.

Every tile in an isohedral tiling is surrounded in a consistent way by its neighbours, and so for

every tiling edge there is a well-defined rigid motion that carries the tile on one side of that edge to the tile on the other side. The motion will either be a half-turn around the edge's center (in the case of an **S** edge), a reflection across the edge (in the case of an **I** edge), or a translation (in the cases of **J** and **U** edges). The kind of transform that applies can be determined from the tiling type's incidence symbol, and the numeric values in the transform matrix depend on the positions of the tiling vertices that delimit the edge. We call such a rigid motion a "hop" across a tile edge. In a tile with n edges, we can label the hops unambiguously as $H_1, \ldots, H_n$. Each `rule` encodes a sequence of hops that, when chained together, transform a tile to a new aspect or to the same aspect in a neighbouring translational unit.

Aspect 1 is always given the identity matrix as its transform, and the other aspect transforms are computed from it. In the example, the first rule (line 6) says that the transform for creating aspect 2 from the first aspect is the hop across edge 1 of the first aspect — that is, a reflection across the first edge, labelled `a+`, in the incidence symbol. We will store H_1 as the aspect's transform matrix. Similarly, the second rule (line 7) says that the transform for creating aspect 3 from the first aspect is H_6, a reflection about the edge labelled `a-`.

The two translation vectors are specified in the same way. Here, we can obtain translation vector $\overrightarrow{T_1}$ in two hops, first from the first aspect across edge 1 into some neighbouring tile, and from *that* tile across edge 5. The resulting transform matrix would be $H_1 H_5$. Note, however, that this matrix does not necessarily represent a translation, and so we cannot just take $\overrightarrow{T_1}$ to be the translational component of that matrix. The problem is that the matrix may contain internal symmetries of the tile shape, which were accumulated when composing the hops together. Fortunately, we can still extract the translation in a simple way as the vector joining the centroids of the transformed and untransformed tiling vertices. This calculation works because the centroid is independent of internal tile symmetries, operations that merely permute the vertices.

In general, a `rule` may specify any number of hops to get from the first aspect to another aspect or a translation. Each step in the `rule` names an edge of the tile, and the transform is computed by composing together the associated hops.

One piece of per-tiling-type information missing from the template file is the set of tiling vertex parameterizations. The parameterizations are more easily described in code than in a table-driven format, and are embedded in the source code, each as a C++ class. A Python file `params.py` that

```
def ih16_params( v0 ):
    m = 0.5 / math.sqrt( 3.0 )
    T1 = match( Point( 0.5, v0 ), Point( 1, 0 ) )
    T2 = match( Point( 0, 0 ), Point( 0.5, v0 ) )

    return (
        Point( 0.5, -m ),
        Point( 1, 0 ),
        T1 * Point( 0.5, m ),
        Point( 0.5, v0 ),
        T2 * Point( 0.5, m ),
        Point( 0, 0 ) )
```
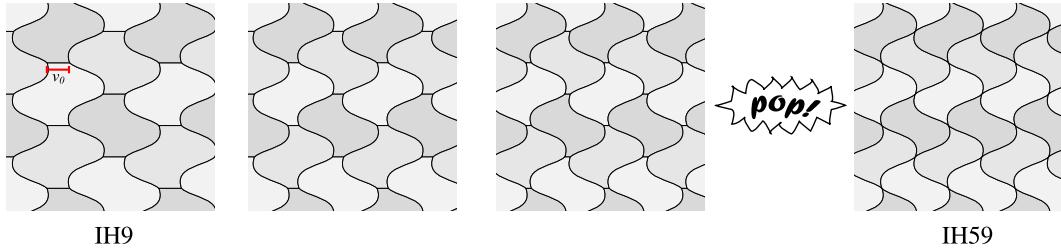
Figure 4.9 Sample Python code implementing the tiling vertex parameterization for IH16. When called with a single real parameter `v0`, the function returns a tiling polygon. The function `match` takes two points as arguments and returns a direct rigid motion that maps the unit interval onto the line segment given by the two points.

implements the parameterizations is available on the World Wide Web along with `isohedral.ih`. An example from `params.py` is given in Figure 4.9.

This representation of isohedral tilings suffers from a flaw related to degenerate edges in the tiling polygon. If two consecutive tiling vertices are made to coincide, then the hop across their shared edge is undefined, and any `rules` that use the degenerate edge give invalid transforms. In a purely mathematical treatment of the subject there is no problem, because there is no such thing as a degenerate edge in the tiling polygon. As two adjacent tiling vertices merge, they fuse into a single vertex and the tiling as a whole slips into a different (but related) isohedral type, as shown in Figure 4.10. The representation given here can manipulate non-degenerate tiles without any difficulty, but it cannot handle these discontinuous transitions. We will return to the subject of these discontinuities in Section 5.4. An interesting extension to the library would be to classify these isohedral types that are related through degeneracy, and transparently slip from type to type as tiling vertices move through each other.

Note also that in the tiling vertex parameterizations given, some adjacent tiling vertices simply cannot be brought into coincidence. These correspond to the edges of the tiling polygon that are assumed to have unit length. Any enhancement allowing degenerate edges would have to provide

Figure 4.10 An example of how a degenerate tile edge leads to a related tiling of a different isohedral type. As parameter v_0 goes to zero in this IH9 tiling, the tiles deform continuously. But at the instant that v_0 becomes zero, pairs of tiling vertices fuse and the tiling passes through a topological discontinuity to type IH59.

alternate parameterizations in which any edge of the tiling polygon not otherwise constrained could become degenerate.

4.4.2   Isohedral prototiles

All the information related to a specific isohedral prototile is stored in the `IsohedralTile` class. A great deal of data is stored in every `IsohedralTile`:

- **Geometry information** includes the parameters for the tiling vertex parameterization, a cached tiling polygon, and the cached aspect transforms and translation vectors derived from the `rules` section of the `IsohedralTemplate`.

- **Shape information** contains polygonal paths that make up the non-redundant portion of the tile's outline (called the "fundamental edge shapes"). The shape information also includes a cached copy of the tile's outline for fast drawing.

- **Colouring information** contains a colouring (like the one that appears in `isohedral.ih`) and actual RGB triples for each symbolic colour.

A callback mechanism ensures that when part of the tile's description changes (for example, when a vertex parameter is adjusted), all cached information that depends on it is automatically updated.

The `IsohedralTile` class is designed to be extensible, flexibly storing additional information supplied by client code. One example of such an extension is a prototype system for associating vector-based artwork with the tile. The artwork is made up of a set of "markings," shapes that have familiar properties such as line width, line colour, and fill colour.

An instance of `IsohedralTile` can be written out to a "tile file," an XML document that contains the non-cached information in the tile object. Each extension is also given the opportunity to serialize itself into the tile file.

Each fundamental edge shape is an array of points representing a path starting at $(0,0)$ and ending at $(1,0)$. By default, the points are interpreted as a sequence of line segments, but to increase the aesthetic appeal of our tilings we have implemented the ability to treat them as control points for a subdivision curve. As a further enhancement, each control point has an associated weight. The higher the weight, the more subdivision steps will go by before that point is averaged with its neighbours. In effect, the weight controls the sharpness of the curve near the control point, with maximum weight yielding a sharp corner that interpolates the control point. (This approach is a curve-based analogue of the "hybrid subdivision" technique of DeRose *et al.* [35].) The subdivision weights are not built in to `IsohedralTile`, but implemented using the extension mechanism described above.

The shape information in the prototile contains a hierarchical model of rigid motions whose leaves are the fundamental edge shapes. The model makes multiple references to fundamental edges to express the redundancy inherent in the tile's outline. To rebuild the tile shape, we apply the tiling vertex parameterization to obtain the positions of the tiling vertices and use the hierarchical model to construct edge shapes between them.

There are at most three levels in the hierarchical model between a fundamental edge shape and a point on the outline of the tile. The first level takes into account the symmetries of **U** and **S** edges. Half of the **U** or **S** edge comes directly from the fundamental edge. The other half is derived from the first half as needed through rotation or reflection. **J** edges are passed unmodified through this level, and since **I** edges are immutable, all tiles share a single system-wide copy of an **I** edge.

At the next level up, we recognize that edges with different names in the incidence symbol may still have related shapes. In IH16, for example, the edge named b+ is adjacent to c+, forcing the two edge shapes to be congruent. In this case, the two edges share the same shape passed up from

the level below.
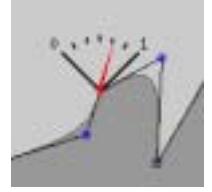
Finally, the topmost level maps the unit interval to an edge of the tiling polygon; this mapping will move an edge shape from its normalized coordinate system into a portion of the tile's outline. At this level, all edges with the same label in the incidence symbol share a lower-level shape object.

4.4.3 Interactive viewing and editing
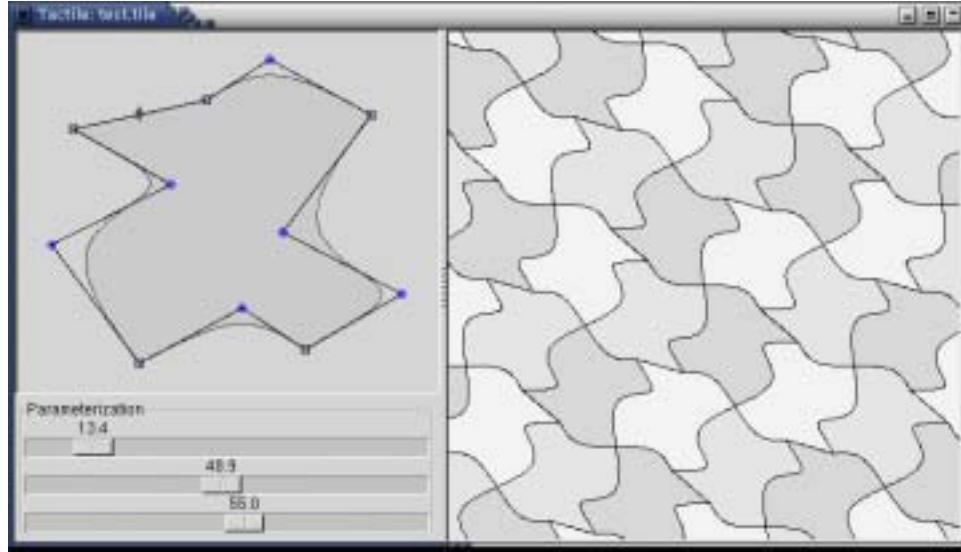
Using the library described above, we have implemented a number of programs that read, edit, and write isohedral tilings. For simple editing of tilings, the most important of these programs is **Tactile**, an interactive editor and viewer for tile files. Tactile is highly responsive, running at interactive rates on an off-the-shelf Linux system with no graphics acceleration. A screenshot of **Tactile** is given in Figure 4.11. Because of the deep sharing of information in the tile representation, when a part of the tile is edited, the system provides immediate feedback by showing all parts of the tile (and tiling) that are affected by the change.

When subdivision is enabled, we provide a novel gauge-based interface for editing weights on control points. The gauge pops up at the vertex location and is set with a radial motion. Setting weights integrates very comfortably with the general process of editing the vertices.
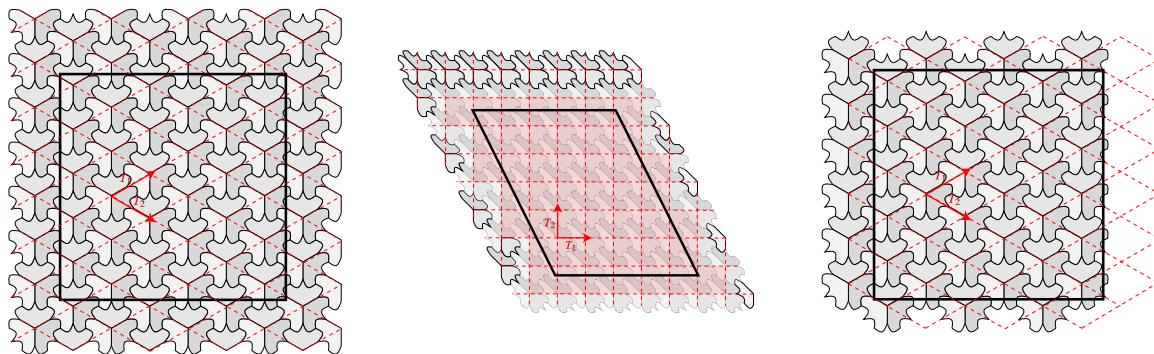
The viewer portion of the interactive system relies on an algorithm for filling a region of the plane with copies of the prototile. Given a tile shape in its local coordinate system and a viewing region, we need to find the set of rigid motions that replicate the tile to cover the region.

To find these motions, we compute the coordinates of the viewing region's corners in the basis formed by the tiling's translation vectors. In that coordinate system, the translational units become lattice squares; we draw the translational units corresponding to the lattice squares that intersect the projection of the viewing region. For each needed translation, we iterate over the tiling's aspects, placing a tile relative to the rigid motion formed by composing the translation with the aspect's transform. This algorithm is demonstrated in Figure 4.12. The figure shows that the algorithm is only approximate — it can leave a border of the viewing region unfilled. However, it is adequate for the purposes of the work presented here. When it is vital that the viewing region be completely

Figure 4.11 A screen shot from Tactile, the interactive viewer and editor for isohedral tilings.

Figure 4.12 The replication algorithm for periodic Euclidean tilings. The left image shows the tiling to be replicated, with a superimposed black square representing the desired viewing region. The red parallelograms delineate translational units of the tiling, based on vectors T_1 and T_2. In the middle image, the whole diagram is shown in a coordinate system where T_1 and T_2 are an orthonormal basis. In this coordinate system, translational units are lattice squares, and it is easy to choose the squares that overlap the viewing region. The chosen units are drawn in the untransformed image on the right. Note that the algorithm is imperfect (it leaves part of the viewing region unfilled) but it is suitable for the purposes of interactive design.

filled with tiles, a simple solution is to inflate the region into a larger rectangle, and apply the filling algorithm on it instead. The filling rectangle can be made large enough that any unfilled part of its border will still lie outside the viewing region.

As was mentioned above, there is a prototype implementation of vector-based artwork for tiles, but it is rarely used because of the engineering work that would be needed to create an interface for designing and editing the artwork effectively. Doing so would be tantamount to re-implementing great drawing software like Adobe Illustrator. One future solution would be to implement a marking system as an Illustrator plug-in that takes advantage of their easy-to-use interface and exports artwork compatible with tile files.

In the meantime, I make much more frequent use of image-based tile artwork. I have implemented an image-based tiling renderer based on **libart**, a free image manipulation library that provides a sophisticated imaging model [100]. The renderer takes a tile file and a set of images (one for each colour in the tiling's colouring) to serve as backdrops. For each tile in a region, it starts with the image backdrop for that tile's colour, optionally applies a semi-transparent wash of the tile colour, rasterizes any markings that are present, draws the tile's outline, and transforms the composited tile into its position in the final rendering. Note that in this model, the user can still incorporate vector-based artwork from an external tool by rasterizing the artwork and feeding the resulting image to the renderer. This approach is not ideal, but it produces satisfactory renderings in most cases (see, for example, Figures 4.15(d) and 4.26(b)).

4.5 Escherization

Our goal in this chapter is to understand Escher's tesselations with the ultimate aim of creating new imagery in the same style. Based on the tools described so far, this goal is already possible. We can use **Tactile** just as we would use commercial products like TesselMania! or Tess to design ornamental tilings and render them with decorations. These tools are a boon to artists because the computer can absorb all the tedium of replication and the difficulty of accounting for the constraints on a tile's shape. The artist is free to explore and to develop an intuition for the aesthetic ranges of different tiling types.

This design process might be called a "forward tiling process": we start from a simple shape

that is known to tile and evolve it until the outline sparks the imagination, evoking some real-world form. At that point we can tweak the shape to better convey the form, and paint the tiles. The Escher-like tesselations by contemporary artists are for the most part developed using a forward process. One must imagine that Escher himself worked this way, though he was certainly gifted with an uncommon intuition. His son recalls Escher's singular ability to tease animal forms and faces from the random patterns of clouds, wood grain, and random swirls of paint [49, Page 7].

The suggestive use of the word "forward" above implies that there ought to be an "inverse" tiling design problem. Whereas in the forward process we start with a mathematically simple shape and evolve it to create an artistic result, here we would start with a desired form and "devolve" it, imposing the mathematical constraints necessary to make that shape tile. The constraints will force some deformation upon the original shape, but with luck the change will be small enough that the result will still be recognizable.

We formalize the inverse tiling design problem as the "Escherization problem":

Problem ("ESCHERIZATION"): Given a closed plane figure S (the "goal shape"), find a new closed figure T such that:

1. T is as close as possible to S; and
2. T admits a monohedral tiling of the plane.

As Section 2.3 points out, little is understood about monohedral tilings in general. To make Escherization tractable, we immediately retreat from the fully general problem statement and restrict our attention to IH (later, we will consider Escherization over some other families of tilings). The previous sections have shown how to parameterize the space of isohedral tilings and implement that parameterization as a software library. We also know (as was mentioned at the start of Section 4.3) that the isohedral tilings are a good match for the sorts of tesselations Escher created.

The nature of a solution to the Escherization problem hinges on the interpretation of the word "find." We may imagine a space $\mathcal{S}$ formed by the set of all possible shapes in the plane, together with some sort of metric that measures the "closeness" between two shapes. The prototiles of isohedral tilings form a subspace $\mathcal{T}$ of shape space. A given goal shape S is a point somewhere in $\mathcal{S}$. Optimistically, an analytic solution to Escherization would project shape space onto its tiling subspace, moving S to a closest point $T \in \mathcal{T}$.

Though appealing to the mathematical aesthetic, such an approach is infeasible. However, the mental model provided by S and T, and the idea of closeness as a metric, provide the intuition needed to formulate an attack on Escherization. We envision starting from some known prototile and trying to move through T in a way that brings us ever closer to S. In other words, our proposed solution will be based upon continuous optimization, a common technique in computer science for solving problems that are not easily invertible. Using optimization, we may never reach a prototile that is globally optimal with respect to S, but perhaps we can get close enough to satisfy the artistic intent of S.

A continuous optimization problem consists of a *configuration space* and an *evaluation function* that maps configurations to real numbers. The goal is to find the configuration that minimizes (or equivalently, maximizes) the value of the evaluation function. In the optimization procedure for Escherization, the configuration space is $\mathbb{R}^n$; the parameters that describe the shape of an isohedral prototile can readily be expressed as a tuple of real numbers, as will be shown below. Other optimizers (see, for example, the work of Agrawala on route maps [4]) maintain a more abstract configuration space and rely on a user-defined function to suggest perturbations to configurations.

The evaluation function in Escherization is precisely the closeness metric in shape space — by decreasing the value of that metric (for some formal definition of closeness), we find tiles that are more like the goal shape.

The remainder of this section formalizes the Escherization algorithm by providing an efficient closeness metric and a framework in which the continuous optimization takes place. The implementation is able to find reasonable-looking tiles for many real-world shapes (see, for example, the "Escherized" version of Escher's own self-portrait, shown in Figure 4.15(a)). Subsequent sections will then build upon the ideas presented here by showing other forms of Escherization.

4.5.1   The shape metric

The Escherization problem raises the difficult question of how to compare two shapes. An answer should be in the form of a metric that would take two outlines and return a nonnegative real number; zero would mean that the outlines are identical, and higher positive values would denote shapes that are increasingly dissimilar. We would also like the metric to be insensitive to rigid motions or

138

uniform scaling of either of the shapes.

Fortunately, such metrics have been developed by computer vision researchers. We use the metric devised by Arkin *et al.* for comparing polygons [6]. Their metric represents the input polygons as *turning functions*, functions that map fraction of arc length in a polygon to the angle of the polygon at that point (with respect to the positive x axis). Turning functions are naturally translation- and scale-independent. Vertical and horizontal translation of a turning function correspond respectively to rotation of the polygon and movement of the point where the measurement of arc length begins. They show that the minimal L^2 distance between all possible translations of the two turning functions satisfies the mathematical definition of a metric and corresponds to an intuitive measure of closeness. It turns out that this minimum is achieved at one of a relatively small number of translations of the turning functions, allowing for an efficient algorithm that searches these configurations for that minimum. They provide an implementation that compares two polygons with m and n vertices in time $O(mn \log n)$.

This algorithm has the drawback that detail is assumed to be directly proportional to arc length. In other words, two pieces of a polygon's boundary with the same length carry the same amount of detail even if one piece packs that detail into a much smaller space. It helps to keep this fact in mind and aim for a consistent level of detail when creating a goal shape to be Escherized. I have also had some success running a low-pass filter over the goal shape to smooth out local areas of high detail.

I use the polygon comparison metric for both polygons and subdivision curves. A subdivision curve is simply approximated by a polygon with a large number of vertices.

4.5.2 *Optimizing over the space of tilings*

Armed with a set of tilings, parameterizations over those tilings, and a good shape metric, we are now ready to address the problem of building an optimizer that can search over the space of those tilings to find an instance whose prototile is close to the goal shape.

Our optimizer is based on simulated annealing. It works roughly as follows:

function FINDOPTIMALTILING(GOALSHAPE, FAMILIES):

    INSTANCES ← CREATEINSTANCES(FAMILIES)

    while $||$INSTANCES$|| > 1$ **do**

        for each i **in** INSTANCES **do**

            ANNEAL(i, GOALSHAPE)

        end for

        INSTANCES ← PRUNE(INSTANCES)

    end while

    return CONTENTS(INSTANCES)

end function

The optimizer takes as input a goal shape and a set of isohedral families in which to search for an optimal tiling. The optimizer begins by creating a set of multiple `IsohedralTile` instances for each isohedral type. The "default" tiling for each isohedral type is chosen to be the type's underlying Laves tiling. The Laves tiling can always be expressed within the parameterization space of any isohedral type of the same topology. So each new instance starts with the Laves tiling and applies a small random perturbation to its shape.

The optimizer then calls a re-entrant simulated annealing procedure to improve each one of these instances. (This ANNEAL() procedure is discussed in more detail below.) After each of the instances has been optimized to some degree, the instances are evaluated according to the shape metric, and the worst ones are removed. The annealing is continued on the remaining instances. This iterative process of alternately pruning the search space and then improving the remaining instances is repeated until just a single `IsohedralTile` is left. This tiling is returned as the output of the Escherization algorithm.

The set of tiling types to be passed to the optimizer is at the user's discretion, but some guidelines can be used to drastically reduce the number of types that need to be checked. As has been mentioned before, some isohedral types "subsume" others, in the sense that a tiling type with a symmetric prototile can be seen as a special case of a related tiling type with no prototile symmetry. The more symmetric type, being more constrained, can get no closer to the goal shape than its asymmetric counterpart, and so it need not be included in the optimization. The best possible solution can be found by optimizing only over the 28 types with asymmetric prototiles (correspond-

ing to the 28 Heesh types). Once a solution is found, it is always possible to re-run the optimizer with the symmetric children of the solution's tiling type, in search of a solution with more prototile symmetry.

