Curve Evolution Schemes for Parquet Deformations

Craig S. Kaplan
Cheriton School of Computer Science
University of Waterloo
csk@uwaterloo.ca

Abstract

In this paper, I consider the question of how to carry out aesthetically pleasing evolution of the curves that make up the edges in a parquet deformation. Within the framework of simple arrangements of square tiles, I explore curve evolution models based on grids, iterated function systems, and organic labyrinthine paths.

1 Introduction

Parquet deformations are a style of graphic design invented by William Huff in the early 1960s [4, Chapter 10]. Each design is a kind of "spatial animation" in the plane, a tiling whose tiles gradually evolve along one (or sometimes two) dimensions. In an unpublished manuscript [5], Huff notes the aesthetic ties to Escher's metamorphoses (a connection I discussed previously [6]), though unlike Escher's tessellations, parquet deformations are intended to be abstract exercises in form. The examples from his studio are all black line drawings on a white background.

In previous work [6], I studied the mathematical challenges associated with the construction of parquet deformations. I chose to view this problem in the context of isohedral tilings, an expressive yet algorithmically tractable class of tilings of the plane by copies of a single shape [7]. Given two arbitrary isohedral prototiles, which act like keyframes in an animation, I asked whether there could exist a patch of tiles that interpolates smoothly between the keyframes. I defined a sequence of increasingly difficult variations of this problem as the two keyframe tiles shared fewer geometric properties.

That previous work was concerned mostly with the problem of identifying corresponding edges, with the expectation that any interpolation algorithm could then be applied to edges found to correspond. All examples in that paper were constructed using a naive algorithm for arclength-based linear interpolation between two piecewise polygonal paths; an example of this approach is shown in Figure 1. In this paper, I consider an orthogonal problem. That is, I assume that the tiling layouts are fixed and edge correspondences are trivial, and focus on different possible interpolation algorithms and the aesthetic possibilities they offer.

For the purposes of this paper, I consider parquet deformations made from a single isohedral tiling of topological type (4^4) (in which all tiles are quadrilaterals meeting four around every vertex). Furthermore, I fix the tiling vertices to a square lattice. All such tilings can be constructed via systematic modification of the edges of an infinite regular tiling of unit squares. I also assume that the goal is to "evolve" the initially straight edges of a prototile into more complex shapes (as opposed to morphing between two arbitrary prototiles). In this context most tiling-theoretic considerations become trivial; all that remains is to choose how tiling edges evolve. Here I present three promising possibilities for curve evolution that lead to interesting parquet deformations. These are: a grid-based method that evolves in discrete steps (Section 3), a method based on iterated function systems (Section 4), and a method based on organic evolution of labyrinthine paths (Section 5).

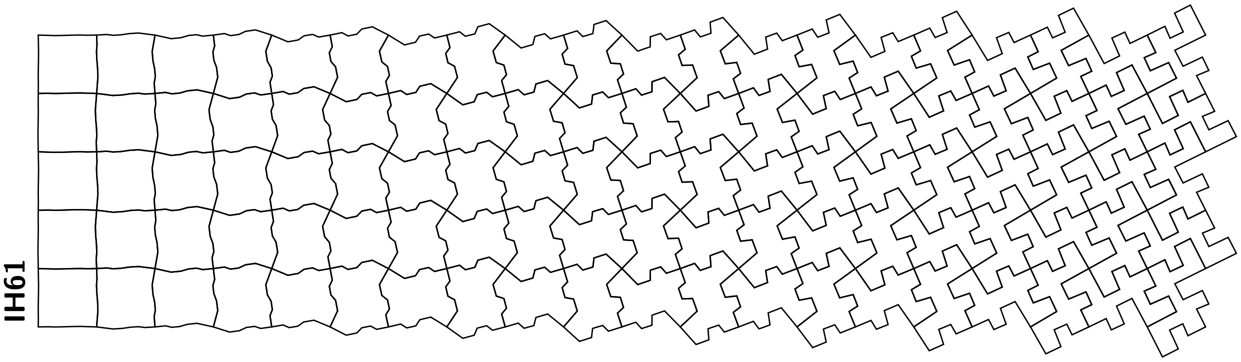

Figure 1 : *An example of a parquet deformation constructed via a naive linear interpolation of piecewise polygonal paths defining the edges of a prototile in a tiling. In all parquet deformations in this paper, the isohedral tiling type is identified at the left of the illustration.*

2 A framework for parquet deformations

The parquet deformations in this paper are based on manipulation of the edges of isohedral tilings with topology (4^4) and tiling vertices fixed in a square lattice. I have developed a software library for representing the shapes of prototiles belonging to this class, and for rendering the resulting deformations. The library uses previously described data structures and algorithms [7] to represent the shapes of the edges, sharing edge representations internally to express the redundancies in the tile's boundary. The rendering algorithm recognizes that every tile is aligned with a square in an infinite grid. The orientation of that tile is given by one of the eight symmetries of the square, and can be encoded compactly by giving a rectangular grid of orientations that repeats by horizontal and vertical translation.

Unlike previous work, each tile edge is parameterized by an additional time parameter t in the range $[0,1]$, allowing the edge to evolve as it is queried at different time values. Every placed copy of the prototile replaces some unit square in an infinite grid of squares. The (x,y) coordinates of the midpoints of this square's edges are passed to a "parameter space", a function that assigns a time value to every point in the plane. A typical parameter space would linearly interpolate time values between the minimum and maximum x positions of the parquet deformation that will eventually be drawn. These time values are passed to the parameterized edges, and the resulting curves are concatenated to produce the placed tile's boundary.

3 Grid-based curves

Among the parquet deformations documented by Huff, we find several in which straight tile edges evolve into convoluted paths by the successive addition of small unit square indentations. This approach is natural in a context where all changes to the pattern must be calculated and drawn by hand. Of course, it is also a technique that is well suited to a software implementation.

Let the plane be covered with an infinite grid of unit squares, and let a simple connected rectilinear path be drawn along edges of the squares. Any square with one or more edges on the path can be used to define a modification to the path by forcing it to travel around the square along the edges that are not currently on the path. The result depends only on the number of the square's edges that lay on the path initially (we forbid modifications that would produce paths with loops or self-intersections). The examples in Figure 2 summarize the possible local modifications.

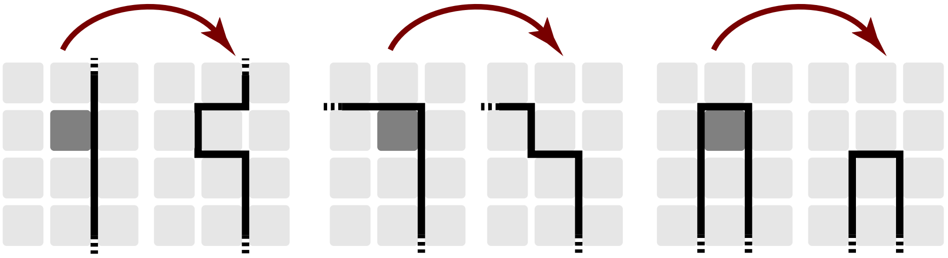

Figure 2: *The three possible modifications induced by a square adjacent to a path along the edges of a square grid. The left diagram in each pair shows the initial curve and highlights the square that defines the modification. The right diagram shows the modified curve.*