The annealer is a re-entrant procedure, which works roughly like this:

> **procedure** ANNEAL(TILING, GOALSHAPE):
>> **loop**
>>> **while** $T > T_{\min}$ **do**
>>>> OPTIMIZETILING(TILING, GOALSHAPE, T)
>>>> $T \leftarrow$ REDUCE(T)
>>>> **suspend**
>>> **end while**
>>> SMOOTHEDGESHAPES(TILING)
>>> SPLITEDGESHAPES(TILING)
>>> $(T, T_{\min}) \leftarrow$ UPDATESCHEDULE(T, $T_{\min}$)
>> **end loop**
> **end procedure**

The annealer takes a given tiling instance and a goal shape as input. It acts like a coroutine, periodically suspending itself while maintaining its state, so that it can be resumed seamlessly later. FINDOPTIMALTILING runs each annealer for a few hundred iterations at a time, so that they can be compared against each other to track progress. The ANNEAL procedure takes a number of cooling steps, reducing the "temperature" at each step. Within this inner loop, it makes a call to a procedure called OPTIMIZETILING(). This procedure implements the continuous multidimensional simulated annealing algorithm described by Press *et al.* [120, Section 10.9]. Their algorithm is based on a fuzzy version of the "downhill simplex method," where a simplex of proposed solutions evolves towards a global minimum by pushing the worst solution through the hyperplane formed by the remaining ones.[2] The procedure attempts to improve all of the parameters of the tiling, comprised of the parameterization of the tiling vertices and the positions of the shape vertices. The procedure always accepts a downhill step (one that brings the tile shape closer to the goal shape) and sometimes accepts an uphill step, with probability depending on the temperature T. Once the temperature has

---

[2]This technique should not be confused with the simplex method of linear programming.
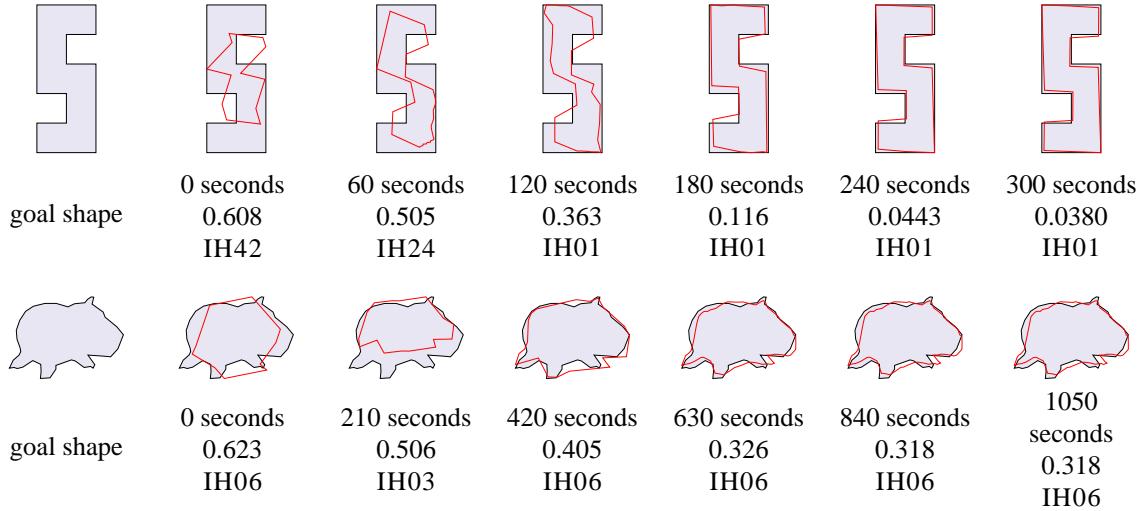
cooled to some minimum temperature $T_{\min}$, the inner loop terminates. At this stage, the algorithm runs through the vertices of the tile and removes any vertices that are nearly collinear with their neighbors; these vertices are not being used by the optimizer to improve the tile shape, and are needlessly slowing down the optimization. Next, the edges of the tile are subdivided, reintroducing vertices where detail might get added as the optimization continues. (Note that during this process, the dimensionality of the annealing problem may change, and we must ensure that OPTIMIZETIL-ING can cope with the change.) The cycle of smoothing and subdividing edges is highly effective in keeping degrees of freedom only where they are needed. Finally, the cooling schedule restarts, generally with slightly lower values of T and $T_{\min}$.

One additional part of the optimization, which is not shown in the pseudocode and which is optional, is to automatically convert the vertices of the tiles into control points for subdivison curves after a certain stage in the optimization. The annealer can then incorporate the weights on the subdivision control points as additional degrees of freedom.
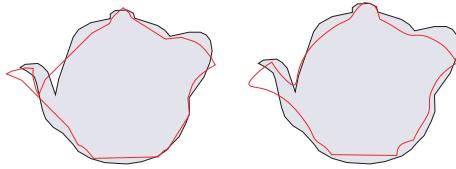
The use of simulated annealing is subject to the usual practicalities. First, the success of the optimization for a single instance of a single tiling type depends to some extent on the initial shape of the tiling polygon and the initial positions of the shape vertices. I therefore generally start with multiple, randomly perturbed instances for each tiling type. An interesting alternative would be to seed the initial tiles with segments from the goal shape's outline. As with any simulating annealing algorithm, the choice of cooling schedule can also make a difference. I use a very simple approach where the temperature T is multiplied by a factor of ϕ after every N iterations, with $T = 0.1$, $N = 250$, and $\phi = 0.9$ to start. When the temperature reaches 5% of its initial value ($T_{\min} = 0.05T$), the optimization resets, lowering the starting and minimum temperatures by a factor of 0.6, increasing the number of iterations N by a factor of 1.2, and reducing the temperature multiplier ϕ by a factor of 0.1. I did not spend a lot of time "optimizing" this cooling schedule, so other reasonable choices would probably work equally well or better.

4.5.3 Results

Figure 4.13 shows snapshots from two sample runs of the Escherizer. The goal shape in the first run is a simple test polygon, part of a series used to verify and tune the optimizer. The second goal

Figure 4.13 Timelines for two sample Escherization runs. Each step shows the current best tile in the system (in red) overlaid on the goal shape. The caption indicates the elapsed time, the score for that tile, and its isohedral type.
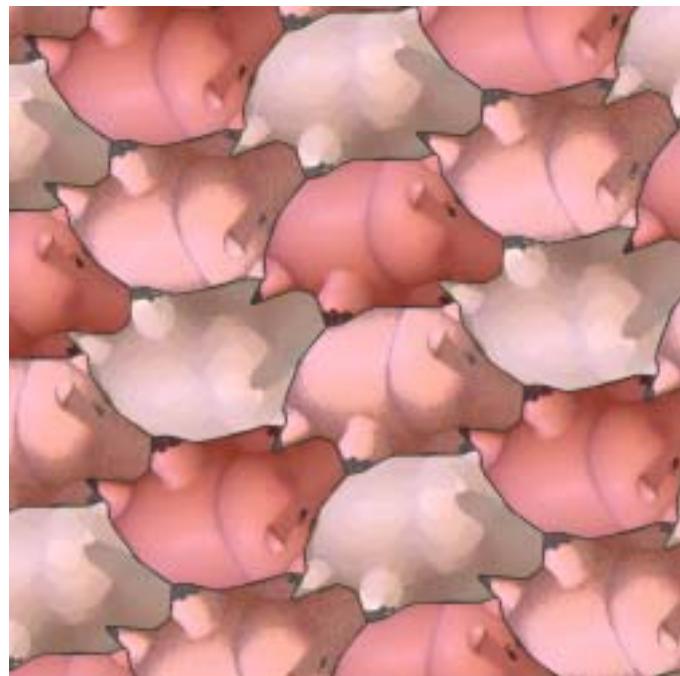
Figure 4.14 A comparison between the tile returned by the optimizer and the same tile with user modifications. Note also that the second tile has subdivision enabled.

shape is a more typical real-world outline. The more complicated shape takes longer to run, and the convergence is not quite as complete (as should be expected from a real-world outline).

The optimizer does not require user intervention, but it does run interactively so that its progress may be watched. In practice, some constructive intervention is possible while watching the running Escherizer. If it is clear that the goal shape will simply not work as a tiling, the process can be interrupted. On the other hand, if one particular tiling type seems to be performing very well on the goal shape, the program can be stopped and restarted with many instances of that type, resulting in a narrower and deeper search.
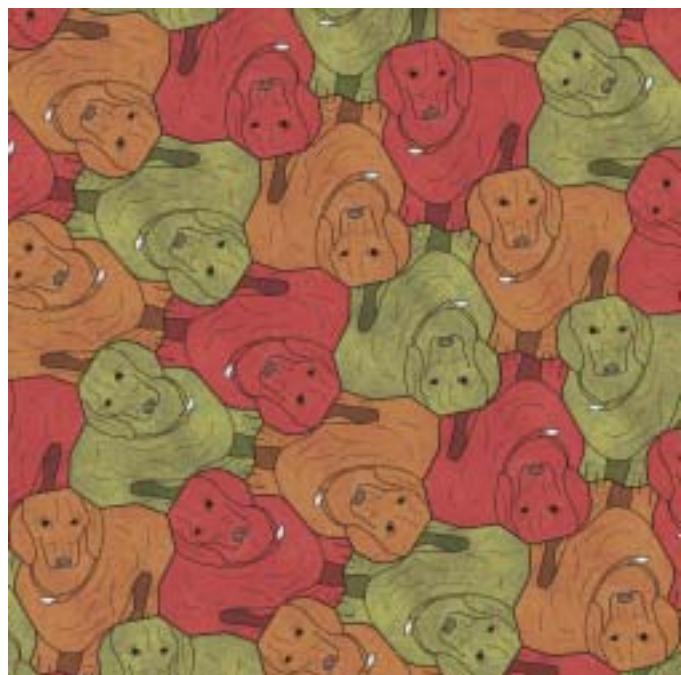
Figure 4.14 shows the tile result produced by the optimizer for a teapot image, followed by the

(a) *Escher's Escher Escherized* (IH1)

(b) *Pigs in 2-Space* (IH3)

Figure 4.15 Some examples of Escherized images and the tilings they generate. Hamm the pig appears courtesy of Disney/Pixar.
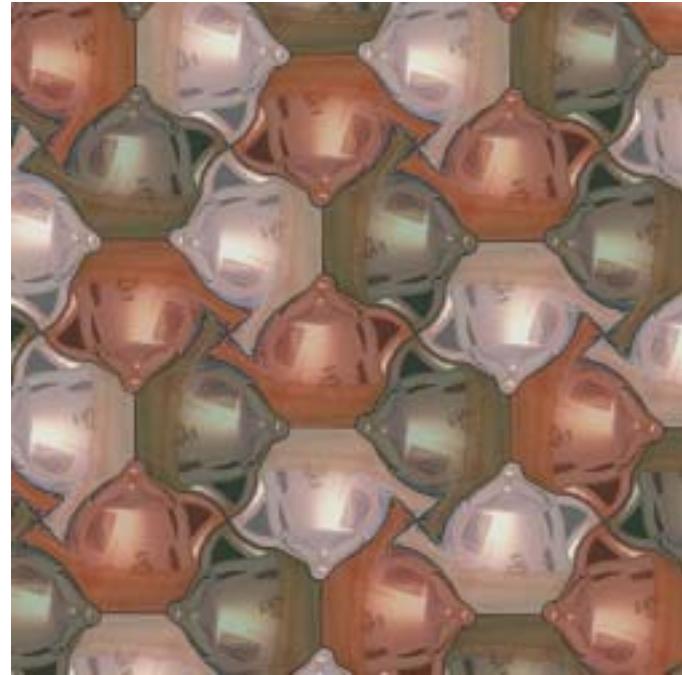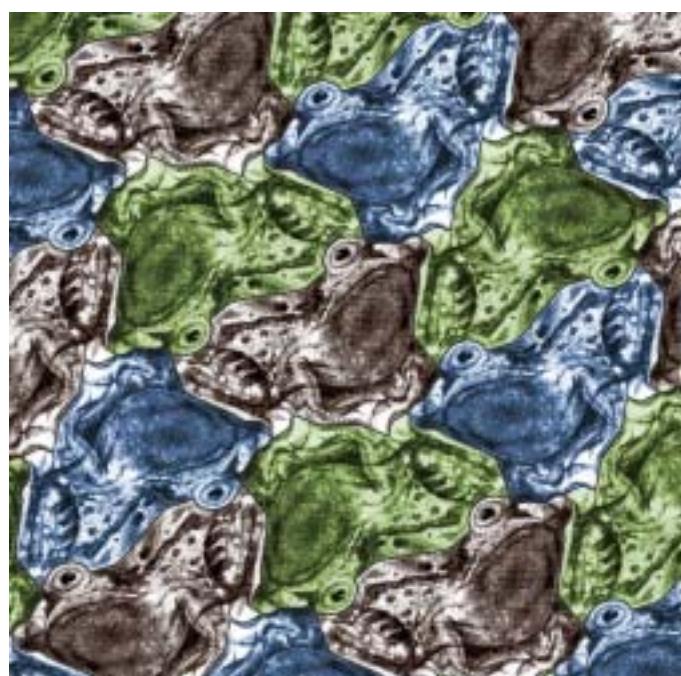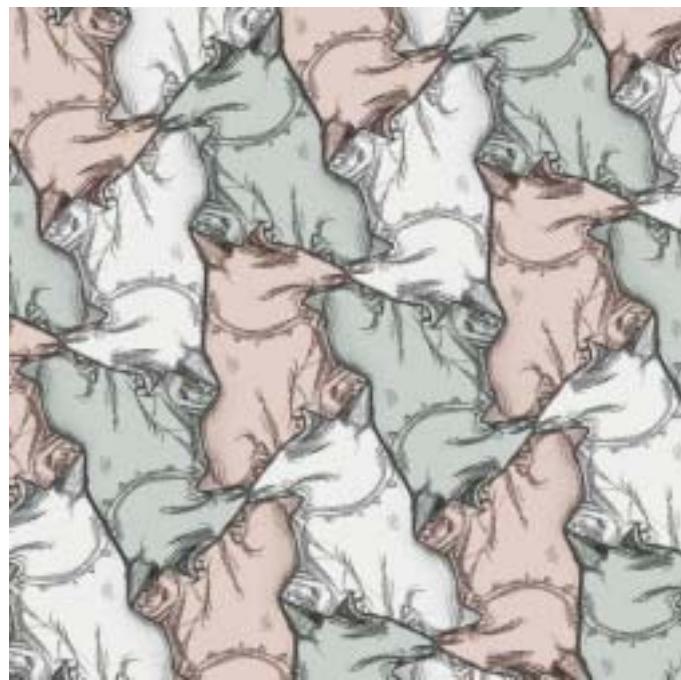
(c) *Dogs; Dogs Everywhere* (IH4)
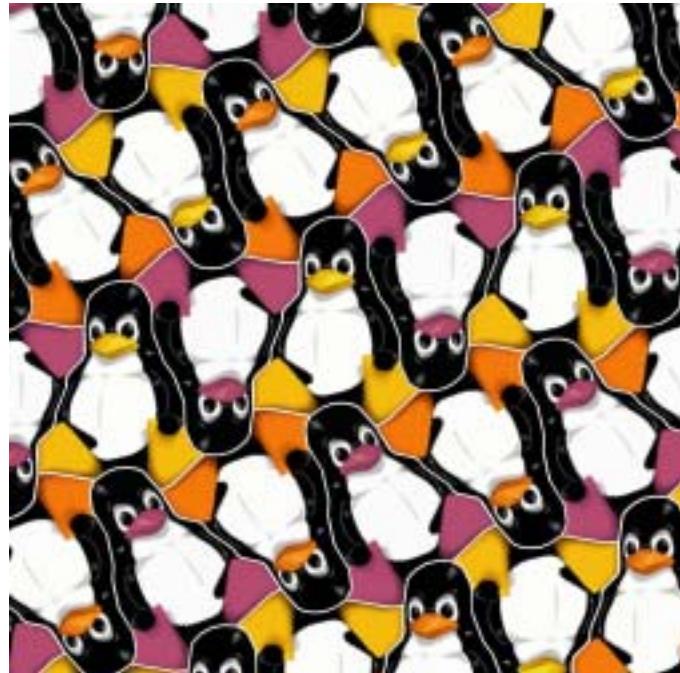
(d) *Weiner Dog Art* (IH5)

Figure 4.15 (continued)

(e) *Tea-sselation* (IH28)

(f) *Twisted Sisters* (IH86)

Figure 4.15 (continued).

(g) *Sketchy Dogs* (IH6)

(h) *A Plague of Frogs* (IH6)

Figure 4.15 (continued). Sketchy dog appears courtesy of Disney and Pixar.

(i) *Tux-ture mapping* (IH6)

(j) *Bubbles the Cat* (IH1)

Figure 4.15 (continued). Tux the penguin appears courtesy of Larry Ewing.

tile after a small amount of hand-tweaking in the interactive editor. For the results shown here, the edits took a minute or two to perform and were fairly typical of the experience of creating tilings using Escherization. In general, hand editing of Escherized outlines should be expected and even welcomed. The success of Escherization does not lie in its ability to replace an artist with a black box tool that produces tilings from thin air. An artist will have a specific intent in mind, expressed roughly (but perhaps not precisely) with a goal shape. The Escherization algorithm achieves a "quantum leap of aesthetics" by proposing a tiling with a prototile that roughly resembles the goal shape, a feat that exceeds the grasp of most human intuition. Still, a result that is perfectly sound mathematically might lack in artistic merit. All Escherization sees of the artist's goals is the imperfect communication of those goals as a shape to approximate. As part of the process of turning the Escherized tiling into a finished drawing, it is only natural that the artist would want to start by examining the tile shapes themselves.

The Escherization workflow starts with the user defining a goal shape. Typically, this goal shape is specified by tracing a feature in an image (say, the outline of a favourite pet). An obvious enhancement would be to incorporate computer-assisted tools such as intelligent scissors to extract contours in an image [114].

The natural choice for decorating an Escherized tile is to use the interior of the goal shape in the image that was originally traced. Using the correspondence provided by the comparison metric, I do a Beier-Neely style image warp [8] to deform the interior of the goal shape in the source image into the interior of the Escherized tile shape. When the deformation is not too great, the result is an attractive tiling out of motifs that resemble the original image. When the automatically-determined correspondence produces too much distortion (which can happen when the goal shape and tile shape differ in level of detail), it can be edited by hand to create a better match. This use of image warping to decorate tiles is valuable in a somewhat sneaky way as well: it can help to cover up deficiencies in the tile shapes. The eye latches onto features of the decoration, and is willing to forgive a certain lack of believability in the shapes of tiles. Escher himself was no stranger to this technique. His deft addition of features as simple as circles for eyes could turn even the most abstract form into a whimsical creature. (In *Understanding Comics*, McCloud eloquently illustrates this point in the context of comics [110, Page 32].)

To further increase the appeal of an image-based rendering, I apply various filters and effects to

the warped tile image before replication. This post-processing step gives the artist creative control over the appearance of the final tiling. Alternatively, as was discussed in Section 4.4.3, the original image can be discarded entirely in favour of vector-based art created using a commercial drawing package. The artwork is then rasterized and re-inserted into the rendering pipeline. The user can also create multiple versions of the decorated tile, which are placed according to the tiling's colouring.

I have used the Escherization algorithm and decoration tools to produce a number of ornamental tesselations from various sources of imagery. These results can be seen in Figure 4.15. The vector-based art for *Bubbles the Cat* (j) was created using the prototype marking system. The decorations for *Weiner Dog Art* (d), on the other hand, are a combination of line art created in Adobe Illustrator and textures and colouring created in the GIMP [53]. *Pigs in 2-Space* (b) uses various artistic filters from Adobe Photoshop.

4.6   *Dihedral Escherization*

The Escherization algorithm described in the previous section is limited to producing monohedral tilings, and while an artist can use the system to produce a variety of designs in the style of M. C. Escher, we have only scratched the surface in terms of the complete set of tesselations he created. In addition to the monohedral tesselations that make up his collected symmetry drawings [124], we also find many tesselations with two or (less frequently) more motifs.

Although these multihedral tilings make up the minority of his drawings, they are a very important aspect of his work. Some of his most famous prints (for example, *Sky and Water*, *Verbum*, and *Metamorphosis II*) make use of one or more dihedral tilings. Furthermore, the use of multiple motifs agrees with Escher's predisposition to imbue his work with narrative structure (most clearly expressed in the *Metamorphosis* prints [49, Page 48]). A single motif, unchanging forever except for colour, is largely incapable of telling a story. With two or more motifs, suddenly there is the opportunity for contrasts, for harmony or discord, for interaction and drama. In symmetry drawing 45 [124, Page 150], better known as "Heaven and Hell," or "Angels and Devils," Escher plays with the balance that exists between good and evil, light and dark. Drawing 63 [124, Page 165], later immortalized in the print *Encounter*, depicts smiling optimists and frowning pessimists attempting to work out their differences.

Naturally, we would like to rework the Escherization algorithm to handle two goal shapes simultaneously and produce dihedral tilings as output. We can immediately formulate a revised version of the original Escherization problem:

Problem ("DIHEDRAL ESCHERIZATION"): Given closed plane figures S_1 and S_2 (the "goal shapes"), find new closed figures T_1 and T_2 such that:

1. T_1 and T_2 are as close as possible to S_1 and S_2, respectively; and

2. T_1 and T_2 admit a dihedral tiling of the plane.
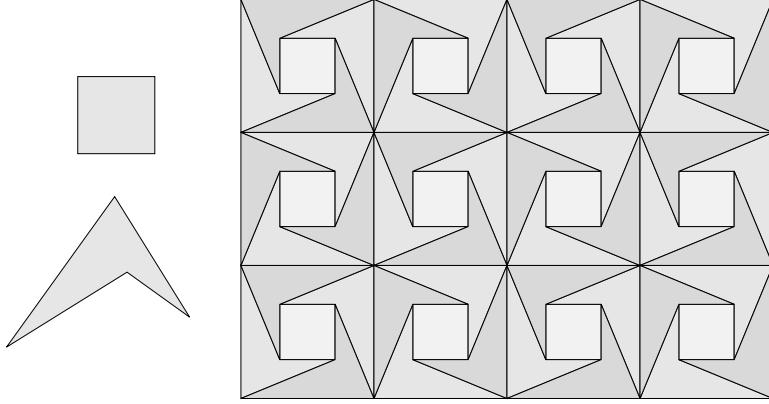
The success of the monohedral Escherization algorithm suggests that we should take a similar approach here. I structure the dihedral Escherization algorithm as a continuous optimization over some space of dihedral tilings. At every step in the optimization, the parameters will somehow be converted into a pair of tile shapes, whereupon I can use the original polygon comparison metric (twice) to see how similar these tile shapes are to the two goal shapes. Some combination of the two comparisons will act as an evaluation function for the dihedral optimization.

The majority of the work is therefore limited to the definition of an appropriate space of tilings to plug in to the optimization. In this section, I present two spaces of dihedral tilings that yield satisfactory results. The first consists of the *split isohedral tilings*, which naturally capture the sorts of two-motif tesselations Escher created. The second set of tilings is made up of generalizations of Penrose's aperiodic tile sets $P2$ (kites and darts) and $P3$ (rhombs).

4.6.1 Split isohedral Escherization

Our search for a space of useful dihedral tilings begins with Escher himself. A meticulous notetaker, he documented his exploration of two-motif systems [123, 124]. In every case, he starts with one of his monohedral systems and draws a path through the prototile to break it into two shapes. When that division is copied symmetrically to all other tiles, the result is a dihedral tiling.

Escher's splitting process was carried out consistently for every tile, yielding a dihedral tiling with two prototiles A and B. It follows that the resulting dihedral tiling will have exactly two transitivity classes, one containing all the A tiles and the other containing all the B tiles. In other
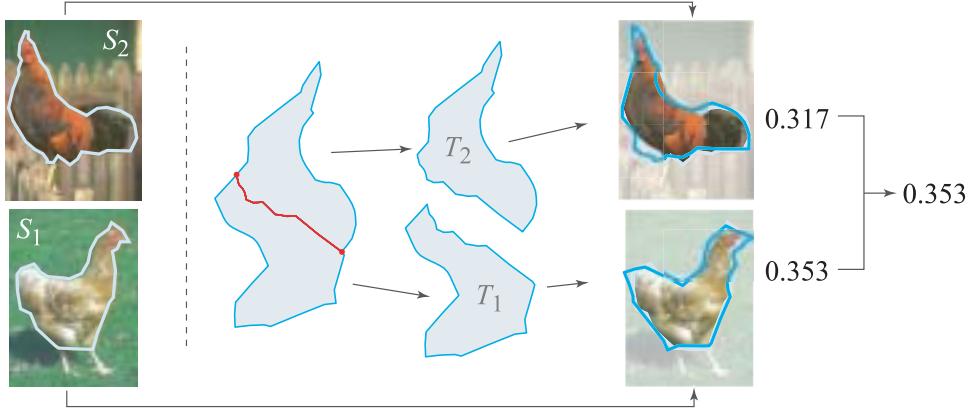
Figure 4.16 A 2-isohedral tiling with different numbers of A and B tiles. The two prototiles, a square and an irregular dart, are shown on the left. The tiling on the right has four times as many darts as squares in every translational unit.

words, Escher's two-motif systems are just 2-isohedral tilings. We can conclude that to create dihedral tesselations in the spirit of Escher, we need to parameterize at most the 2-isohedral tilings.

On the other hand, Escher focused on only a small part of all possible 2-isohedral tilings. His tilings always have equal numbers of A and B tiles (in each translational unit), whereas there exist many 2-isohedral tilings that do not (an example is given in Figure 4.16). Moreover, his base tilings and splitting paths were always chosen to give a tiling that could be coloured using only two colours. As Schattschneider points out, this decision was motivated partly by the practicality of working in the medium of printmaking, which in its simplest form produces pictures with two colours: the ink colour and paper colour. Our space of tilings should be at least broad enough to contain Escher's understanding of two-motif designs.

Delgado-Friedrichs *et al.* carry out a complete enumeration of the 2-isohedral tilings [33]. They prove a general result that every $(k+1)$-isohedral tiling can be constructed from a k-isohedral tiling through a combination of two operations: SPLIT and GLUE. The SPLIT operation corresponds with Escher's use of a splitting curve, and they show that any *fundamental* (asymmetric) prototile of a $(k+1)$-isohedral tiling can be derived from a fundamental prototile of a k-isohedral tiling through a single SPLIT. The GLUE operation erases the edge between two adjacent tiles, producing tilings with symmetric prototiles, some of which have different numbers of A and B tiles. Using SPLIT

Figure 4.17 A summary of our process for split isohedral Escherization. On the left, two goal shapes S_1 and S_2 are traced from images. Next, the isohedral tile and splitting path are shown at a late stage in the optimization. The quality of this configuration is judged by breaking the tile into two shapes T_1 and T_2, which are then compared with S_1 and S_2. The optimization attempts to minimize the value at the right, the maximum of the two comparisons.

and GLUE, they show that there are over a thousand 2-isohedral tiling types, making a dihedral Escherization algorithm based on this full classification impractical. Instead, I look for an approach that exploits the work already done on the isohedral tilings.

For the purposes of dihedral Escherization, I concentrate on the *split isohedral tilings*, tilings that can be derived from an isohedral tiling through a single application of SPLIT. This family of tilings contains all of Escher's two-motif systems. It fails to distinguish those tilings that can be coloured with only two colours, though in some sense this fact is not critical because results can easily be printed in multiple colours. (Alternatively, from Escher's notes it would be possible to derive the necessary restrictions on tiling type and splitting path that would produce exactly his two-motif systems.) Later, in Section 4.6.2, we will encounter an even stronger restriction that yields a special family of dihedral Escher tilings dubbed "Heaven and Hell" patterns [38].

A split isohedral tile is represented by a class `SlicedTile`, a subclass of `IsohedralTile`. I augment the isohedral prototile with a splitting path, which is embodied by a new set of parameters: two parameters that control the path's start and end positions along the boundary of the isohedral prototile, and the degrees of freedom that control the path's shape.

The optimization process works in exactly the same way as the monohedral case, except that the evaluation function must be modified to handle two goal shapes instead of just one. The splitting path in a `SlicedTile` divides the isohedral prototile into two shapes T_1 and T_2. The evaluation function uses the shape metric defined in Section 4.5.1 to compare T_1 with S_1 and T_2 with S_2, and returns the maximum of the two comparisons as the optimization's evaluation function. By returning the maximum (a kind of L_∞ norm), we ask both tile shapes to approximate their goal shapes as closely as possible. This process is illustrated in Figure 4.17.

Let S_1' and S_2' denote reflections of S_1 and S_2. To find the best split isohedral tiling corresponding to two goal shapes S_1 and S_2, two instances of the Escherization algorithm are required: with S_1 and S_2, and with S_1 and S_2' (or S_1' and S_2). Although the shape comparison metric is insensitive to translation and rotation, it does distinguish between a polygon and its reflection. It might happen that S_1 and S_2 interact more favourably if one is reflected.[3]

Figure 4.18 shows some examples of Escherization using split isohedral tiles. One might guess that because of the need to match two goal shapes simultaneously, dihedral Escherization would have a lower success rate than monohedral Escherization. I have found that the additional degrees of freedom offered by the splitting edge more than compensate for the added complexity of the problem, and that the dihedral objective function performs comparably to or better than the monohedral one.
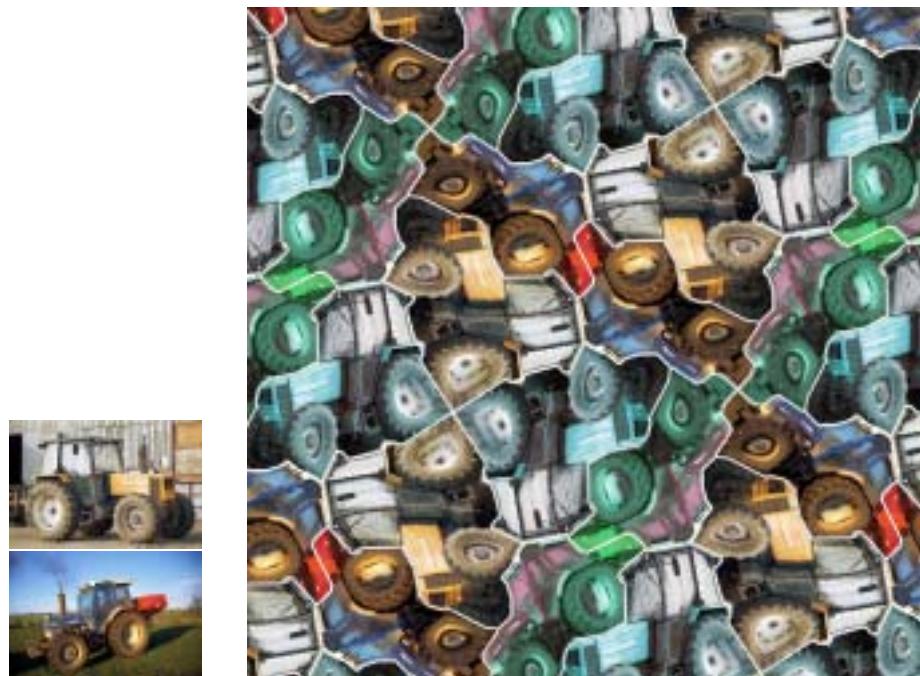
4.6.2 Heaven and Hell Escherization

Some of Escher's dihedral tesselations, such as *Heaven and Hell* [124, Page 150], have additional geometric structure that sets them apart from the rest. Not only is the tiling colourable using only two colours, but each colour is the exclusive domain of one of the classes of tiles; in Heaven and Hell, every angel is white and every devil is black. This sort of colouring is possible when every tiling vertex is surrounded by an alternating sequence of A and B tiles, or equivalently, when every A tile shares edges only with B tiles (and vice versa).
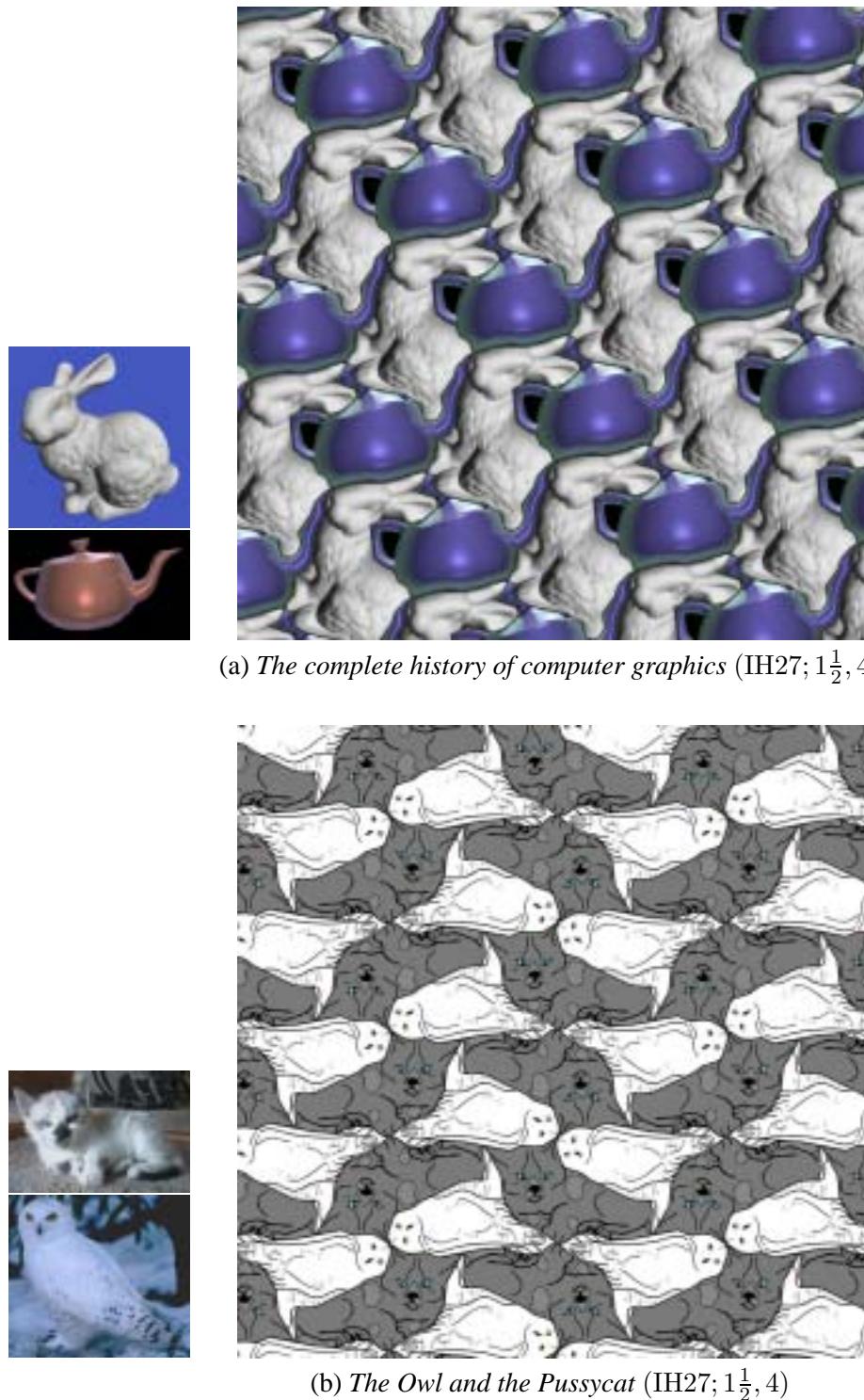
Aesthetically, such tilings are particularly effective because each transitivity class of tiles plays

---

[3]Note that only the relative parity matters; the flexibility of the isohedral tilings guarantees that (S_1, S_2) is equivalent to (S_1', S_2'), and that (S_1', S_2) is equivalent to (S_1, S_2').

(a) *Strange 'Tractors* (IH28)

(b) *Gödel, Bach (Braided): an Eternal Escherization* (IH2)

Figure 4.18 Examples of dihedral Escherization using the split isohedral tile method.

(c) *Pen/Rose Tiling* (IH1)

(d) *Rembrandt and Mrs. van Rijn* (IH1)

Figure 4.18 (continued)

(a) *The complete history of computer graphics* (IH27; $1\frac{1}{2}, 4$)

(b) *The Owl and the Pussycat* (IH27; $1\frac{1}{2}, 4$)

Figure 4.19 Examples of Heaven and Hell Escherization.

the role of ground to the other's figure: the A tiles exactly fill the negative space created by the B tiles. Moreover, the fact that the colours can be unambiguously associated with tile shapes allows them to become part of the identities of those shapes, as in the white angels and black devils of *Heaven and Hell*. Escher used this particular space of tilings to produce some of his best-known prints, several of which are mentioned at the beginning of this section.

The class of 2-isohedral tilings with this additional structure were enumerated by Dress [38], who dubbed them "Heaven and Hell" patterns. Based on an analysis using Delaney symbols, he classified the Heaven and Hell patterns into 37 distinct types. As always, we can develop an Escherization algorithm tailored to this particular classification, allowing us to construct these especially satisfying tesselations.

Twenty-nine of Dress's types can be constructed by applying the SPLIT operation to an isohedral tile and are therefore special cases of split isohedral tilings. The additional structure comes from a careful choice of locations for the endpoints of the splitting path. Analysis of Dress's classification (and examination of the diagrams in his paper) reveals that for these twenty-nine types, the path's endpoints are limited to a small number of possible locations. An endpoint will always be either one of the tiling vertices of the underlying isohedral prototile, or the midpoint of one of its tiling edges. If the isohedral prototile has n tiling vertices, we can enumerate this set of locations as $L = \{1, 1\frac{1}{2}, 2, 2\frac{1}{2}, \ldots, n, n + \frac{1}{2}\}$, where a whole number k refers to a tiling vertex and $k + \frac{1}{2}$ refers to the midpoint of the edge from k to $k + 1$. The numbering of the tiling vertices can be taken from the ordering of the edges in the tiling type's incidence symbol, as given in `isohedral.ih` (see Section 4.4). Each of the 29 types based on splitting can then be given the notation $(\mathrm{IH}m; a, b)$, where $\mathrm{IH}m$ denotes one of the 93 isohedral types, and where $a, b \in L$.

I represent the Heaven and Hell tiling $(\mathrm{IH}m; a, b)$ using a slightly modified version of the class `SlicedTile`. Under this modification, the endpoints of the splitting path are not treated as degrees of freedom in the optimization but fixed according to the locations a and b. Once the two endpoints are fixed in this way, the remainder of the dihedral Escherization algorithm can be applied as is.

As has been mentioned before, tiling types can often be placed in a hierarchy, where types with asymmetric prototiles subsume their more symmetric children. Dress discusses this ordering, and his diagram shows the 37 Heaven and Hell types laid out according to their heirarchy. His explicit ordering makes it easy to recognize that of the 29 types based on splitting, we need only optimize

over 12, as the remaining 17 are subsumed under them by prototile symmetry. Using the notation given above, the 12 types are as follows:

$$(\text{IH1}; 1, 4), (\text{IH2}; 2, 5), (\text{IH3}; 2, 5), (\text{IH5}; 1, 4), (\text{IH27}; 1\tfrac{1}{2}, 4), (\text{IH31}; 1, 3)$$

$$(\text{IH33}; 1, 3), (\text{IH41}; 1, 3), (\text{IH43}; 1, 3), (\text{IH47}; 2\tfrac{1}{2}, 4\tfrac{1}{2}), (\text{IH52}; 1, 3), (\text{IH55}; 2, 4)$$

The remaining eight types in Dress's classification require the use of the GLUE operation and produce tilings with different numbers of A and B tiles. Escher did not create any two-motif patterns with this property. In principle, it would be simple to add these types to the Escherization process, although it would require the implementation of a new class of tiling. I choose not to consider these types in the present work.

Figure 4.19 gives two examples of Heaven and Hell Escherization. The special figure-and-ground relationship maintained by the two prototiles is best depicted by using only two colours: one colour each for the A tiles and B tiles.
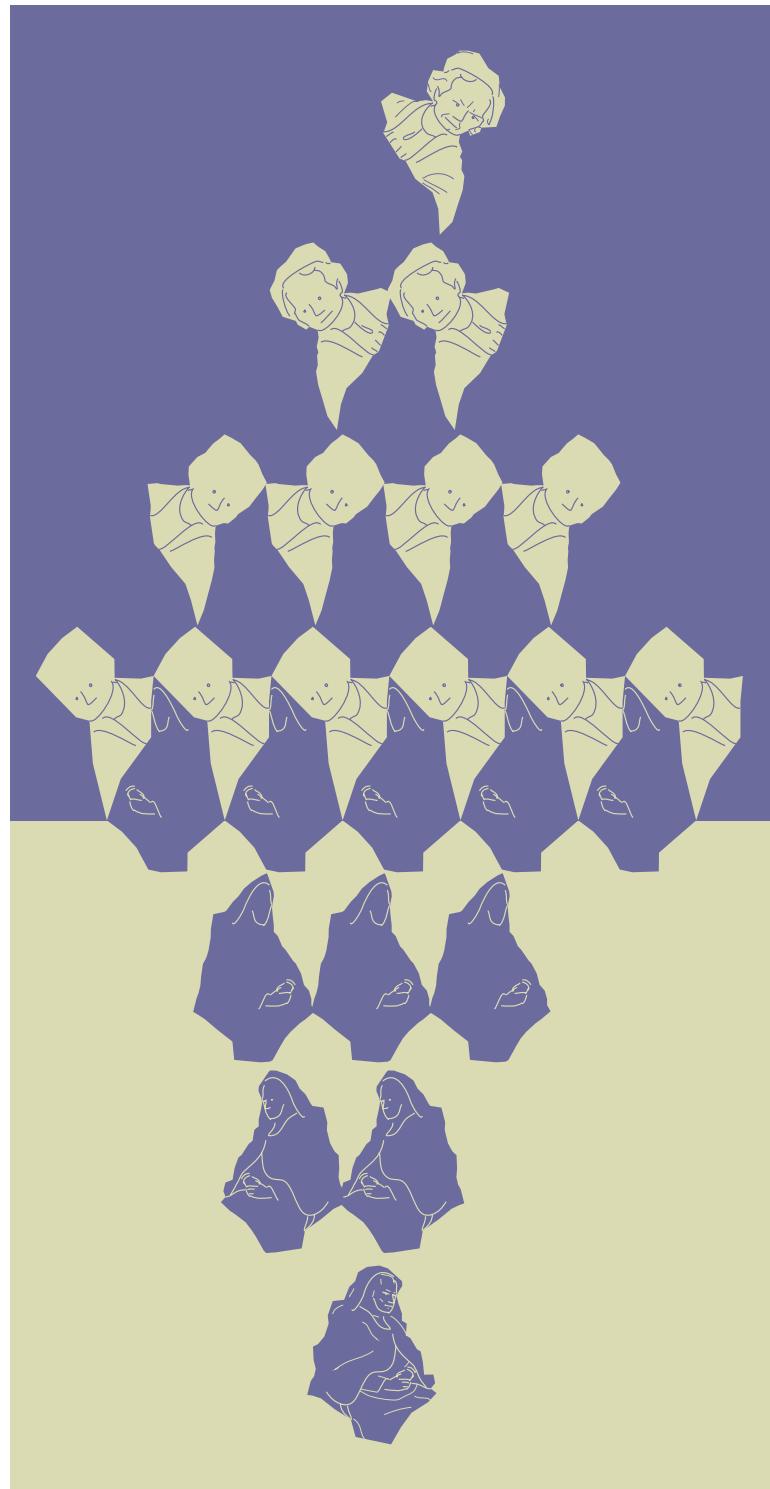
4.6.3 Sky and Water designs

Escher's print *Sky and Water* is a very special application of Heaven and Hell tilings. What starts out in the center of the print as a dihedral tiling of stylized fish and birds evolves towards the top and bottom into realistic drawings: birds above, and fish below. Escher used this device in many prints and sometimes multiple times in a single print (as in *Verbum* and *Metamorphose II*).

It is critical that the central tiling where the birds and fish interface be a Heaven and Hell tiling, and not just a split isohedral tiling. The stylized birds evolve into the background for the realistic fish (and vice versa), and so the tiling needs to be colourable with one colour for each tile shape.

Heaven and Hell Escherization seems very well suited to the construction of Sky and Water designs because the realistic goal shapes are already part of the process that leads to the stylized tile shapes. To turn a Heaven and Hell tiling into a Sky and Water design, it suffices to gradually blend the tile shape into the goal shape as tiles are placed successively farther from a given "interface line."

I have embedded the basic Heaven and Hell Escherization algorithm into a suite of interactive tools for constructing Sky and Water designs. One tool lets the user specify a set of tiles to draw and an interface line. Another tool lets the user add decorations to tiles with monochromatic vector-based strokes. Each stroke is a sequence of Bézier curves with user-specified widths; the curves are

Figure 4.20 An example of a Sky and Water design, based on the goal shapes of Figure 4.18(d).

fit to the user's drawing gestures using the method of Schneider [125]. Additionally, each stroke is given a "priority" that determines how far from the interface line the tile must be before the stroke is drawn. This approach allows for a prioritized sequence of strokes ordered by their importance in expressing a stylized version of the goal shape.

Finally, a renderer assembles the final drawing (in Postscript), taking the output of the other two tools as input, together with colours for the A and B tile shapes. For every tile, the renderer interpolates that tile with its corresponding goal shape by an amount determined from its distance to the interface line. The interpolation is carried out so that any tiles that touch or cross the line are set to the tile shape, and maximally distant tiles are set to the goal shape. The A tiles are then drawn over a solid background of the B tile colour, and vice versa. Finally, the strokes are warped into place and drawn if they have sufficiently high priorities.

Figure 4.20 shows an example of an Escherized Sky and Water design.

4.6.4 Escherization using Penrose tiles

In Section 3.10, I showed how to construct quasiperiodic Islamic star patterns based on a lattice-projection method that produces tilings by rhombs. It happens that a suitably-chosen lattice projection yields the Penrose set $P3$, consisting of a thin and a thick rhomb. Here, we return to the subject of Penrose tiles and consider the question of creating Escher-like tilings based on Penrose sets $P2$ and $P3$.
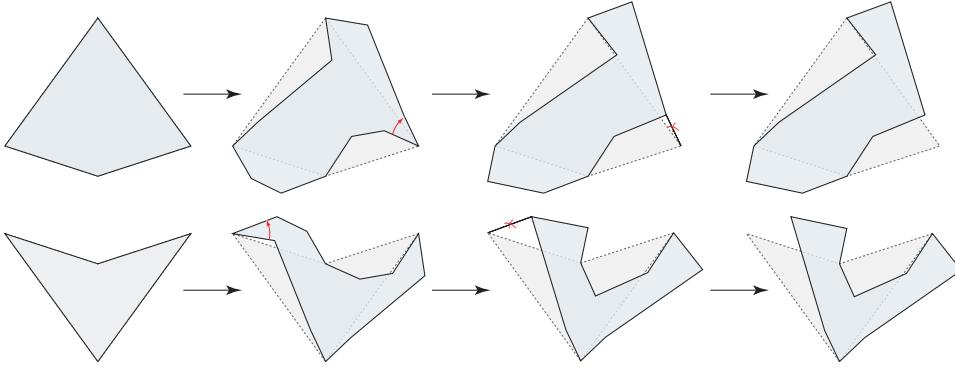
Unfortunately, Escher did not live to see the development of Penrose tilings, and so we can only imagine what sorts of creatures he might have discovered in them. Penrose himself, who corresponded regularly with Escher, expresses his regret at the missed opportunity [118]. He also gives an example of what Escher might have drawn: a modification of the aperiodic tile set $P2$ where the kites and darts have been turned into chickens. Over time, other artists have created Escher-like designs based on Penrose tiles, though traditional periodic designs are still much more popular. This discrepancy can be explained at least partly by the fact that the range of legal shapes of Penrose tiles is far from obvious. Whereas any layperson can quickly grasp the structure of the simpler isohedral types, understanding the Penrose tiles requires some awareness of the underlying mathematics.

As was pointed out in Section 2.3.3, the basic kite and dart shapes are not an aperiodic tile set; many different periodic tilings can be constructed from them. In order to enforce aperiodicity, the tiles must be augmented with matching conditions that determine the legal ways one tile may be placed next to another. These matching conditions are expressed in several different ways. Symbolic colours can be assigned to the tile's vertices, in which case every tile that shares a tiling vertex must have the same colour there. Or the edges can be labeled, in which case two tiles that share an edge must label it the same way. Most importantly, the edges can be deformed so that tiles only fit together in the desired ways. Grünbaum and Shephard give such geometric matching conditions for Penrose's aperiodic sets $P2$ and $P3$ [68, Section 10.3]. In both cases, the matching conditions are boiled down to two non-congruent **J** edges and the way they are arranged around the two tiles in the set.

The geometric matching conditions immediately give rise to an Escherization algorithm for Penrose tiles. The tiling vertices remain fixed, and the optimization operates on the degrees of freedom in the two fundamental edge shapes. These edge shapes are assembled into two tile shapes that are then compared against two goal shapes as usual.

Unfortunately, this interpretation of the possible shapes of Penrose tiles is rather limited, as can be seen in Grünbaum and Shephard's reproduction of Penrose's aperiodic chicken tiling [68, Figure 10.3.13]. They overlay the chickens with the corresponding unmodified tiling. The registration of these two tilings reveals that the chickens have tiling vertices that are different from those of the original tiling! There are additional degrees of freedom to the Penrose tilings that must be explored and exploited if we are to extend the reach of aperiodic Escherization. Accordingly, I derive a parameterized space that generalizes the aperiodic set $P2$, the Penrose kite and dart. The same arguments apply to set $P3$, the Penrose rhombs.

The additional degrees of freedom are wrapped up in the positions of the tiling vertices, and so we seem to be looking for yet another set of tiling vertex parameterizations. But some care must be taken when thinking about the tiling vertices. The Penrose tilings make no guarantee about the transivity of tiles, meaning that in a given tiling, different instances of the same prototile will be surrounded differently. As a result, two tiles with the same shape may nevertheless have different tiling vertices. In that case, it makes little sense to speak of parameterizing a prototile's tiling vertices once and for all.
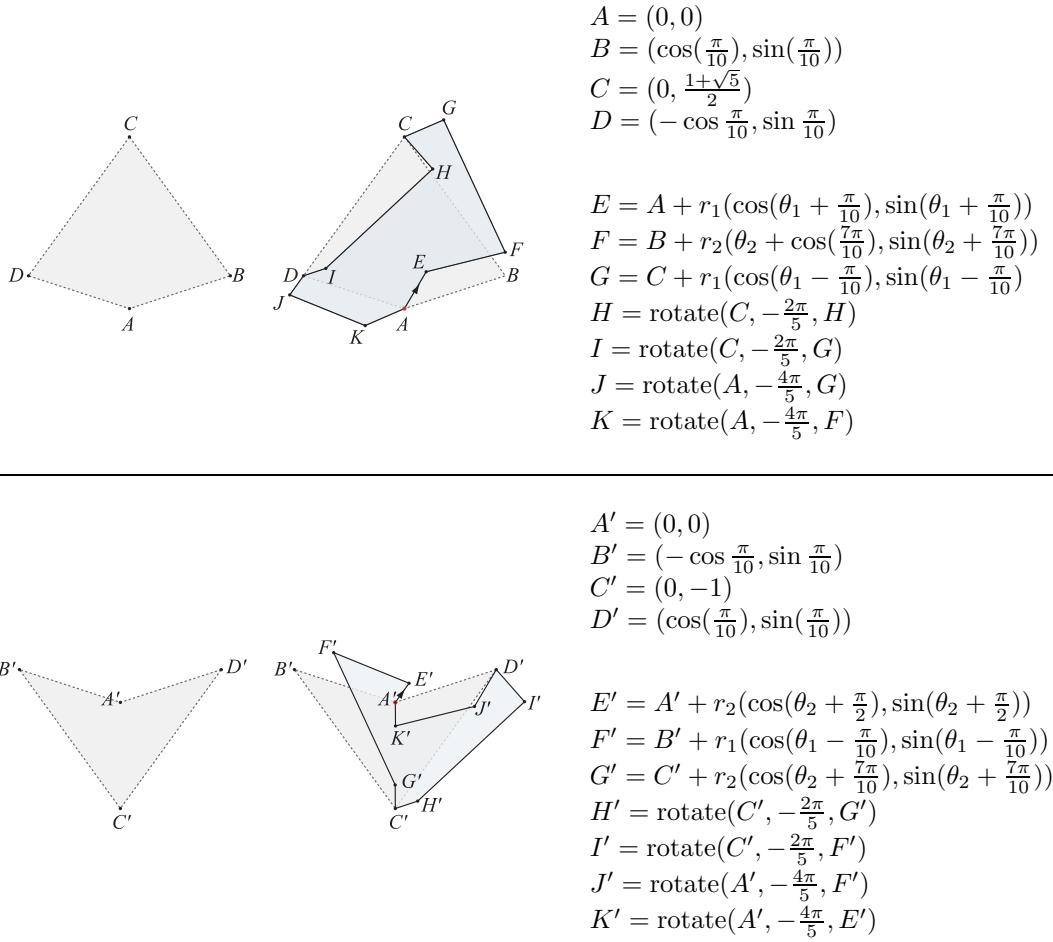
Figure 4.21 An illustration of how a tiling vertex parameterization can be derived for the Penrose kite and dart. The original edges are modified using Grünbaum and Shephard's edge matching conditions [68, Page 539]. When adjacent edges coincide, they are removed, displacing the tiling vertices between the edges. The kite and dart each have one unconstrained tiling vertex. The others are all implied by the original matching conditions.

By experimenting with the geometric matching conditions, I have discovered an extended set of points that can be parameterized like the tiling vertices of an isohedral tiling. I call these points the prototile's *quasivertices*. The quasivertices include all the points on a prototile's boundary that act as tiling vertices somewhere in a Penrose tiling, and some additional points that are forced into existence by the creation of these new tiling vertices.
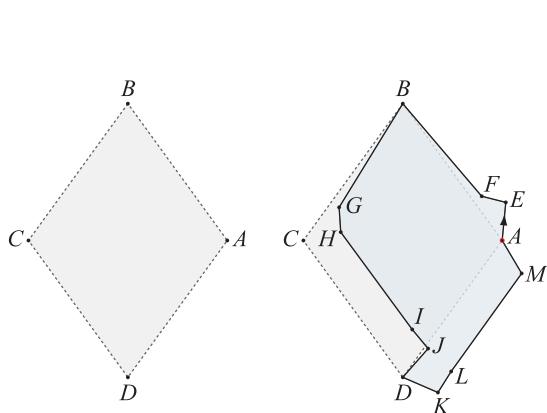
Figure 4.21 shows how one may use the ordinary geometric matching conditions to derive the new set of parameterizable points for the kite and dart. The fundamental edge shapes are modified so that they partially overlap. The overlapping regions can then be excised from the tiles, producing new tiles that no longer share all of the tiling vertices of the original kite and dart. This process necessarily introduces other vertices into the shapes of the two tiles.

From Figure 4.21, we conclude that the quasivertices of the kite and dart can be parameterized using four real-valued parameters, determined by the positions of the tiling vertices created at the tips of the two excised regions. Similarly, four parameters suffice to parameterize the Penrose rhombs. Once the free parameters are understood, we can derive explicit formulae for the positions of the quasivertices. Formulae are given for the kite and dart in Figure 4.22, and for the rhombs in Figure 4.23.

This additional effort at parameterization may at first seem superfluous, because once portions of

$A = (0, 0)$

$B = (\cos(\frac{\pi}{10}), \sin(\frac{\pi}{10}))$

$C = (0, \frac{1+\sqrt{5}}{2})$

$D = (-\cos\frac{\pi}{10}, \sin\frac{\pi}{10})$

$E = A + r_1(\cos(\theta_1 + \frac{\pi}{10}), \sin(\theta_1 + \frac{\pi}{10}))$

$F = B + r_2(\theta_2 + \cos(\frac{7\pi}{10}), \sin(\theta_2 + \frac{7\pi}{10}))$

$G = C + r_1(\cos(\theta_1 - \frac{\pi}{10}), \sin(\theta_1 - \frac{\pi}{10})$

$H = \mathrm{rotate}(C, -\frac{2\pi}{5}, H)$

$I = \mathrm{rotate}(C, -\frac{2\pi}{5}, G)$

$J = \mathrm{rotate}(A, -\frac{4\pi}{5}, G)$

$K = \mathrm{rotate}(A, -\frac{4\pi}{5}, F)$

$A' = (0, 0)$

$B' = (-\cos\frac{\pi}{10}, \sin\frac{\pi}{10})$

$C' = (0, -1)$

$D' = (\cos(\frac{\pi}{10}), \sin(\frac{\pi}{10}))$

$E' = A' + r_2(\cos(\theta_2 + \frac{\pi}{2}), \sin(\theta_2 + \frac{\pi}{2}))$

$F' = B' + r_1(\cos(\theta_1 - \frac{\pi}{10}), \sin(\theta_1 - \frac{\pi}{10}))$

$G' = C' + r_2(\cos(\theta_2 + \frac{7\pi}{10}), \sin(\theta_2 + \frac{7\pi}{10}))$

$H' = \mathrm{rotate}(C', -\frac{2\pi}{5}, G')$

$I' = \mathrm{rotate}(C', -\frac{2\pi}{5}, F')$

$J' = \mathrm{rotate}(A', -\frac{4\pi}{5}, F')$

$K' = \mathrm{rotate}(A', -\frac{4\pi}{5}, E')$
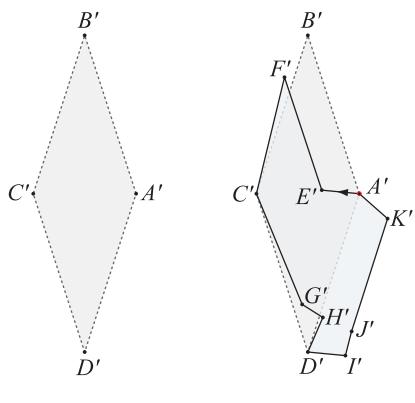
Figure 4.22 A tiling vertex parameterization for generalized Penrose kites and darts, controlled by four real-valued parameters r_1, θ_1, r_2, and θ_2. The vertices are enumerated in counterclockwise order starting at A for the kite and A' for the dart. The function $\mathrm{rotate}(p, \theta, q)$ rotates point q by angle θ about point p.

$A = (\sin \frac{\pi}{5}, 0)$
$B = (0, \cos \frac{\pi}{5})$
$C = (-\sin \frac{\pi}{5}, 0)$
$D = (0, -\cos \frac{\pi}{5})$

$E = A + r_1(\cos(\frac{7\pi}{10} - \theta_1), \sin(\frac{7\pi}{10} - \theta_1))$
$F = A + r_2(\cos(\frac{7\pi}{10} - \theta_2), \sin(\frac{7\pi}{10} - \theta_2))$
$G = \text{rotate}(B, -\frac{2\pi}{5}, F)$
$H = \text{rotate}(B, -\frac{2\pi}{5}, E)$
$I = D + r_2(\cos(\frac{\pi}{2} - \theta_2), \sin(\frac{\pi}{2} - \theta_2))$
$J = D + r_1(\cos(\frac{\pi}{2} - \theta_1), \sin(\frac{\pi}{2} - \theta_1))$
$K = \text{rotate}(D, -\frac{2\pi}{5}, J)$
$L = \text{rotate}(D, -\frac{2\pi}{5}, I)$
$M = \text{rotate}(D, -\frac{2\pi}{5}, H)$

$A' = (\sin \frac{\pi}{10}, 0)$
$B' = (0, \cos \frac{\pi}{10})$
$C' = (-\sin \frac{\pi}{10}, 0)$
$D' = (0, -\cos \frac{\pi}{10})$

$E' = A' + r_1(\cos(\frac{6\pi}{5} - \theta_1), \sin(\frac{6\pi}{5} - \theta_1))$
$F' = B' + r_2(\cos(\frac{7\pi}{5} - \theta_2), \sin(\frac{7\pi}{5} - \theta_2))$
$G' = \text{rotate}(C', -\frac{4\pi}{5}, F')$
$H' = D' + r_1(\cos(\frac{3\pi}{5} - \theta_1), \sin(\frac{3\pi}{5} - \theta_1))$
$I' = D' + r_1(\cos(\frac{\pi}{5} - \theta_1), \sin(\frac{\pi}{5} - \theta_1))$
$J' = \text{rotate}(A', \frac{4\pi}{5}, F')$
$K' = \text{rotate}(A', \frac{4\pi}{5}, E')$

Figure 4.23 A tiling vertex parameterization for generalized Penrose rhombs, controlled by four real-valued parameters r_1, θ_1, r_2, and θ_2. The vertices are enumerated in counter-clockwise order starting at A for the thick rhomb and A' for the thin rhomb.

Figure 4.24 Edge labels for the tiling edges of the two sets of Penrose tiles, in the spirit of the incidence symbols used for the isohedral tilings. The kite and dart are shown on the left, and the two rhombs on the right. (The edge labels are not related between the two sets.) Pairs of labels correspond as described in the text.

the tile boundaries are made to overlap, the tiling is indistinguishable from the one created with the overlapping edges excised. The problem is that although tiles with degenerate regions are visually indistinguishable from those without, they are very different from the point of view of the shape metric. The degenerate regions still occupy part of the tile's arclength, and become quite visible when the tile shape is converted into a turning function. We choose to solve this problem at the level of the original tiles, rather than attempting to extend the shape metric with heuristics to detect and discount such regions.

Now that we have grown our original set of tiling vertices to this new set of quasivertices, some additional work needs to be done to give new matching conditions on the edges that join quasivertices. Taking inspiration from the use of incidence symbols in isohedral tilings [68, Section 6.2], the possible edge shapes can be specified by labeling the edges around each tile and indicating adjacency rules for the labels. The edges of the kite and dart can be labeled $abcdaefghd$ and $ghefgcdabf$ respectively, where the enumerations start at the edges marked with arrows in Figure 4.22. To enforce matching between adjacent tiles, we require that the pairs (a, d), (b, h), (c, e), and (f, g) interlock. In effect, a given kite and dart will have only four non-congruent edge shapes between them. Similarly, we can label the edges of the thick and thin rhombs respectively as $abcdefegabhg$ and $afcdegabhg$, with the requirement that pairs (a, g), (b, e), (c, d), and (f, h) interlock. These labelings of the edges of the Penrose tiles are shown in Figure 4.24.
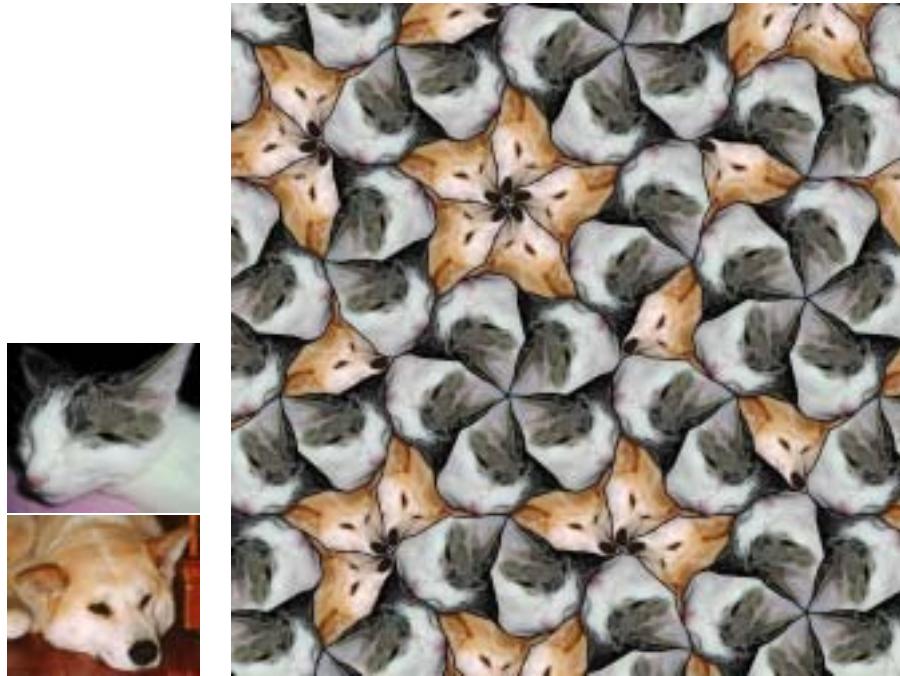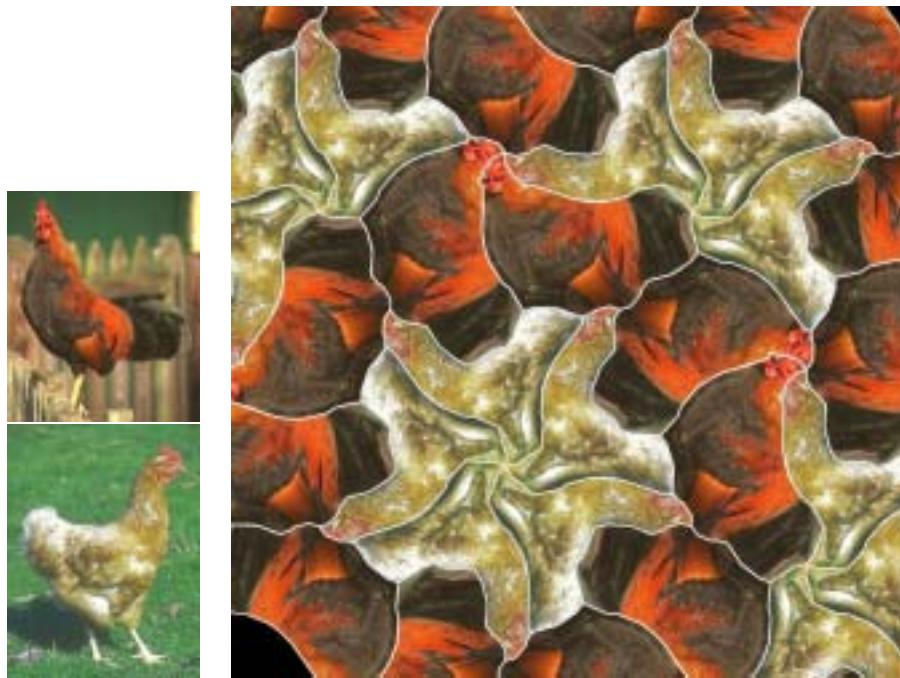
The edge shapes, combined with the four parameters controlling the tiling vertices, yield a space suitable for Penrose Escherization. As with the split isohedral case, I optimize over this space, attempting to minimize the maximum of the pairwise shape comparisons. Note that this parameterization cannot in general represent both a particular pair of tiles shapes and the reflections of those shapes, and that in each of the Penrose sets $P2$ and $P3$ the two prototiles are fundamentally different shapes. For these reasons, given two goal shapes S_1 and S_2 and their reflections S_1' and S_2', we must optimize for the eight combinations $\{(S_j, S_k), (S_j', S_k), (S_j, S_k'), (S_j', S_k')\}_{j,k \in \{1,2\}, j \neq k}$ (where S' is a reflection of S, as in Section 4.6.1).

Rendered drawings based on Penrose tilings are given in Figure 4.25 and Figure 4.26. In general, it is much more difficult to discover satisfactory Escherizations using Penrose tilings. The range of possible tile shapes is limited and somewhat peculiar, always having many sharp angles. More obviously, the number of Penrose "tiling types" is much smaller than with the isohedral tilings: we no longer have the luxury of hunting over many quite different types for one that happens to be particularly well suited to a given pair of goal shapes. The results are therefore less successful than in the isohedral and 2-isohedral cases, but interesting nevertheless for their connection to the interaction (both mathematical and personal) between Escher and Penrose. On the other hand, the interactive editor for Penrose tiles still allows profitable forward exploration of the space of tilings. Figure 4.26(b) shows an example of a tiling that was not Escherized but developed from scratch in a few minutes, and then decorated in a cartoon style. As always, Escherization is but one possible step in the design of decorative tilings. The remaining steps, divorced from the Escherization algorithm, may still be used to create attractive tesselations.
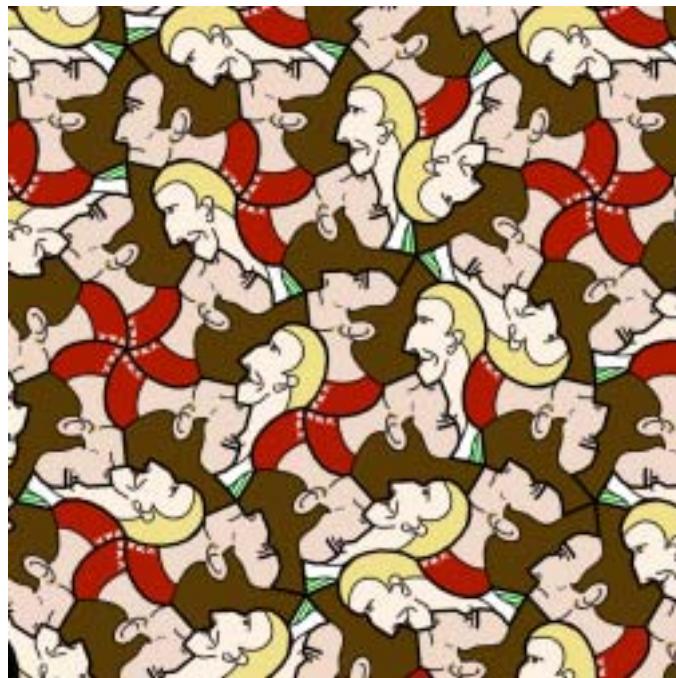
4.7   Non-Euclidean Escherization

Our search for methods to create Escher-like tesselations brings us finally to non-Euclidean geometry.

One of Escher's personal artistic quests was the representation of infinity. The infinite manifested itself in many ways in his art — as the infinity of three dimensional perspective in *Depth* and *Cubic Space Division*, as the infinity of endless spirals in *Path of Life* and *Sphere Surface with Fish*, and as the implied infinity of strange loops in *Ascending and Descending* and *Waterfall*.

(a) *Dogs and Cats Living Together* (P2)

(b) *Busby Berkeley Chickens* (P2)

Figure 4.25 Examples of dihedral Escherization using Penrose kites and darts. Because of the more limited search space, kites and darts enjoy fewer successful Escherizations than split isohedral tilings.

(a) *A Walk in the Park* (P3)

(b) *The Pentalateral Commission* (P3)

Figure 4.26 Examples of ornamental tesselations based on Penrose rhombs. The top image was Escherized in the usual sense; the bottom one was developed from scratch in a few minutes using an interactive editor.

It was in his regular divisions of the plane that Escher searched the hardest. There, the artistic infinity might find expression in a mathematical infinity, and the totality of an imagined world might be realized on a single piece of paper.

Certainly, his notebook drawings offer the suggestion of infinity, though only in a theoretical sense. We can imagine the tiles of a tiling being extended out in all directions but any drawing of the tiling can only contain a small patch. Escher worked hard at surmounting this obstacle, devising a number of clever systems of tiles that interlocked with scaled-down versions of themselves, creating an ever-diminishing pattern that crept toward a limit. Often, that limit was the center of a circle (as seen for example in *Smaller and Smaller*). Escher found these designs unsatisfying because there was no natural boundary to the drawing. New tiles could always be added to the outer edge, implying that infinity had still not been reined in. Truly capturing infinity in a single drawing would require a limit along the drawing's *outer* edge; that way, an entire universe of tiles would be completely contained in a finite region with no logical room for additions [49, Page 41].

Escher finally achieved his goal when he learned of the work of Coxeter. Coxeter's books on geometry contain visualizations of hyperbolic symmetry groups in the Poincaré model, visualizations that were precisely the kinds of tilings Escher was searching for. With Coxeter's guidance, he managed to produce a small number of remarkable *Circle Limit* prints. Each one is a testament not only to Escher's artistry and intuition, but also to his tenacity — he tirelessly pushed the design as close as possible to the bounding circle, down to the physical limits of his tools, his medium, and his body.

Escher also created several lovely spherical interpretations of his tesselations. They are beautifully carved bas-relief wooden sculptures, sometimes stained in multiple shades. These spheres are a clear expression of boundlessness, an intellectual and aesthetic cousin of infinity.

I would of course like to create Escher-like tesselations in the hyperbolic plane and on the sphere. There are two approaches that might be taken to accomplish this goal. We could rewrite the Escherization algorithm to operate directly in hyperbolic or spherical geometry (or, as was done in Chapter 3, perhaps we could write a single algorithm in absolute geometry). Alternatively, we could adopt the approach that Dunham uses to great effect [40, 43], taking existing tilings and adapting them to other geometries (or other symmetry groups in the same geometry).

The first option, though theoretically possible, is infeasible in practice. This section will ul-

timately present tools based on the second option, but beforehand some discussion is warranted concerning the challenges involved in expressing the Escherization algorithm in non-Euclidean geometry.

As always, the natural approach to an Escherization algorithm is to view it as an optimization process and to provide a suitable parameterized space and evaluation function.

The definition of an isohedral tiling carries over into non-Euclidean geometry, and so our first step would be to parameterize the space of non-Euclidean isohedral tilings. The combinatorial structure of these tilings is well understood. Grünbaum and Shephard give a classification of the spherical isohedral tilings in the same spirit as their Euclidean classification [67]. Delaney symbols can be used to enumerate the hyperbolic types [86]. Although there are infinitely many isohedral tiling types in the hyperbolic plane, it would be possible to enumerate a manageable set of them that are suitable for Escherization.

Deriving the tiling vertex parameterizations would be the next challenge. As before, for each tiling type we can write down a set of constraints that must be satisfied by the angles and edges of the tiling polygon. Unfortunately, in non-Euclidean geometry those constraints will not have solutions that are linear in some real-valued parameters. The tiling vertex parameterizations will therefore be more complex.

The main problem would be in adapting the shape comparison metric. "Shape" does not mean the same thing in Euclidean and non-Euclidean geometry. Arkin's metric takes for granted the fact that shape is to be considered independent of a polygon's absolute size. But in the hyperbolic plane and on the sphere, size is intimitely tied to shape. A triangle with a given set of interior angles will necessarily have a uniquely-determined size. Factoring out scale does not make sense outside of the Euclidean plane (rotation, too, would present difficulties). New shape metrics would have to be developed for the hyperbolic plane and on the sphere, metrics that do not require ambient space to be affine.

We therefore take Dunham's approach, and look for a way to re-map a preexisting tiling into another geometry. The technique and results presented here are similar to his and are included as a proof of concept, in the interest of completeness. One difference from Dunham's method is that I demonstrate how to render hyperbolic designs from *images*, yielding non-Euclidean image-based results. Dunham works only from vector-based art. I also demonstrate a technique for mapping

some Euclidean tilings to the sphere.
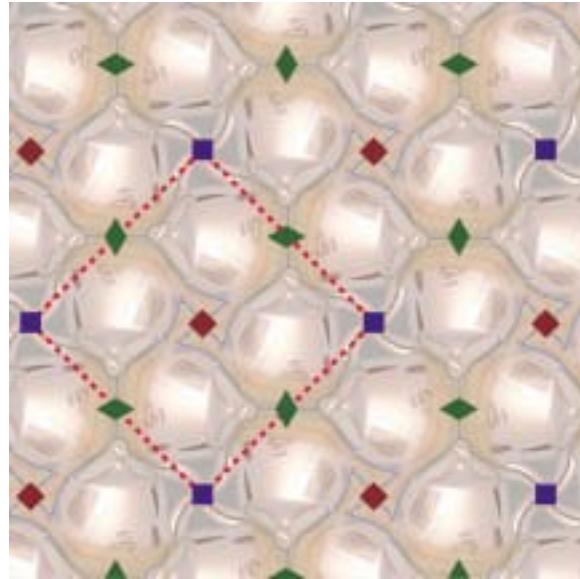
The translation into non-Euclidean geometry begins with the identification of a region of a Euclidean tiling called a "swatch," which is suitable for adaptation. Typically, the swatch will be a square or an equilateral triangle (the examples in this section use a square).

Then, a transformation maps the swatch into a region of the Poincaré model of the hyperbolic plane that plays an equivalent role in a hyperbolic symmetry group. If the mapping is well-behaved, then the transformed copy of the original region can be replicated to cover the hyperbolic plane, elaborating a tiling similar to the Euclidean source. An analogous process maps Euclidean swatches onto the surface of a sphere.

Not every Euclidean tiling can be adapted in this way, and not every non-Euclidean symmetry group can serve as the target of translation for some given source region. The periodicity of the isohedral tiling is a Euclidean property, defined in terms of translational symmetries. A translation to non-Euclidean geometry must be made based on properties of tilings that do not depend on translation.

We do have one well-known family of symmetry groups at our disposal that span Euclidean and non-Euclidean geometry. These are the groups of the form $[p, q]$ and $[p, q]^+$ (the latter were mentioned in Section 3.7.3). We have already seen in the case of Islamic star patterns how small changes to p or q can create a family of related designs in different geometries: see Figure 3.26 for examples. It seems reasonable therefore that Euclidean tilings with symmetry groups $[4, 4]$, $[4, 4]^+$, $[6, 3]$, and $[6, 3]^+$ could be adapted to produce closely related non-Euclidean interpretations. In the usual international notation for wallpaper groups, these groups are known respectively as $p4m$, $p4$, $p6m$, and $p6$. We can look for isohedral tiling types with these symmetry groups as sources for experimentation. Groups $p4m$ and $p6m$ are only associated with tiling types that make for uninteresting Escher tilings. The remaining isohedral types worth considering in this approach are IH11, IH21, IH28, IH31, IH34, IH39, IH55, IH61, IH62, IH79, IH88, and IH90. Of course, other types can be given non-Euclidean analogues, as Dunham consistently demonstrates. The types above are those that apply directly in the simple method presented here.

For the moment, we will ignore the problem of colouring, and focus on translating an uncoloured tiling. Let us consider the tiling called "Tea-sselation," shown in Figure 4.15(e). It is of isohedral type IH28, which has symmetry group $[4, 4]^+$ as discussed above. In this symmetry group, centers
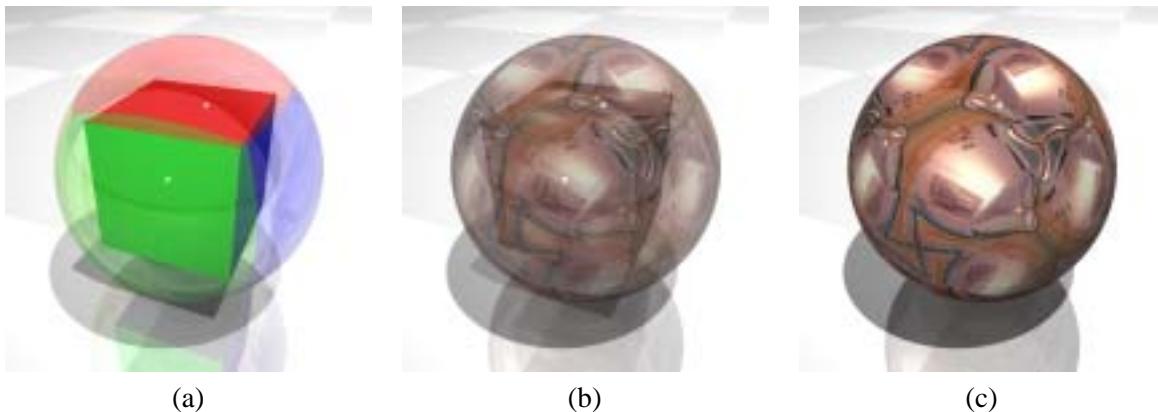
Figure 4.27 A visualization of the symmetry group for *Tea-sselation*. Red and blue squares indicate two inequivalent families of fourfold rotational symmetries. Green lozanges indicate twofold rotational symmetries. The highlighted square can be extracted from the tiling and used to construct related designs in non-Euclidean geometry.
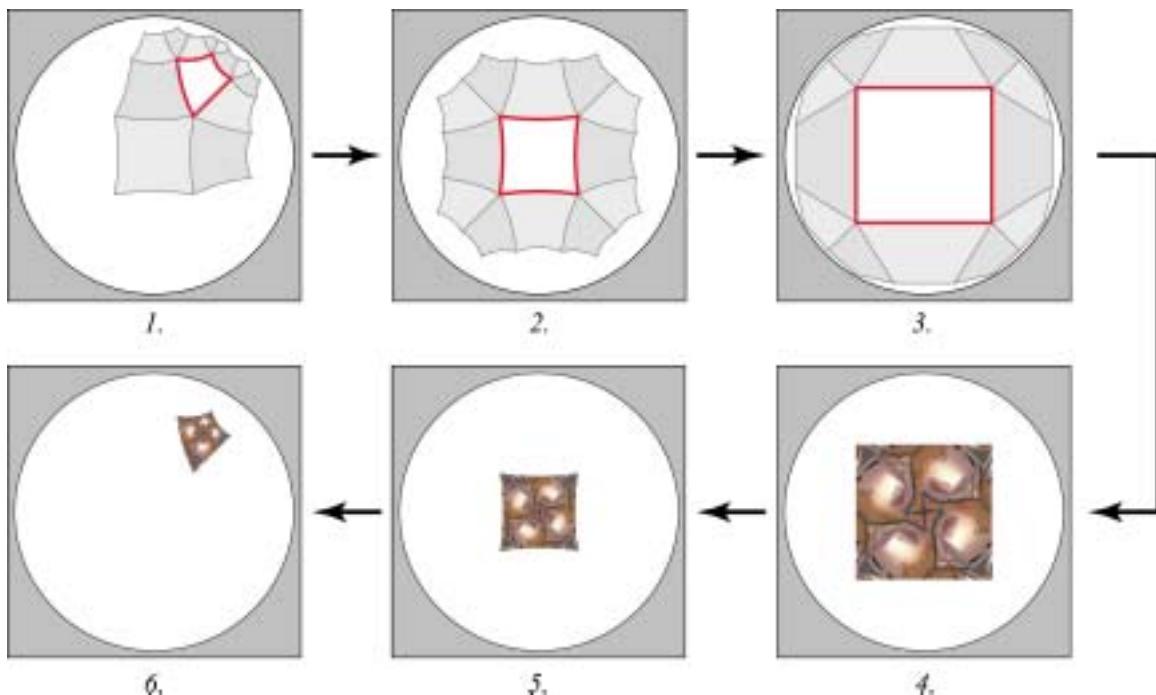
of fourfold rotation come in two inequivalent families. By joining up one set of centers into square units as shown in Figure 4.27, we can think of any pattern with $p4$ symmetry as being expressed by a collection of square swatches. In "Tea-sselation," there are two ways to build this region: we can join together the meeting places of the handles, or of the spouts. I choose the former. By varying q, we can translate the teapots to hyperbolic tilings with symmetry groups $[4, 5]^+$ and $[4, 6]^+$ (and, more generally, any $[4, q]^+$ for $q > 4$). We can also translate the tiling to $[4, 3]^+$ (the symmetry group of the cube).

It is relatively easy to map the tiling onto a sphere, using the symmetry group of the cube. Given a point on the sphere, we project that point onto the surface of a concentric cube. Each face of the cube is square and can be covered with a copy of the swatch. When those pixels are mapped back onto the spherical surface, the result is a spherical interpretation of the original tiling. This mapping process is illustrated in Figure 4.28.
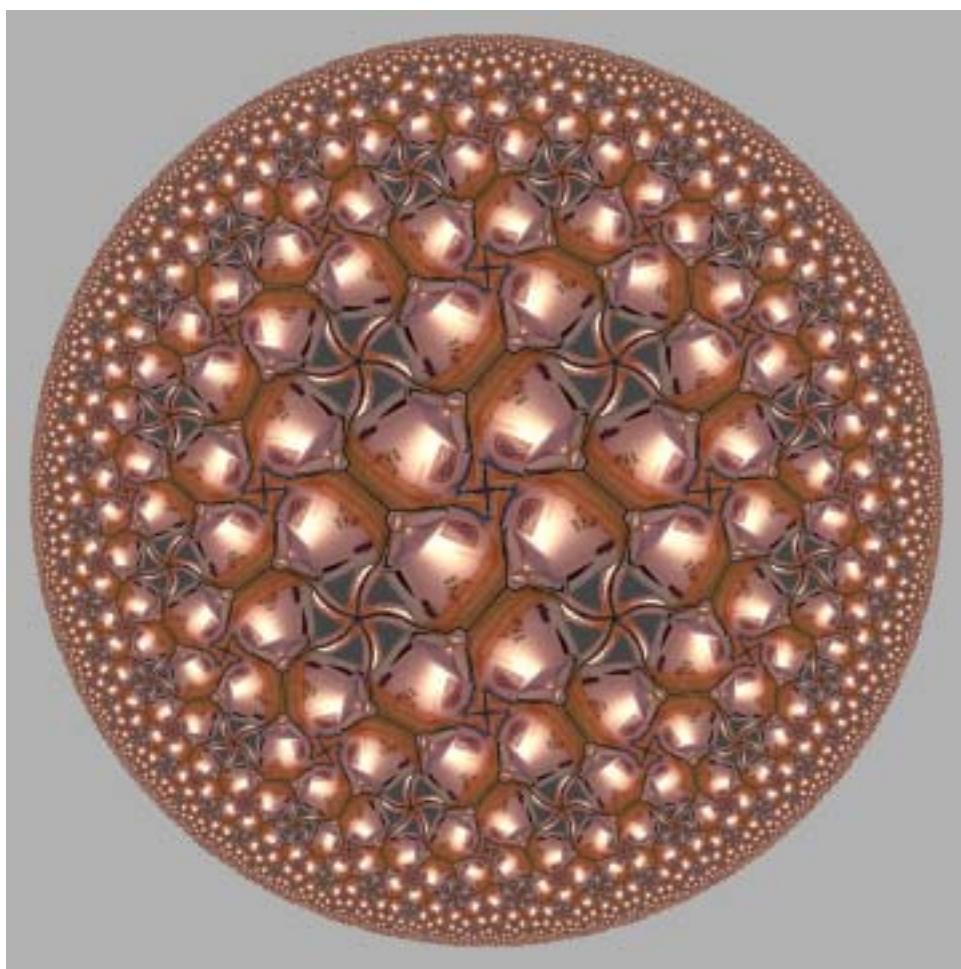
For the hyperbolic plane, we can transfer the design to symmetry group $[4, q]$ for any $q > 4$. We

(a)   (b)   (c)

Figure 4.28 A visualization of how a texture is mapped onto the surface of a sphere via the faces of a cube. In (a), we see how each face of the cube relates to a portion of the sphere. The same mapping is repeated in (b), but with the texture mapped to the cube faces. The resulting spherical tiling is shown in (c).

Figure 4.29 Six steps showing how a square Euclidean texture can be warped to fit a given equiangular quadrilateral in the hyperbolic plane. The first step shows the quadrilateral. We use a hyperbolic rigid motion to move it to the origin in step 2, and reproject into the Klein model in step 3. In the Klein model, the quadrilateral is drawn as a square, allowing the texture to be mapped into it easily in step 4. We then undo the reprojection and rigid motion in steps 5 and 6, yielding the final mapped quadrilateral.

Figure 4.30 An uncoloured interpretation of "Tea-sselation" mapped into the hyperbolic plane with symmetry group $[4, 5]^+$.
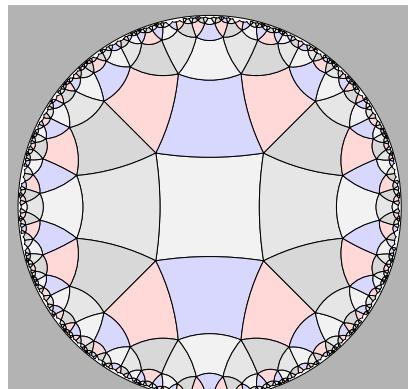
tile the hyperbolic plane with the tiling $\{4, q\}$. Each tile is a regular 4-gon. For each tile, we can rigidly move the tile until it lies centered at the origin of the unit disk in the Poincaré projection. We then switch to the Klein projection, where the regular 4-gon must get mapped to a square. This square can receive the original swatch, and we can then undo the transformations to move the pixels back up to the original 4-gon. The sequence of transformations is visualized in Figure 4.29. An interpretation of "Tea-sselation" in the hyperbolic symmetry group $[4, 5]^+$ is shown in Figure 4.30.

It is also possible to add colour to these designs. The notion of perfect colouring works fine in non-Euclidean geometry. I create coloured designs by constructing multiple swatches that express all the colour combinations I wish to use in the non-Euclidean interpretation, and assembling those swatches according to a selection rule that embodies a perfect colouring. For spherical designs in $[4, 3]^+$, it is easy enough to place the six swatches manually; an example using teapots is shown in Figure 4.32. In the hyperbolic plane, I implement colouring as described by Dunham [39]. He extends the data structure representing a rigid motion to include a permutation of the set of colours. Each primitive rigid motion (rotations around the p-centers and q-centers) get initial permutations. As rigid motions are composed, their associated permutations are composed as well. The colour of every tile in the resulting tiling can be read out as the first element of the transformed permutation. Figure 4.31 gives a perfectly 5-coloured hyperbolic teapot tiling in $[4, 5]^+$. Moving to $[4, 6]^+$ allows for a simpler perfect colouring with only three colours, as shown in Figure 4.33.
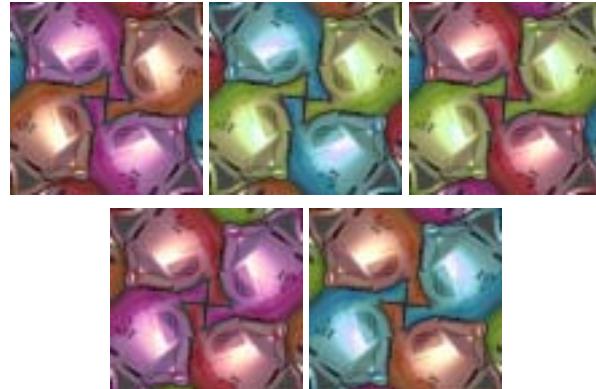
Other tilings can serve as a source for translation to non-Euclidean geometry. As a final example, I turn to Escher's own notebooks. His well known symmetry drawing number 104 features black and white lizards in a tiling of isohedral type IH55. Once again, we can translate this design from the Euclidean $[4, 4]^+$ to the hyperbolic $[4, 6]+$ to simulate a new member of Escher's *Circle Limit* series. The resulting design, "Circle Limit V," is shown in Figure 4.34. Because the original tiling is of topological type 4^4, the hyperbolic interpretation can be perfectly coloured using only two colours.

4.8 Discussion and future work
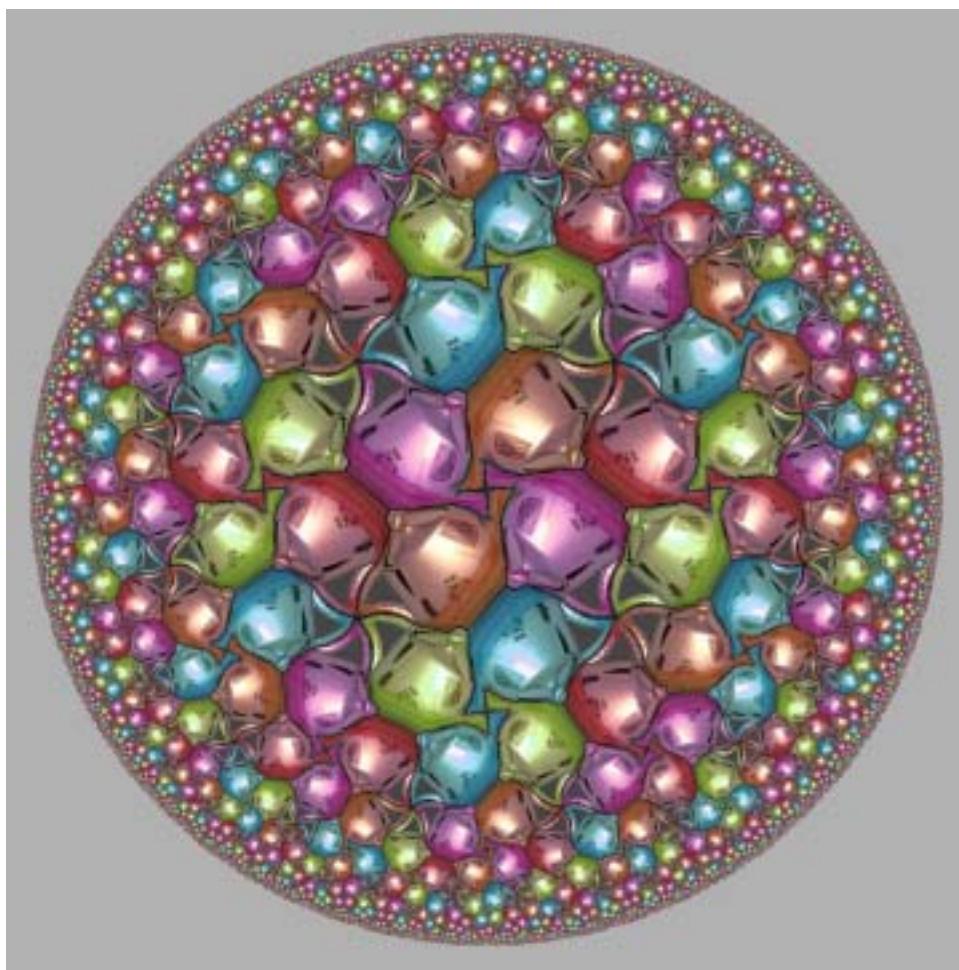
Most polygons are not tiles. For just about any goal shape, an Escherizer will have to produce an approximation, and a better Escherizer will produce a closer approximation. When the shape metric
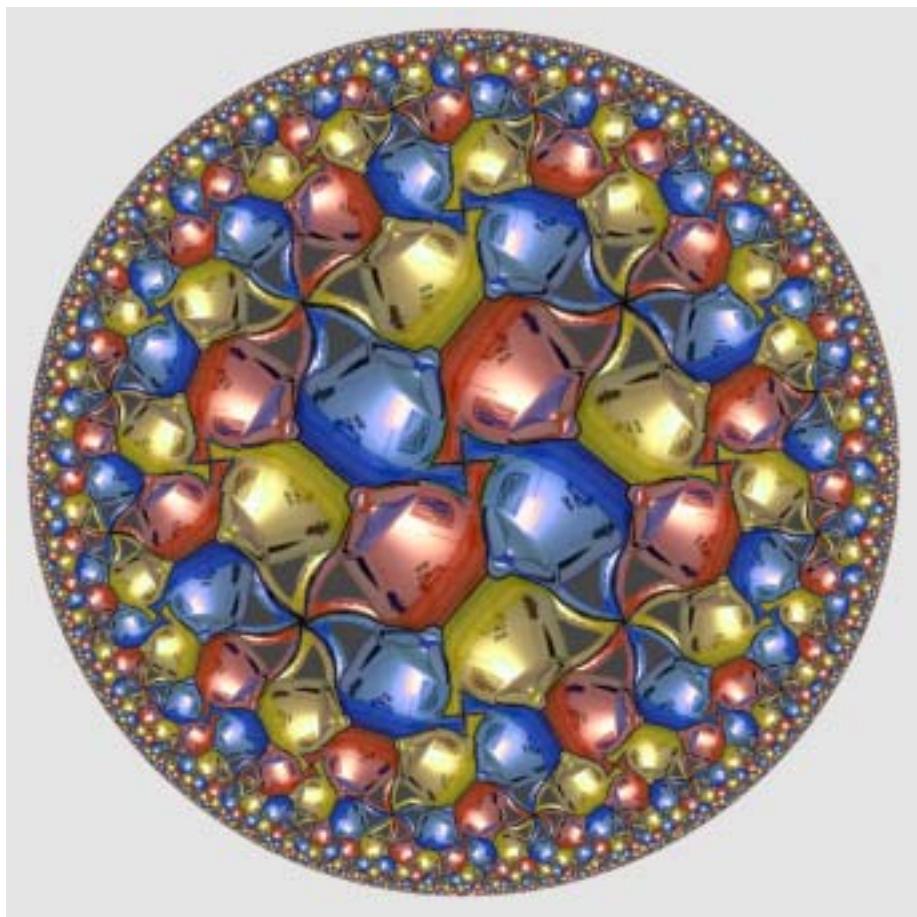
(a)   (b)

Figure 4.31 A hyperbolic interpretation of *Tea-sselation* using symmetry group $[4, 5]^+$. This realization contains almost seven thousand tiles. The tiling is perfectly coloured using five colours, mapped in pairs onto the five swatches shown in (b). The choice of swatch is dictated by the colouring shown in (a).
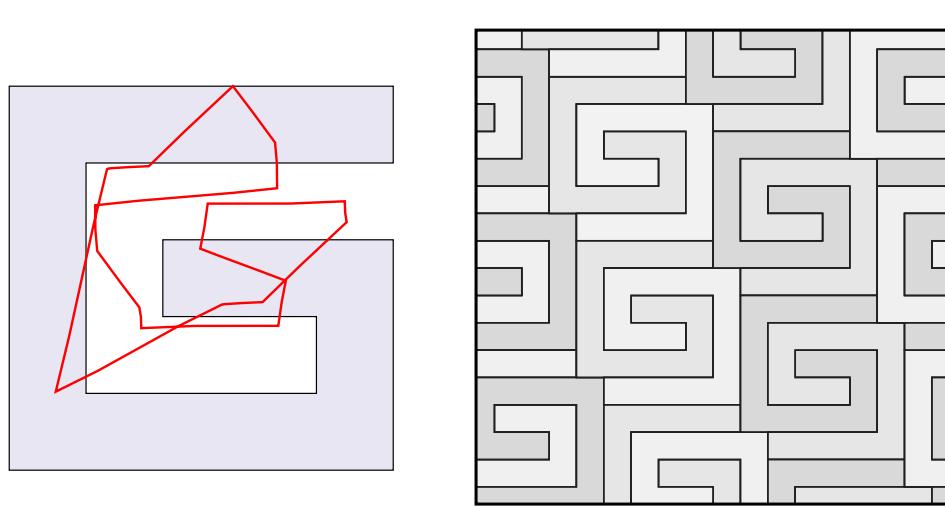
Figure 4.32 Teapots mapped onto a sphere, together with spheres mapped onto a teapot. The sphere features a perfectly 3-coloured teapot tiling in the symmetry group $[3,4]^+$.

Figure 4.33 Teapots in the hyperbolic symmetry group $[4,6]^+$. This symmetry group permits a perfect colouring of the tiling using only three colours.

Figure 4.34 A simulated successor to Escher's *Circle Limit* drawings, constructed by translating Escher's symmetry drawing number 104 into the hyperbolic symmetry group $[4,6]^+$. *M.C. Escher's symmetry drawing number 104 ©2000 Cordon Art B.V. – Baarn – Holland. All rights reserved.*

Figure 4.35 A goal shape for which Escherization performs badly, and a tiling that it admits.

is used as the measure of closeness, a perfect Escherizer would determine the smallest distance over all possible tile shapes, and return the tiling that achieves that bound. Our imperfect optimizer, by contrast, coarsely samples the space of isohedral tilings in a directed fashion and returns the best sample it finds. Consequently there are seemingly easy cases, such as the one in Figure 4.35, that our algorithm cannot successfully Escherize.

In practice, our Escherization system performs well on convex or nearly convex shapes. The shapes that tend to fail are the ones with long, complicated edges between the tiling vertices. It is difficult for the optimizer to come up with just right the sequence of vertex adjustments to push a tendril of detail out, especially when constrained by the "no non-uniform noise" condition of the metric. A shape metric for Escherization should not be sensitive to local deformations. As a tendril of detail is pushed out, its contribution to the overall perimeter of the tile shape compresses the rest of the tile's turning function horizontally, potentially worsening the overall comparison. The metric of Arkin *et al.* seems to resist these local changes in detail.

Furthermore, in the shape comparison metric used, the importance of a section of outline is directly proportional to its length as a fraction of the perimeter of the goal shape, even if from an aesthetic point of view outlines may obey different measures of significance. For example, the precise profile edge of a face in silhouette, descending along eyes, nose, and mouth, is much more important than the hairline. But to the current shape metric these might be relatively insignificant details. It would be valuable to investigate an extension to the polygon comparison metric wherein a section of outline could be assigned a measure of importance, a weight controlling which parts of the polygon should match more closely.

Although Escher's tiles are almost always immediately recognizable as particular kinds of animals, they generally bear little actual resemblance to a real image: they are more like conventionalizations, or cartoons. The results presented here inherit a distinctive three-dimensionality from the goal images they capture. My algorithm does not "understand" the shapes it is manipulating, so it has no way to deform them while preserving their essential recognizability. It must instead rely on a purely geometric notion of proximity. A powerful extension to the algorithm would be to add degrees of freedom that control the goal shape as well as the tile shape. In a lizard outline, for instance, these new parameters might allow the legs, head, and tail to bend. The optimization could proceed as before, but now instead of having only the tile evolve to approximate the goal, we would

have both shapes work towards a common meeting point. A very coarse version of this enhancement would be to assemble a collection of related goal shapes, any of which satisfies the artist's intent, and optimize over all of them simultaneously. In the dihedral case, we might for example have a set of bird outlines and a set of fish outlines, and attempt dihedral Escherization on all pairs of a bird and a fish.

Another way to add degrees of freedom to the goal shape would be to begin with a three dimensional model as a goal, and include rotation parameters in the optimization. The model is converted into a polygon by rotating it and taking the silhouette. Some care would be needed here to keep the model from rotating into a position that tiles very well by hiding all of its interesting geometry inside the silhouette. For additional guidance, we might turn to Escher's writing. He points out [49, Page 106] that certain living forms are most easily recognized when viewed from certain characteristic angles: "Four-footed mammals are usually best recognized by looking at them from the side. Reptiles and insects, on the other hand, generally present their most typical aspect when seen from above, while the human figure is at its most characteristic when viewed from the front."

This chapter has presented Escherization algorithms for a few different families of tilings: the isohedral, split isohedral, and Heaven and Hell tilings, and tilings from Penrose sets $P2$ and $P3$. There are always more families of tilings to explore, allowing us to slowly move beyond the mathematical structures Escher worked with. The logical extension of the work with transitive tilings would be to run Escherization on tilings encoded by Delaney symbols, in which case the user might in theory supply any number of goal shapes and have the system discover a k-isohedral tiling that approximates all of them. The number of combinatorial types to be searched grows very quickly with k, but perhaps some small subset of those types could be determined automatically ahead of time. As a benchmark for k-isohedral Escherization, we might take Escher's symmetry drawing 71, a remarkable periodic tesselation featuring twelve distinct bird motifs.

Escher's final symmetry drawing, number 137, hints at an extension to k-anisohedral prototiles (recall that a k-anisohedral prototile is a shape that can only form monohedral tilings with at least k tile transitivity classes). The k-anisohedral tilings are still not very well understood, although Berglund has begun a classification of the case $k = 2$ [9]. The desire to create Escherized anisohedral shapes might help further motivate (and even aid) the search for anisohedral tiles.

Finally, we can consider aperiodic tiling types other than Penrose's. Perhaps by parameterizing

the shapes of other well-known tile sets such as those of Ammann [68, Section 10.4], we might discover candidates for Escherization that can produce more successful tile shapes.

Escherization is merely one powerful tool to be applied to the creation of ornamental tilings. It is no substitute for the mathematical insight or artistic know-how of Escher himself, but it can help the designer of a tiling to take a large step towards a solution. The artist still plays an important role in the process and must harness the power of Escherization to bend it to their aesthetic goals. In creating art by computer, Escherization is simply one means of communication in the dialogue between human and machine.

Chapter 5

CONCLUSIONS AND FUTURE WORK

This dissertation has aimed to explore the possibility of using modern mathematics and computer science to develop new ways of analyzing and synthesizing ornament. Although I chose to focus specifically on Islamic star patterns and Escher's tilings, the underlying goal was to develop general ideas and principles that might then be applied to other ornamental styles. The range of possible styles to investigate is huge, as evidenced by a quick look through a treasure trove like *The Grammar of Ornament* [91]. We should also consider moving on with new forms of ornament that are inseparably tied to the computer. What new vistas in ornamentation will be opened up by the increasing power of computer graphics?

What follows are some further ideas for future work that are not directly related to the work of Chapters 3 and 4.

5.1 Conventionalization

Conventionalization refers to the creation of stylized, iconic interpretations of natural forms, most commonly plants. There is a general trend in the ornamental traditions of many cultures for conventional representations to evolve over time. Wong *et al.* point out that one goal of conventionalization is to distill a naturalistic form down to an abstract essence, freed from the idiosyncracies of any specific instance of the form [139]. Another reason for simplified designs might simply be the practicality of working with certain materials.

Over time, a conventionalized form begins to take on a life of its own, forming its own visual language that evolves as it is passed through history from designer to designer. Gombrich, writing on "The Etymology of Motifs" [60, Page 180], discusses some examples of this progression, including conventionalized representations of the lotus and the acanthus. In another fascinating example, Christie shows how many medieval European frieze patterns evolved from conventionalization of

Arabic calligraphic inscriptions on earlier silk weavings [22, p. 20].

Conventionalization plays an important role in Islamic star patterns. A star pattern is not always depicted as an isolated arrangement of geometric forms. Frequently, the polygonal regions defined by the lines of a star pattern are filled not with solid colours, but with elegant floral designs that extend more or less to the boundary of the region. The wonderful drawings of Prisse d'Avennes, recently reprinted [31], illustrate many examples. The designs are closely associated with arabesques, another form of Islamic ornamentation. Typically, each distinct shape of region has a floral motif assigned to it, and the motif has the same symmetries as its surrounding region. It should be possible to extend the work of Wong *et al.* [139] to create symmetric ornament inside of symmetric boundaries, and to then use this extension to produce appropriate floral motifs for every distinct cell shape in a given star pattern.

Escher's tilings also rely heavily on conventionalization, this time of animal forms. Nobody would mistake the outlines in Escher's sketchbook for real-world animals. Each form is a cartoon, a highly stylized interpretation that nevertheless is highly suggestive of an animal. In some cases, conventionalization gives way to outright invention: shapes are decorated with suggestive eyes and appendages, but are not meant to depict any real animal. Escher uses stylization to great advantage. In freeing his cartoon animals from constraints of realism, he can distort them into shapes that tile the plane without sacrificing the "meaning" of the finished design. His success suggests another direction for future work in conventionalization. To aid the Escherization algorithm, we might turn real-world outlines into conventionalized representations endowed with new degrees of freedom. By varying the degrees of freedom, we might locate a version of the shape that lends itself more readily to tiling the plane.

The long-term goal in this direction would be to devise an algorithm that produces conventionalizations from real-world objects (images or three dimensional models) without any guidance. I believe we are still a long way from fully automatic conventionalization. Since abstraction relies on true understanding of the object being abstracted, an automated process would seem to require real machine intelligence. A feasible intermediate goal would be to investigate what sorts of high level tools might be uniquely qualified to aid the user in the process of conventionalization. Recently, Santella and DeCarlo demonstrated a system that evaluates salience in an image based on eye-tracking data from a human user [32]. They use the measure of salience to guide a painterly rendering

algorithm, but perhaps eye-tracking data could also be used to determine "geometric salience," a measure of the relevance of parts of an object's outline.

5.2 Dirty symmetry

Glassner argues that too much order can be just as unappealing as not enough [54]. A floor covered by a grid of square tiles is so featureless that perhaps it ought not be considered ornamental at all.

We have already seen ways to tamper with the global order of symmetry. Rigid motions are lost in translation from the hyperbolic plane to the Poincaré model, but the projected design still has order. Expanding the horizons of symmetry to include quasiperiodic tilings or fractals questions but ultimately reaffirms our perception of order.

The perfection of symmetry can be tampered with even more easily. For example, the entire plane can be passed through a displacement field based on procedural noise [44]. Motifs in a pattern are distorted slightly, obliterating every symmetry. However, when the magnitude of the displacement is not too great, we have no difficulty whatsoever seeing both the symmetric "essence" of the pattern and the concrete deviation from that essence. The new pattern has the best of both worlds, conveying an organic, loose appearance while hinting at a rigid underlying structure. Møller and Swaddle cite psychoaesthetic and sociological evidence that this sort of imperfect symmetry is preferred over perfection [113].

The preference is well explained by Gombrich's "sense of order" [60]. He adopts an outlook on perception that is very much in line with information theory. We perceive structure in the world by first forming a mental model that predicts perfect regularity, and then evaluating the perceived deviation from this model. We can theorize that a randomly displaced pattern stimulates and engages the perceptual system at both the model-forming and the deviation-measuring levels. Gombrich claims that when the mental model completely explains what is actually seen, the act of perception is made too easy and the result is boredom. On the other hand, a completely irregular pattern allows for no model; robbed of the ability to make sense of such a pattern, the result is confusion. As Gombrich says, delight lies somewhere in between these two extremes [60, Page 9].

It would be interesting to explore how this "dirty symmetry" affects aesthetic judgment of a pattern, in terms of the variety and magnitude of the distortion. Computers are ideally suited to the task,

allowing us to perform random distortions with ease. As Gombrich says, "Maybe the greatest novelty here is the ability of computers not only to follow any complex rule of organization but also to introduce an exactly calculated dose of randomness [60, Page 94]." We could use these "calculated doses of randomness" to measure experimentally the aesthetic response to varying amounts of distortion. In image processing, a small amount of colour noise can improve the qualitative appearance of an image. Can geometric noise be added to any design to increase its appeal?

5.3 *Snakes*
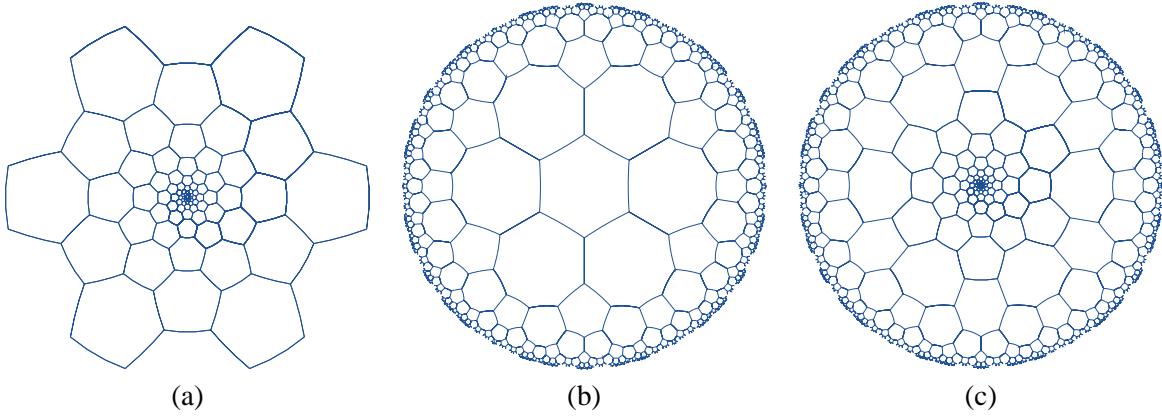
In Section 4.7, I discussed Escher's lifelong quest to capture and represent infinity. Through his own investigations, he discovered methods to make motifs diminish in size towards the center of a disk. Later he found in Coxeter's writings on non-Euclidean geometry the key to putting the limit on the outer edge of the disk, thus capturing an entire universe on a finite page.

In his final print, *Snakes*, Escher fuses these two ideas with astonishing mathematical ingenuity. *Snakes* features an arrangement of interlocking rings that diminish in size both at the center and at the outer edge of a disk. Here we have the infinities of both sorts of limiting patterns, still confined to a finite space with no possibility for further extension.

Very little has been written about the geometry of *Snakes*. Ernst [47, Page 110] acknowledges that the "Coxeter network" (a hyperbolic tesselation in the Poincaré model) describes the outer rings. Ernst's description is also valuable because it includes several of Escher's many preliminary sketches for this print; many more sketches can be found in a wonderful new catalog of Escher's work [105]. Rigby [122] goes into much more detail in describing the outer and inner rings.

In some of Escher's preliminary sketches, we see a network of triangles made by drawing line segments joining the centers of adjacent rings. Let us consider instead the dual of this network, which can be constructed by joining the points where triplets of rings pass by each other in a weave. This network will be a tiling where each tile corresponds to one ring. Rigby shows that in the outer portion of the design, the tiling is just the hyperbolic (6.8^2), which can be constructed by truncating the regular tiling (4^6).

Ernst does not attempt to describe the structure of the rings at the center of *Snakes*. Rigby describes the structure as a "radiating framework," but does not give details on how such a framework

Figure 5.1 A visualization of the geometric basis of Escher's *Snakes*. In the three tilings, the tiles represent rings from the original print. The tiling in (a) corresponds to the inner portion of the design, where rings diminish in size towards the center of a disk. The tiling in (b) is the hyperbolic tiling corresponding to the outer part of *Snakes*. In (c), the two are roughly combined in a manner consistent with Escher's design.

may be constructed. The answer can be found in Dixon's "antiMercator" transformation [37], which can be defined in the complex plane by $f(z) = e^z$. The antiMercator operation transforms an infinite horizontal strip of width 2π to the whole plane. Horizontal lines are mapped to lines radiating out of the origin, and vertical lines to circles. Other lines are mapped to equiangular spirals.

Given any periodic tiling with minimal translation vectors $\overrightarrow{T_1}$ and $\overrightarrow{T_2}$, and any integers a and b (not both zero), we can scale and rotate the tiling so that the vector $a\overrightarrow{T_1} + b\overrightarrow{T_2}$ is vertical with length 2π. The antiMercator transformation of the scaled and rotated tiling is an attractive pattern that tiles the plane with a single limit point (a place where infinitely many tiles meet) at the origin. In general, antiMercator transforms of periodic tilings resemble phyllotaxis patterns, with counterrotating spirals of tiles emanating from the origin. In the case of *Snakes*, the central arrangement of rings can now be viewed as the antiMercator transform of the regular tiling by hexagons with $a = 6$ and $b = 6$.

Of course, the trick lies not just in defining the outer hyperbolic and inner antiMercator tilings, but in stitching the two together in a hybrid. Here Escher's intuition guides him flawlessly. Rigby shows how Escher makes the two parts of the design link up by turning some of the circular rings into ovals. Examination of *Snakes* also reveals one set of heptagonal tiles (or rings that are linked

to seven others) in the stitched-together region of the design. For comparison, the antiMercator, hyperbolic, and stitched-together tilings are shown in Figure 5.1. Some of the tiles in the crossover region are more distorted than in Escher's design, but the topology is correct relative to the original print.

Is there a general theory that can produce these marvelous hybrid tilings? It seems as if many pairs of antiMercator and hyperbolic tilings might be fused together to serve as a basis for new doubly-infinite patterns. The challenge is to find a general theory for determining which pairs are compatible, and for stitching the tilings together when they are. Aside from linked rings, we can imagine using these tilings as a basis for other sorts of ornament, such as Escher tilings or Islamic star patterns. I find this future direction particularly appealing, since *Snakes* is seemingly the only artwork ever created with this geometry.

Snakes is just one example of a generalization of planar symmetry that is still highly regular. Many other opportunities await for exploring similar kinds of hybrid structures and the ornamental patterns that can be derived from them.

5.4 Deformations and metamorphoses

Many of Escher's prints feature divisions of the plane that change or evolve in some way. The most well-known is probably *Metamorphosis II*, a long narrow print containing a variety of ingenious transitions between patterns, tilings, and realistic scenery. Escher was quite explicit about the temporal aspect of these long prints. He would not simply describe the structure of *Metamorphosis II* – he would narrate it like a story [49, Page 48].

A survey of Escher's work (as collected by Bool *et al.* [15]) turns up sixteen pieces employing some kind of transition device. By studying these sixteen pieces, I have identified six categories of transition. *Metamorphosis II* serves as a kind of atlas, as it incorporates all six varieties. They are as follows:

T1. **Realization:** A geometric pattern is elaborated into a landscape or other concrete scene. In *Metamorphosis II*, a cube-like arrangement of rhombs evolves into a depiction of the Italian town of Atrani.

T2. **Interpolation:** A tiling evolves into another tiling by smoothly deforming the shapes of tiles. Escher used this device to change simple tilings into his familiar interlocking animal forms (for example, squares into reptiles in *Metamorphosis II*, and triangles into a variety of forms in *Verbum*).

T3. **Sky and Water:** Realistic shapes meet in a dihedral tiling. We have already encountered this sort of transition in Section 4.6.2, in the discussion of Sky and Water designs. This sort of transition starts with copies of some realistic shape A, ends in copies of another realistic shape B, and moves between them by passing through a dihedral tiling whose tiles resemble A and B. This device is used to produce single transitions, as in *Sky and Water*, and also as part of longer structures, as in *Metamorphosis II* (the transitions in the latter are rotated ninety degrees with respect to the former).

T4. **Abutment:** Two distinct tilings are abrubtly spliced together along a shared curve. The transition works when the two tilings have vaguely similar geometry and can be made to abut one another without too much distortion. Escher uses this device exactly once, to transition from hexagonal reptiles to square reptiles in *Metamorphosis II* (later, he embedded the same sequence into the larger *Metamorphosis III*).

T5. **Growth:** Motifs gradually grow to fill the negative space in a field of pre-existing motifs, resulting in a multihedral tiling. Often, after a Sky and Water transition, the result is a pattern of realistic motifs that do not tile. In several cases, Escher transitions back to tilings by growing another set of motifs into the empty spaces of the pattern. The new motifs need not occupy all the empty space; in *Metamorphosis II*, red birds grow to occupy half the space between black birds. When the two sets of motifs finally fit together, they leave behind a white area in the form of a third bird motif.
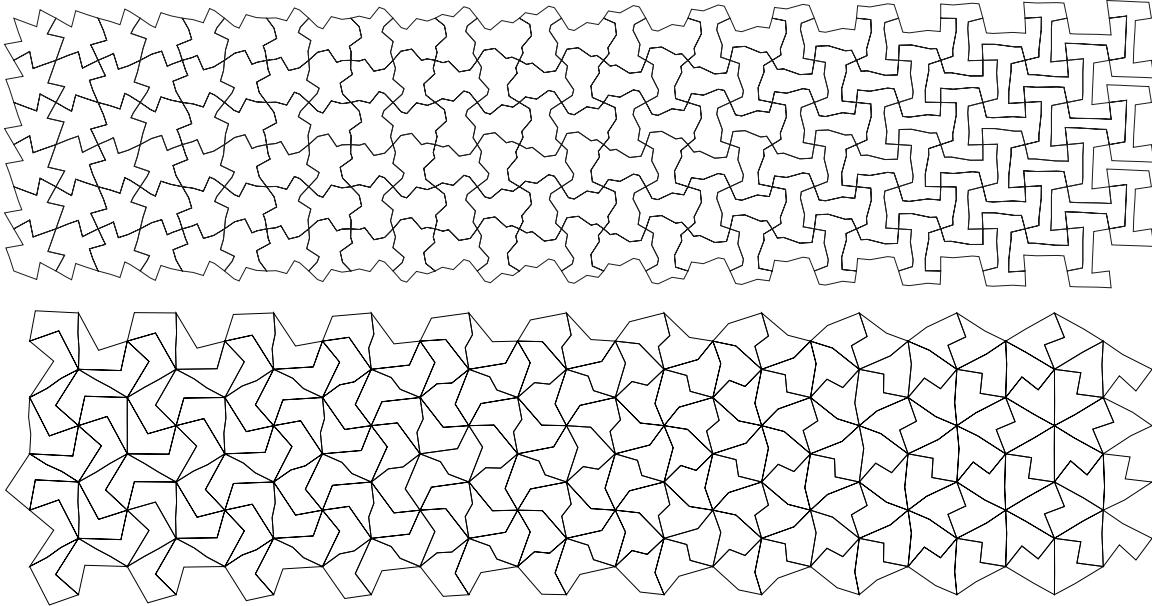
T6. **Crossfade:** Two designs with compatible symmetries are overlaid, with one fading into the other. Escher also applies this device sparingly, using it only to transition from a rectilinear arrangement of copies of the word "metamorphose" into a checkerboard (and later, to make the reverse transition).

Putting all these transition types together, the sequence of transitions in *Metamorphosis II* might then be read as

- **T6** (copies of "metamorphose" into a checkerboard)
- **T2** (a checkerboard into a square arrangement of reptiles)
- **T4** (square reptiles into hexagonal reptiles)
- **T2** (hexagonal reptiles into hexagons)
- **T1** (hexagons into a honeycomb with bees)
- **T3** (bees into fish)
- **T3** (fish into black birds)
- **T5** (black birds into birds of three different colours)
- **T2** (birds into a cube-like arrangement of rhombs)
- **T1** (rhombs into the town of Atrani, which then becomes a chessboard)
- **T1** (a chessboard into an orthographic checkerboard — an elaboration in reverse)
- **T6** (a checkerboard into copies of "metamorphose")

By far, the most important transition type in Escher's work is the Sky and Water device. We have seen how the Heaven and Hell Escherization algorithm of Section 4.6.2 might be applied to build Sky and Water designs. We might then consider that technique to be a first step in the creation of a "Metamorphosis toolkit," a system that would simplify the construction of images like Escher's. The other five transition types hold a collection of interesting challenges. Crossfade seems to be primarily a matter of registration and image processing. Growth seems approachable, though emulating Escher's use of growth would require a means of representing 3-isohedral tilings. Abutment is rather special, and would probably not be very widely applicable. Nevertheless, a program could examine paths that follow tiling edges in the two tilings, in search of two paths that resemble each other as much as possible. Elaboration seems like a fascinating but very difficult problem. Given a tiling and, say, an image, an elaboration algorithm would have to search the image for a region that could be expressed as a gradual geometric enrichment of the tiling. The image might also need to be distorted in a very specific way to accommodate the tiling.

Interpolation is a beautiful mathematical problem that deserves a more extended discussion. Given two tilings $\mathcal{T}_1$ and $\mathcal{T}_2$, we ask for a smooth geometric transition between the two tilings.

Figure 5.2 Examples of parquet deformations.

Presumably, a one-to-one correspondence is established between the tiles of $\mathcal{T}_1$ and $\mathcal{T}_2$, and as a parameter t moves from 0 to 1, each individual tile gradually deforms from its $\mathcal{T}_1$ shape to its $\mathcal{T}_2$ shape. The transition might be carried out spatially as in Escher's art, or even temporally as a smooth animation from $\mathcal{T}_1$ to $\mathcal{T}_2$.

As was mentioned in Section 3.4.1, the parquet deformations of William Huff are a kind of spatial animation. Huff was inspired directly by Escher's Metamorphoses. He distilled the style down to an abstract core, considering only interpolation transitions, and favouring abstract geometry rendered as simple line art to Escher's decorated animal forms. As reported by Hofstadter [83, Chapter 10], Huff decided further to focus on the case where $\mathcal{T}_1$ and $\mathcal{T}_2$ are "directly monohedral," in the sense that every tile is congruent to every other through translation and rotation only. We may also assume he had only periodic tilings in mind. Finally, he asked that in the intermediate stages of the deformation the tile shapes created could each be the prototile of a monohedral tiling (Hofstadter amends this rule, pointing out that some deformation might be necessary to make the intermediate shapes tile).

Inspired by parquet deformations and by Escher's interpolation transitions, we may pose the

related problem of finding a smooth transition between any pair of isohedral tilings. A solution to this problem might then be expanded to encompass Escher's work (by considering a k-isohedral extension) or parquet deformations (by introducing the restrictions mentioned above). In any case, the isohedral problem is sufficiently interesting, and the results sufficiently attractive, that it can be fruitfully studied in isolation.

Besides choosing between temporal and spatial transitions, there is a succession of increasingly difficult problems to solve, depending on the relationship between $\mathcal{T}_1$ and $\mathcal{T}_2$:

1. The two tilings are of the same isohedral type and have congruent arrangements of tiling vertices.

2. The two tilings are of different isohedral types and have congruent arrangements of tiling vertices.

3. The two tilings are of the same isohedral type.

4. The two tilings are of the same topological type.

5. The two tilings are isohedral.

The first two cases are trivial to solve. There is a rigid motion that maps the tiling vertices of $\mathcal{T}_2$ onto the tiling vertices of $\mathcal{T}_1$, and the registration afforded by this rigid motion reduces the general interpolation of tilings to interpolation of tile edges. Any algorithm that interpolates continuously between two paths can be applied to effect a smooth transition. Linear interpolation applied to piecewise-linear tile edges can produce designs like the ones shown in Figure 5.2. Note that when the two tilings are of different isohedral types, there may be several different intermediate shapes. This situation arises when the aspects of the two tiling types do not match up, causing tiles with different relative orientations to be identified. It is not clear how to resolve this problem if we wish to satisfy Huff's goal of having a single shape at every stage in the interpolation.

The third case is easy to carry out temporally. Because the two tilings are of the same isohedral type, it is easy to create a one-to-one correspondence between their combinatoric features (tiles, tiling vertices, and tiling edges). We can then interpolate tiling vertices by linearly interpolating between the two tiling vertex parameters, and interpolate tile edges as described above. Though continuous, this interpolation may cause the tiling to undergo an arbitrary affine transform (as in the

case of squares deforming into parallelograms), which does not necessarily make for a very "stable" animation.

The spatial variation of the third case is difficult. To draw the interpolation, we must first lay down an arrangement of tiling vertices that gradually changes from that of $\mathcal{T}_1$ to that of $\mathcal{T}_2$. But even within a single isohedral type, configurations of tiling vertices can change dramatically. The problem is exacerbated by the fact that the interpolation is done in the same space that the tiling is drawn. In the temporal case, there is no such interference. One possible solution is to use the underlying correspondence between tiling vertices to linearly interpolate between a tiling vertex's positions in $\mathcal{T}_1$ and $\mathcal{T}_2$. In this case, it makes sense to minimize the global affine transform between the two sets of tiling vertices, in order to make the line segment connecting any two corresponding vertices as short as possible. This minimization can be achieved using an iterated closest point algorithm such as Umeyama's [133]. Again, once the tiling vertices are laid out, tile edges can be interpolated. This approach can produce unsatisfactory results because even when the global affine transform is minimized, the interpolation can still bend and bulge, destroying the clean linear progression found in Huff's deformations. More work is needed to determine how to align the two tilings in such a way that the interpolation can be done cleanly in a strip.

The fourth case is very much like the third. Because the two tilings have the same topology, the Laves tiling with that topology can be expressed in the parameterizations of the isohedral types of both $\mathcal{T}_1$ and $\mathcal{T}_2$. This shared tiling can then be used to form the correspondence between tiling vertices, from which the previous interpolation methods follow.

The general case is the trickiest; in addition to all the difficulties encountered so far, we must account for a change in the very topology of the tiling. Thus, there can no longer be a clear correspondence between tiling vertices. On the other hand, many of Huff's examples achieve topological transitions without much effort. It seems important to handle this case to some extent.

As was pointed out in Section 4.4, as a tiling edge degenerates in an isohedral tiling, the tiling undergoes a discontinuous transition to an isohedral type of a different topology. Although the transition is topologically discontinuous, it has a smooth appearance, and is therefore suitable for parquet deformations. I hypothesize that these degeneracies might be used as "gateways" to make the transition between topologies.
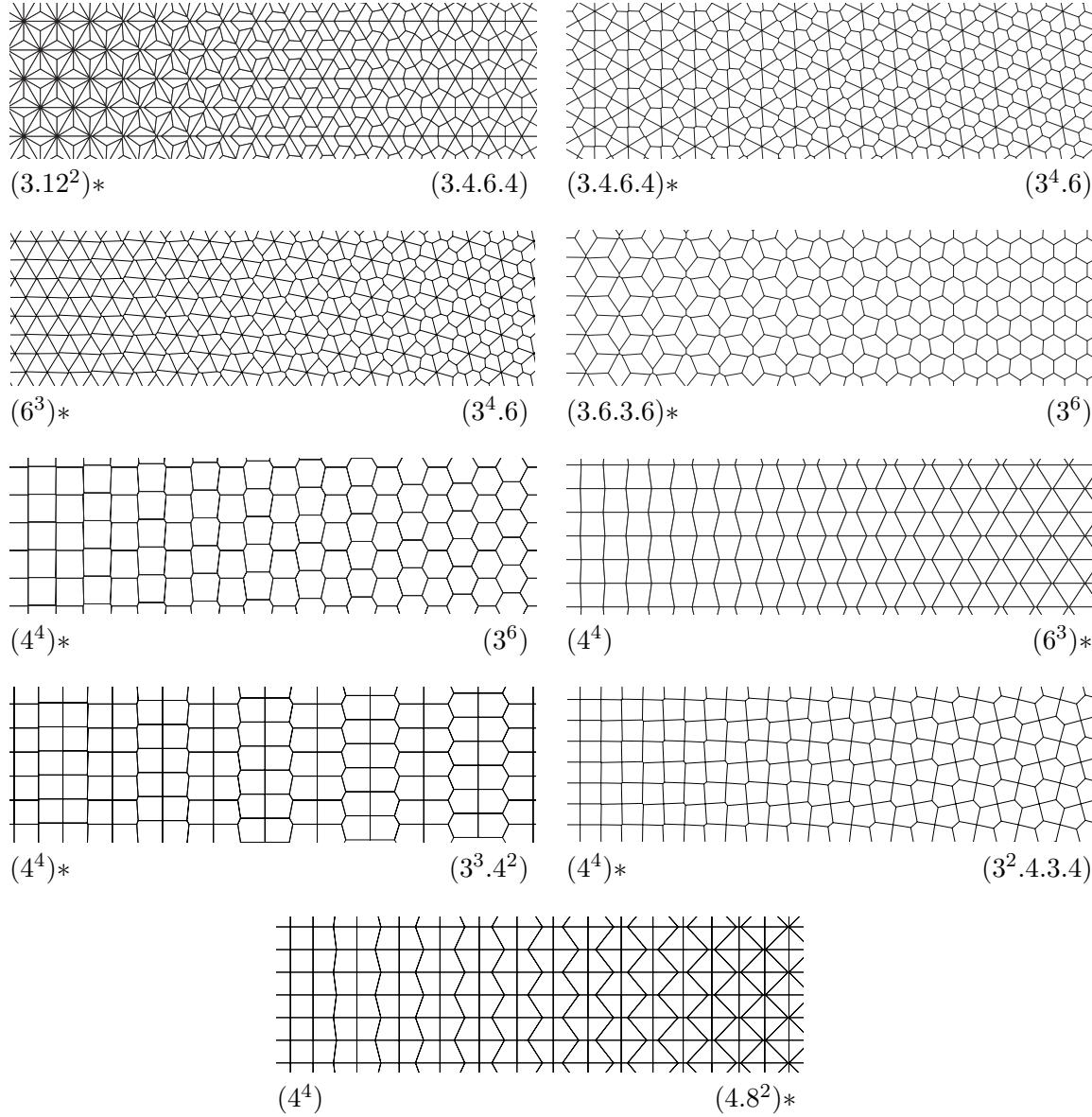
We can now imagine carrying out general interpolations. Given isohedral tilings $\mathcal{T}_1$ and $\mathcal{T}_2$ of

distinct topological types, we could identify a gateway tiling $\mathcal{T}_G$ that has the same topology as $\mathcal{T}_1$ and is also the degenerate case of a tiling with the same topology as $\mathcal{T}_2$. We could then build a transition by concatenating (temporally or spatially) the transitions from $\mathcal{T}_1$ to $\mathcal{T}_G$ and $\mathcal{T}_G$ (viewed now as degenerate) to $\mathcal{T}_2$.

Even when it is not clear how to transition directly between two topology types, it might be possible to break the problem down into multiple steps to be assembled through concatenation. We might then reduce all topological transitions down to a set of base cases, each one a smooth transition from one Laves tiling to another. As long as any pair of Laves tilings is joined via a path of base cases, we should be able to move between any two isohedral types. I have found topological transitions that obey all the restrictions of parquet deformations and that unify all the Laves tilings except for $(4.6.12)$. These deformations are shown in Figure 5.3. I conjecture that no smooth transition is possible into or out of that tiling. Fortunately, ignoring $(4.6.12)$ leaves out exactly one isohedral type: IH77. The unreachability of that type need not be considered a significant shortcoming.

Finally, note that when multiple transitions are chained together, it has been assumed that the chaining is done through simple concatenation. This approach limits the aesthetic range of interpolation. In the temporal case, the passage through a gateway tiling may be continuous, but exhibit a jarring derivative discontinuity. In the spatial case, we would like to pass from tiling $\mathcal{T}_1$ to tiling $\mathcal{T}_2$ without having to see all the intermediate steps used to make the transition. I hypothesize that in addition to *concatenating* interpolations, we should be able to *compose* them, and have both interpolations occur simultaneously. Any sequence of interpolations could then be composed together, yielding a smooth deformation directly from one tiling to another.

As an analogy, consider the motion of a point along a line segment. If we wish to move from position p_1 to p_2 and then from p_2 to p_3, we might simply concatenate the two trajectories; this new path will exhibit a discontinuous change in direction (and speed, if the segments have different lengths). However, de Casteljau's algorithm for drawing a quadratic Bézier curve with control points p_1, p_2, and p_3 short-circuits the linear trajectory and creates a smooth path that composes the two original segments. I would like to search for a method of composing tiling interpolations in the spirit of de Casteljau's algorithm.

194

$(3.12^2)*$ $(3.4.6.4)$ $(3.4.6.4)*$ $(3^4.6)$

$(6^3)*$ $(3^4.6)$ $(3.6.3.6)*$ (3^6)

$(4^4)*$ (3^6) (4^4) $(6^3)*$

$(4^4)*$ $(3^3.4^2)$ $(4^4)*$ $(3^2.4.3.4)$

(4^4) $(4.8^2)*$

Figure 5.3 A collection of parquet deformations between the Laves tilings. Each deformation starts and ends at a Laves tiling, as marked under the diagram. Each one will necessarily have a topological discontinuity somewhere along its length. The deformations presented here all have discontinuities at one of the endpoints; this endpoint is marked with an asterisk. By concatenating or composing these deformations, we should be able to transition between any two Laves tilings other than $(4.6.12)$.

5.5 A computational theory of pattern

Symmetry is a kind of redundancy, but not all redundancy is symmetry. I have presented several examples in this dissertation of mathematical or ornamental structures for which an analysis by symmetry fails to capture some of the redundancy. Quasiperiodic tilings are a prime example; they have a tremendous amount of structure and repetition but almost no symmetry. As another example, in the construction of Islamic star patterns it was necessary to move from symmetry groups to a more localized breakdown of the plane based on tilings. Grünbaum exhibits several more examples in a paper that awakens mathematicians to the fact that symmetry is not a panacea in the study of repeated patterns [64]. In another paper [65], he criticizes the "group theory cult," a cadre of mathematicians and historians who follow Speiser in believing that the only possible characterization of order in ornament is via the group-theoretic methods of symmetry. Simply put, by expressing only the global redundancy in a pattern, symmetry fails to discern any finer structure that occurs locally.

Following more recent writing by Grünbaum on the subject [66], we may use the imprecise term "orderliness" to refer generally to structure or rules in a planar figure. Certainly, symmetry is one possible form of orderliness, but Grünbaum explores several other possible conceptions of order that do not coincide with symmetry. While it is unrealistic to seek a universal theory of orderliness, there are certainly many specific avenues that may be explored. New descriptions of order can allow us to account for more of the features of a design, and to provide a finer-grained classification of patterns than that afforded by symmetry alone.

I believe that one powerful means of understanding orderliness may lie in the study of formal languages. Symmetries are derived from a finished pattern, with no ability to see that pattern as being built step-by-step, one motif at a time. I propose to use a formal language to represent the set of legal motif placements, regarding each word in the language as actively placing a single motif rather than describing some large-scale redundancy of the pattern as a whole.

Epstein *et al.* [46, Section 2.1] provide a bridge between formal languages and group theory that can help us take the first steps away from symmetry (their development seems to owe a great deal to combinatorial group theory [108]). We start with a group G, a set A of formal symbols, and an injective map p from A to G. We can naturally define a function π from A^*, the set of all words over A, to G by interpreting concatenation of symbols as group multiplication. If the set
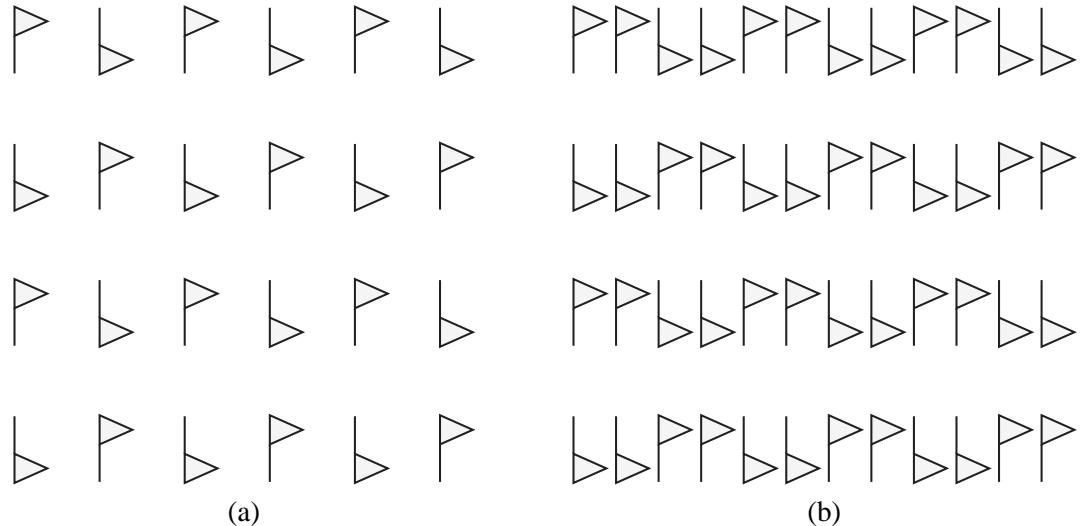
$\{p(x)|x \in A\}$ generates G, then π is a surjection and we can also think of A as "generating" G.

We can now see languages over A as representing subsets of the group G. Of particular interest are those languages L for which π is a bijection between L and G. The words of L then correspond exactly to the elements of G. Working with G via L brings us closer to a computational view of patterns, because in practice we often "assemble" a member of G by composing together a sequence of generators (*i.e.*, a word in L). If L is a particularly well-behaved language, it might even be possible to enumerate the elements of L in a useful order, which could correspond to a program that transforms a motif to every position in a pattern in a disciplined way (very much like the replication algorithms of Section 4.4.3).

The theory of automatic groups [46] shows how certain groups have languages that can be dealt with very efficiently by computer. In particular, for some groups there exists a finite automaton that can be used to enumerate the elements of the group in order by length. This automaton can then be converted into a table-driven algorithm that replicates motifs according to the $[p, q]$ symmetry groups of the hyperbolic plane [101].

The next step is to sever all ties with group theory and investigate patterns generated directly from formal languages. We define a (discrete) pattern as a motif M and a language L over an alphabet A, where each $x \in A$ corresponds to an automorphism of the plane. Every word in L maps to a transformation by composing the automorphisms associated with each symbol in the word. Computational pattern theory is then the study of the properties of planar patterns that can be determined from the behaviour of these languages. We can generally consider languages L that are surjective on a given pattern (several words in L might map to the same transformation), or focus more precisely on bijective languages. It is also easy to work with finite pieces of a pattern through finite subsets of its language.

As an elementary example, consider the two patterns of Figure 5.4. Although clearly different, these two patterns have the same symmetry group. In (b), pairs of adjacent flags with the same orientation will be lumped together, because there is no way to recognize them as two independent, congruent motifs using symmetry. On the other hand, if L_1 is the language over alphabet A_1 corresponding to the pattern of (a), we can define $A_2 = A_1 \cup \{t\}$, where t maps to a horizontal translation by half the distance between two flags in (a). It then follows that the language of the pattern in (b) can be defined over A_2 as $L_1 \cup L_1 t$ (where for language L and symbol x, $Lx = \{wx|w \in L\}$).

Figure 5.4 Examples of two patterns for which symmetry groups fail to make a distinction, but formal languages might. The "egalitarian patterns" presented by Grünbaum [65] are another form of orderliness sufficiently rich to distinguish between these patterns.

Note that this outlook on the structure of (b) also seems reminiscent of Leyton's "generative theory of shape," [102] where a final shape (a pattern in our case) is expressed as a "control-nested structure." An outer group called the control group, here corresponding to L_1, operates on an inner fiber group, here corresponding to the set $\{\epsilon, x\}$ (ϵ is the empty word over A_2). In Leyton's theory, the two groups are combined using what is called a *wreath product*.

The isohedral tilings motivate a second example. In an isohedral tiling, every tile is surrounded by its neighbours in a consistent way. If the prototile has k tiling edges, we can define an alphabet $H = \{h_1, \ldots, h_k\}$ where h_i maps to the hop across the ith tiling edge (as defined in Section 4.4). Then the very simple language H^* is surjective on the tiling. By contrast, a tiling that is not isohedral cannot be of the form H^* for any alphabet H. It is slightly more complicated, but still straightforward, to give a bijective language for an isohedral tiling. Because the tiling is periodic, we can define symbols t_1 and t_2 for the two translations, and symbols $a_1, \ldots, a_m$ for the m aspect transforms of the tiling (again as determined in Section 4.4). We also use T_1 and T_2 as "formal inverses" of t_1 and t_2. The formal inverses are symbols that stand for the mathematical inverses of the automorphisms associated with t_1 and t_2. A bijective regular language for the tiling is then
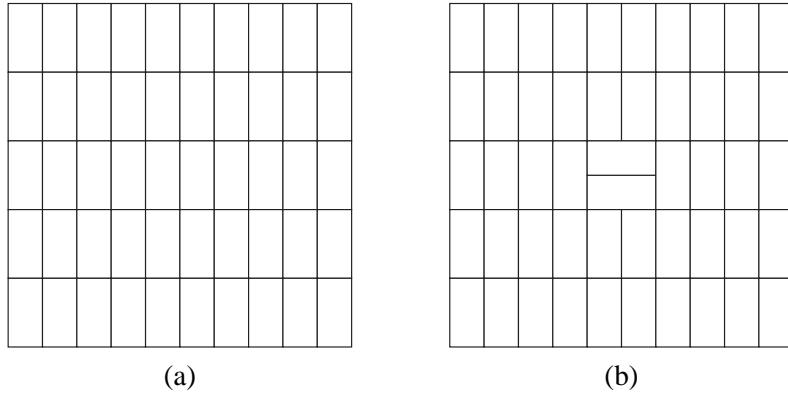
given by the regular expression $(t_1^* \cup T_1^*)(t_2^* \cup T_2^*)(a_1 \cup a_2 \ldots \cup a_m)$. The first two factors select a particular translational unit, and the third an aspect within that unit.

The languages that can be associated with orderly patterns are more numerous than the seventeen wallpaper groups, and so it seems likely that more can be said about a pattern from its language than from its symmetry group. One especially interesting question that might benefit from a language-theoretic approach is that of measuring the information content of a pattern. The traditional tools of information theory do not apply here. However, we might consider measuring the *Kolmogorov complexity* of a pattern. Roughly speaking, the Kolmogorov complexity of a string of symbols is measured as the length of the shortest program that emits that string (when run with no input) and halts [103]. (As with classical complexity theory, we do not seek a numerical value for Kolmogorov complexity. Rather, we are interested in asymptotic results and relative complexity of different strings. Kolmogorov complexity is also useful in establishing the existence of a string with certain properties from among a collection of strings.)

It seems as if Kolmogorov complexity could be extended to patterns by considering the shortest program that loops forever, spitting out words from a language that correspond to a non-redundant enumeration of the pattern's motifs. Alternatively, it might be more fruitful to consider the shortest decision procedure for the language (a decision procedure for a language is an algorithm that is given a word and always halts, producing a yes/no answer depending on whether the word belongs to the language).

The two tilings of Figure 5.5 are a kind of converse of Figure 5.4. Here we have two tilings that are nearly the same, but for which the symmetries are vastly different. Kolmogorov complexity might help quantify the meaning of "nearly the same" here. The tiling in (b) should have only slightly more information than that of (a), because its language can be constructed by taking the language of (a), removing the words for two tiles, and grafting in rotated versions. In a similar vein, the tiling of Figure 2.10, derived in a contrived manner from the digits of π, ought to have a Kolmogorov complexity that depends fundamentally on the complexity of π as a string.

The ideas of a computational theory of patterns and of the formal complexity of patterns raise many deep and promising questions for future work. Here are some first challenges not mentioned above:

Figure 5.5 Two tilings which would appear to have nearly the same information content, but vastly different symmetries.

- What patterns correspond to different classes of language? Every transitive pattern (*i.e.*, where the pattern's symmetries act transitively on the motifs) has a regular language. What patterns might correspond to context-free languages and to context-sensitive languages? Given a pattern, how can we determine the class of its language?

- How can we layer other traditional motif-specific information onto the language-theoretic base presented here? We should at least be able to account for multiple motif shapes and colours.

- What languages correspond to the placement of tiles in aperiodic tilings? One surjective answer for cases like the Penrose tilings is to let the symbols of an alphabet correspond to the deflation rules of the prototiles, and consider all words over the alphabet of some finite length (mapping to a uniform level of deflation everywhere). Is there a simple language for the Penrose tilings that doesn't rely on scaling?

- Can a complexity measure be used to distinguish between well-known complexity levels in patterns? In other words, if we order patterns as transitive, periodic, quasiperiodic, aperiodic, and so on, are there well defined boundaries of complexity between these classes? Can a measure of complexity help to manufacture new patterns with desired properties?

200

BIBLIOGRAPHY

[1] Jan Abas. A hyperbolic mural. `http://www.bangor.ac.uk/~mas009/ppic7.html`.

[2] Syed Jan Abas and Amer Shaker Salman. *Symmetries of Islamic Geometrical Patterns*. World Scientific, 1995.

[3] Steve Abbott. Steve Abbott's computer drawn Celtic knotwork. `http://www.abbott.demon.co.uk/knots.html`.

[4] Maneesh Agrawala and Chris Stolte. Rendering effective route maps: Improving usability through generalization. *Proceedings of SIGGRAPH 2001*, pages 241–250, 2001.

[5] Nina Amenta and Mark Phillips. Kali. `http://www.geom.umn.edu/java/Kali/`.

[6] E. M. Arkin, L. P. Chew, D. P. Huttenlocher, K. Kedem, and J. S. B. Mitchell. An efficiently computable metric for comparing polygonal shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:209–216, 1991.

[7] George Bain. *The Methods of Construction of Celtic Art*. Dover, 1973.

[8] Thaddeus Beier and Shawn Neely. Feature-based image metamorphosis. *Proceedings of SIGGRAPH'92*, pages 35–42, 1992.

[9] John Berglund. Anisohedral tilings page. `http://www.angelfire.com/mn3/anisohedral/`.

[10] John Berglund. Is there a k-anisohedral tile for $k \geq 5$? *American Mathematical Monthly*, 100:585–588, 1993.

[11] Jinny Beyer. *Designing Tesselations: The Secrets of Interlocking Patterns*. Contemporary Books, 1999.

[12] Jay Bonner. Geodazzlers. `http://www.dstoys.com/GD.html`.

[13] Jay Francis Bonner. *Islamic Geometric Patterns: Their Historical Development and Traditional Methods of Derivation*. Unpublished, 2000.

[14] Roberto Bonola. *Non-Euclidean Geometry*. Dover Publications, 1955.

[15] F. H. Bool, J. R. Kist, J. L. Locher, and F. Wierda. *M. C. Escher: His Life and Complete Graphic Work*. Harry N. Abrams, Inc., 1992.

[16] J. Bourgoin. *Arabic Geometrical Pattern and Design*. Dover Publications, 1973.

[17] Cameron Browne. Font decoration by automatic mesh fitting. In R.D. Hersch, J. Andr, and H. Brown, editors, *Electronic Publishing, Artistic Imaging, and Digital Typography*, pages 23–43. Springer Verlag, 1998.

[18] Paul Burchard, Daeron Meyer, and Eugenio Durand. Quasitiler. `http://www.geom.umn.edu/apps/quasitiler/`.

[19] Jean-Marc Castéra. *Arabesques: Decorative Art in Morocco*. ACR Edition, 1999.

[20] Jean-Marc Castéra. Zellijs, muqarnas and quasicrystals. In Nathaniel Friedman and Javiar Barrallo, editors, *ISAMA 99 Proceedings*, pages 99–104, 1999.

[21] William W. Chow. Automatic generation of interlocking shapes. *Computer Graphics and Image Processing*, 9:333–353, 1979.

[22] Archibald Christie. *Traditional Methods of Pattern Designing*. Oxford University Press, 1929.

[23] Alain Connes. *Noncommutative Geometry*. Academic Press, San Diego, 1994.

[24] H. S. M. Coxeter. Coloured symmetry. In H. S. M. Coxeter *et al.*, editor, *M.C. Escher: Art and Science*, pages 15–33. Elsevier Science Publishers B.V., 1986.

[25] H. S. M. Coxeter. *Non-Euclidean Geometry*. Mathematical Association of America, sixth edition, 1998.

[26] H. S. M. Coxeter and W. O. J. Moser. *Generators and Relations for Discrete Groups*. Springer-Verlag, 1980.

[27] Keith Critchlow. *Islamic Patterns: An Analytical and Cosmological Approach*. Thames and Hudson, 1976.

[28] Hallard T. Croft, Kenneth J. Falconer, and Richard K. Guy. *Unsolved Problems in Geometry*. Springer-Verlag, 1991.

[29] Peter R. Cromwell. Celtic knotwork: Mathematical art. *The Mathematical Intelligencer*, 15(1):36–47, 1993.

202

[30] L. Danzer, B. Grünbaum, and G. C. Shephard. Can all tiles of a tiling have five-fold symmetry? *American Mathematical Monthly*, 89:568–585, 1982.

[31] Prisse d'Avennes. *Arabic Art*. L'Aventurine, 2001.

[32] Doug DeCarlo and Anthony Santella. Stylization and abstraction of photographs. *Proceedings of SIGGRAPH 2002*, pages 769–776, 2002.

[33] Olaf Delgado, Daniel Huson, and Elizaveta Zamorzaeva. The classification of 2-isohedral tilings of the plane. *Geometricae Dedicata*, 42:43–117, 1992.

[34] Olaf Delgado-Friedrichs. Data structures and algorithms for tilings. To be published in *Theoretical Computer Science*.

[35] Tony DeRose, Michael Kass, and Tien Truong. Subdivision surfaces in character animation. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques (SIGGRAPH'98)*, pages 85–94. ACM Press, 1998.

[36] A.K. Dewdney. *The Tinkertoy Computer and Other Machinations*, pages 222–230. W. H. Freeman, 1993.

[37] Robert Dixon. Two conformal mappings. In Michele Emmer, editor, *The Visual Mind: Art and Mathematics*, pages 45–48. MIT Press, 1993.

[38] Andreas W. M. Dress. The 37 combinatorial types of regular "Heaven and Hell" patterns in the euclidean plane. In H. S. M. Coxeter *et al.*, editor, *M.C. Escher: Art and Science*, pages 35–45. Elsevier Science Publishers B.V., 1986.

[39] Douglas Dunham. Hyperbolic symmetry. *Computers and Mathematics with Applications*, 12B(1/2):139–153, 1986.

[40] Douglas Dunham. Artistic patterns in hyperbolic geometry. In Reza Sarhangi, editor, *Bridges 1999 Proceedings*, pages 139–149, 1999.

[41] Douglas Dunham. Hyperbolic Islamic patterns — a beginning. In Reza Sarhangi, editor, *Bridges 2001 Proceedings*, pages 247–254, 2001.

[42] Douglas Dunham, John Lindgren, and David Witte. Creating repeating hyperbolic patterns. *Computer Graphics (Proc. SIGGRAPH)*, pages 215–223, 1981.

[43] Douglas J. Dunham. Creating hyperbolic escher patterns. In H. S. M. Coxeter *et al.*, editor, *M.C. Escher: Art and Science*, pages 241–247. Elsevier Science Publishers B.V., 1986.

[44] David S. Ebert, F. Kenton Musgrave, Darwyn Peachey, Ken Perlin, and Steven Worley. *Texturing and Modeling*. AP Professional, 1998.

[45] David Eppstein. The geometry junkyard: Penrose tiling. `http://www.ics.uci.edu/~eppstein/junkyard/penrose.html`.

[46] David B. A. Epstein, J .W. Cannon, D. F. Holt, S. V. F. Levy, M. S. Paterson, and W. P. Thurston. *Word Processing in Groups*. Jones and Bartlett, 1992.

[47] Bruno Ernst. *The Magic Mirror of M. C. Escher*. Ballantine Books, New York, 1976.

[48] George A. Escher. M.C. escher at work. In H. S. M. Coxeter *et al.*, editor, *M.C. Escher: Art and Science*, pages 1–8. Elsevier Science Publishers B.V., 1986.

[49] M. C. Escher. *Escher on Escher: Exploring the Infinite*. Henry N. Abrams, Inc., 1989. Translated by Karin Ford.

[50] Richard L. Faber. *Foundations of Eucliedean And Non-Euclidean Geometry*. Marcel Dekker, Inc., 1993.

[51] Michael Field and Martin Golubitsky. *Symmetry in Chaos*. Oxford University Press, 1992.

[52] François Dispot. Arabeske home page. `http://www.wozzeck.net/arabeske/index.html`.

[53] The gimp. `http://www.gimp.org/`.

[54] Andrew Glassner. Andrew Glassner's notebook: Aperiodic tiling. *IEEE Computer Graphics & Applications*, 18(3):83–90, may–jun 1998. ISSN 0272-1716.

[55] Andrew Glassner. Andrew Glassner's notebook: Penrose tiling. *IEEE Computer Graphics & Applications*, 18(4), jul–aug 1998. ISSN 0272-1716.

[56] Andrew Glassner. Andrew Glassner's notebook: Celtic knots, part II. *IEEE Computer Graphics & Applications*, 19(6):82–86, nov–dec 1999. ISSN 0272-1716.

[57] Andrew Glassner. Andrew Glassner's notebook: Celtic knotwork, part I. *IEEE Computer Graphics & Applications*, 19(5):78–84, sep–oct 1999. ISSN 0272-1716.

[58] Andrew Glassner. Andrew Glassner's notebook: Celtic knots, part III. *IEEE Computer Graphics & Applications*, 20(1):70–75, jan–feb 2000. ISSN 0272-1716.

[59] Solomon W. Golomb. *Polyominoes: Puzzles, Patterns, Problems and Packings*. Princeton University Press, second edition, 1994.

[60] E. H. Gombrich. *The Sense of Order: A Study in the Psychology of Decorative Art*. Phaidon Press Limited, second edition, 1998.

[61] Chaim Goodman-Strauss. A small set of aperiodic tiles. *European Journal of Combinatorics*, 20:375–384, 1999.

[62] Marvin J. Greenberg. *Euclidean and Non-Euclidean Geometries: Development and History*. W. H. Freeman and Company, third edition, 1993.

[63] Bathsheba Grossman. Bathsheba Grossman – sculpting geometry. `http://www.bathsheba.com/`.

[64] Branko Grünbaum. The emperor's new clothes: Full regalia, G string, or nothing? *The Mathematical Intelligencer*, 6(4):47–53, 1984.

[65] Branko Grünbaum. Periodic ornamentation of the fabric plane: Lessons from Peruvian fabrics. *Symmetry*, 1(1):45–68, 1990.

[66] Branko Grünbaum. Levels of orderliness: Global and local symmetry. In I. Hargittai and T. C. Laurent, editors, *Symmetry 2000*, pages 51–61. Portland Press, London, 2002.

[67] Branko Grünbaum and G. C. Shephard. Spherical tilings with transitivity properties. In Chandler Davis, Branko Grünbaum, and F. A. Sherk, editors, *The Geometric Vein: The Coxeter Festschrift*, pages 65–94. Springer-Verlag, New York, 1982.

[68] Branko Grünbaum and G. C. Shephard. *Tilings and Patterns*. W. H. Freeman, 1987.

[69] Branko Grünbaum and G. C. Shephard. Interlace patterns in Islamic and Moorish art. *Leonardo*, 25:331–339, 1992.

[70] E. H. Hankin. *The Drawing of Geometric Patterns in Saracenic Art*, volume 15 of *Memoirs of the Archaeological Society of India*. Government of India, 1925.

[71] E. Hanbury Hankin. Examples of methods of drawing geometrical arabesque patterns. *The Mathematical Gazette*, pages 371–373, May 1925.

[72] E. Hanbury Hankin. Some difficult Saracenic designs II. *The Mathematical Gazette*, pages 165–168, July 1934.

[73] E. Hanbury Hankin. Some difficult Saracenic designs III. *The Mathematical Gazette*, pages 318–319, December 1936.

[74] István Hargittai and Magdolna Hargittai. *Symmetry through the Eyes of a Chemist*. Plenum Press, New York, second edition, 1995.

[75] George W. Hart. Conway notation for polyhedra. `http://www.georgehart.com/virtual-polyhedra/conway_notation.html`.

[76] George W. Hart. Geometric sculpture. `http://www.georgehart.com/sculpture/sculpture.html`.

[77] George W. Hart. Uniform polyhedra. `http://www.georgehart.com/virtual-polyhedra/uniform-info.html`.

[78] Barbara Hausmann, Britta Slopianka, and Hans-Peter Seidel. Exploring plane hyperbolic geometry. In Hans-Christian Hege and Konrad Polthier, editors, *Visualization and Mathematics*, pages 21–36. Springer, 1997.

[79] H. Heesch. Aufbau der ebene aus kongruenten bereichen. *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen*, pages 115–117, 1935. John Berglund provides an online English translation at `http://www.angelfire.com/mn3/anisohedral/heesch35.html`.

[80] H. Heesch and O. Kienzle. *Flachenschluss*. Springer–Verlag, 1963.

[81] Richard E. Hodel. *An Introduction to Mathematical Logic*. PWS Publishing Company, 1995.

[82] Paul Hoffman. *The Man Who Loved Only Numbers*. Hyperion, New York, 1998.

[83] Douglas Hofstadter. *Metamagical Themas: Questing for the Essence of Mind and Pattern*. Bantam Books, 1986.

[84] Alan Holden. *Shapes, Space, and Symmetry*. Dover Publications, Inc., 1991.

[85] Edmund S. Howe. Effects of partial symmetry, exposure time, and backward masking on judged goodness and reproduction of visual patterns. *Quarterly Journal of Experimental Psychology*, 32:27–55, 1980.

[86] Daniel H. Huson. The generation and classification of tile-k-transitive tilings of the euclidean plane, the sphere, and the hyperbolic plane. *Geometriae Dedicata*, 47:269–296, 1993.

[87] Pedagoguery Software Inc. Tess. `http://www.peda.com/tess/Welcome.html`.

[88] Washington Irving. *The Alhambra*. Macmillan and co., 1931.

[89] Drew Ivans. The origin and meaning of Celtic knotwork. `http://www.craytech.com/drew/knotwork/knotwork-meaning.html`.

[90] Robert Jensen and Patricia Conway. *Ornamentalism: the New Decorativeness in Artchitecture & Design*. Clarkson N. Potter, Inc., 1982.

[91] Owen Jones. *The Grammar of Ornament*. Studio Editions, 1986.

[92] Craig S. Kaplan. `isohedral.ih`. `http://www.cs.washington.edu/homes/csk/tile/escherization.html`.

[93] Craig S. Kaplan. Computer generated islamic star patterns. In Reza Sarhangi, editor, *Bridges 2000 Proceedings*, 2000.

[94] Craig S. Kaplan. Computer generated islamic star patterns. *Visual Mathematics*, 2(3), 2000. `http://members.tripod.com/vismath4/kaplan/index.html`.

[95] Craig S. Kaplan and George W. Hart. Symmetrohedra: Polyhedra from symmetric placement of regular polygons. In Reza Sarhangi, editor, *Bridges 2001 Proceedings*, 2001.

[96] Craig S. Kaplan and David H. Salesin. Escherization. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques (SIGGRAPH 2000)*, pages 499–510. ACM Press/Addison-Wesley Publishing Co., 2000.

[97] David C. Kay. *College Geometry*. Holt, Rinehart and Winston, Inc., 1969.

[98] Haresh Lalvani. Pattern regeneration: A focus on Islamic jalis and mosaics. In Jan Pieper and George Michell, editors, *The Impulse to Adorn*, pages 123–136. Marg Publications, 1982.

[99] A.J. Lee. Islamic star patterns. *Muqarnas*, 4:182–197, 1995.

[100] Raph Levien. libart. `http://www.levien.com/libart/`.

[101] Silvio Levy. Automatic generation of hyperbolic tilings. In Michele Emmer, editor, *The Visual Mind: Art and Mathematics*, pages 165–170. MIT Press, 1993.

[102] Michael Leyton. *A Generative Theory of Shape*, volume 2145 of *Lecture Notes in Computer Science*. Springer, 2001.

[103] Ming Li and Paul Vitányi. *An Introduction to Kolmogorov Complexity and its Applications*. Graduate texts in Computer Science. Springer-Verlag, Inc., New York, second edition, 1997.

[104] Stanley B. Lippman and Josée Lajoie. *C++ Primer*. Addison-Wesley, third edition, 1998.

[105] J. L. Locher. *The Magic of M. C. Escher*. Harry N. Abrams, Inc., 2000. Designed by Erik Thé.

[106] P. Locher and C. Nodine. The perceptual value of symmetry. *Computers and Mathematics With Applications*, 17(4–6):475–484, 1989.

[107] Z. Lučić and E. Molnár. Fundamental domains for planar discontinuous groups and uniform tilings. *Geometriae Dedicata*, 40:125–143, 1991.

[108] Wilhelm Magnus, Abraham Karrass, and Donald Solitar. *Combinatorial group theory: presentations of groups in terms of generators and relations*. Dover Publications, 1976.

[109] George E. Martin. *The Foundations of Geometry and the Non-Euclidean Plane*. Intext Educational Publishers, 1975.

[110] Scott McCloud. *Understanding Comics*. HarperCollins, 1993.

[111] Christian Mercat. Celtic knotwork, the ultimate tutorial. `http://www.entrelacs.net/en.index.html`.

[112] Kevin Mitchell. Semi-regular tilings of the plane. `http://people.hws.edu/mitchell/tilings/Part1.html`.

[113] Anders Pape Møller and John P. Swaddle. The biological importance of imperfect symmetry. In *Asymmetry, Developmental Stability and Evolution*, chapter W1. Oxford University Press, 1997. Available online from `http://www.oup.co.uk/isbn/0-19-854894-X`.

[114] E. N. Mortenson and W. A. Barrett. Intelligent scissors for image composition. *Proceedings of SIGGRAPH 1995*, pages 191–198, 1995.

[115] Joseph Myers. Polyomino, polyhex, and polyiamond tiling. `http://www.srcf.ucam.org/~jsm28/tiling/`.

[116] Gülru Necipoğlu. *The Topkapı Scroll — Geometry and Ornament in Islamic Architecture*. The Getty Center for the History of Art and the Humanities, 1995.

[117] John Allen Paulos. *Mathematics and Humor*. The University of Chicago Press, 1980.

[118] Roger Penrose. Escher and the visual representation of mathematical ideas. In H. S. M. Coxeter *et al.*, editor, *M.C. Escher: Art and Science*, pages 143–157. Elsevier Science Publishers B.V., 1986.

[119] Ivars Peterson. *The Mathematical Tourist: Snapshots of Modern Mathematics*. W. H. Freeman and Company, New York, 1988.

[120] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, second edition, 1992. ISBN 0-521-43108-5. Held in Cambridge.

208

[121] A. Racinet. *The Encyclopedia of Ornament*. Studio Editions, 1988.

[122] J. F. Rigby. Butterflies and snakes. In H. S. M. Coxeter *et al.*, editor, *M.C. Escher: Art and Science*, pages 211–220. Elsevier Science Publishers B.V., 1986.

[123] Doris Schattschneider. Escher's classification system for his colored periodic drawings. In H. S. M. Coxeter *et al.*, editor, *M.C. Escher: Art and Science*, pages 82–96. Elsevier Science Publishers B.V., 1986.

[124] Doris Schattschneider. *M.C. Escher: Visions of Symmetry*. W.H. Freeman, 1990.

[125] P. J. Schneider. An algorithm for automatically fitting digitized curves. In *Graphics Gems*, pages 612–626. Academic Press, Boston, 1990.

[126] G. C. Shephard. What Escher might have done. In H. S. M. Coxeter *et al.*, editor, *M.C. Escher: Art and Science*, pages 111–122. Elsevier Science Publishers B.V., 1986.

[127] A. V. Shubnikov and V. A. Koptsik. *Symmetry in Science and Art*. Plenum Press, 1974.

[128] Peter Stampfli. New quasiperiodic lattices from the grid method. In István Hargittai, editor, *Quasicrystals, Networks and Molecules of Fivefold Symmetry*, chapter 12, pages 201–221. VCH Publishers, 1990.

[129] Desmond Stewart. *The Alhambra*. Newsweek, 1974.

[130] Ian Stewart. *The Problems of Mathematics*. Oxford University Press, 1987.

[131] Peter G. Szilagyi and John C. Baird. A quantitative approach to the study of visual symmetry. *Perception & Psychophysics*, 22(3):287–292, 1977.

[132] C. W. Tyler. The human expression of symmetry: Art and neuroscience. `http://www.ski.org/CWTyler_lab/CWTyler/Art\%20Investigations/Symmetry/S%ymmetry.html`.

[133] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):376–380, April 1991.

[134] David Wade. *Pattern in Islamic Art*. The Overlook Press, 1976.

[135] Dorothy K. Washburn and Donald W. Crowe. *Symmetries of Culture*. University of Washington Press, 1992.

[136] Magnus J. Wenninger. *Polyhedron Models*. Cambridge University Press, 1971.

[137] Hermann Weyl. *Symmetry*. Princeton Science Library, 1989.

[138] Stephen Wolfram. *A New Kind of Science*. Wolfram Media, Inc., 2002.

[139] Michael T. Wong, Douglas E. Zongker, and David H. Salesin. Computer–generated floral ornament. *Proceedings of SIGGRAPH'98*, pages 423–434, 1998.

[140] Jane Yen and Carlo Séquin. Escher sphere construction kit. In *Proceedings of the 2001 symposium on Interactive 3D graphics*, pages 95–98. ACM Press, 2001.

[141] Douglas Zongker. Celtic knot thingy. `http://www.cs.washington.edu/homes/dougz/hacks/knot/.`

[142] Douglas Zongker. Creation of overlay tilings through dualization of regular networks. In Nathaniel Friedman and Javiar Barrallo, editors, *ISAMA 99 Proceedings*, pages 495–502, 1999.

VITA

Craig Steven Kaplan was born in 1972 in Montréal, Québec. As a child, he would spend hours staring at an image of Escher's *Relativity* in a book.

Craig spent five years as an undergraduate at the University of Waterloo, studying pure mathematics and computer science. He also spent a number of internships at Alias|Wavefront in Toronto. He then ventured to the beautiful Pacific Northwest to pursue a PhD at the University of Washington in Seattle. At the end of 2002 Craig, his wife Nathalie, their new daughter Zoë, and their three cats will make the trek back to Ontario, where he will return to Waterloo's School of Computer Science, this time as a member of the faculty.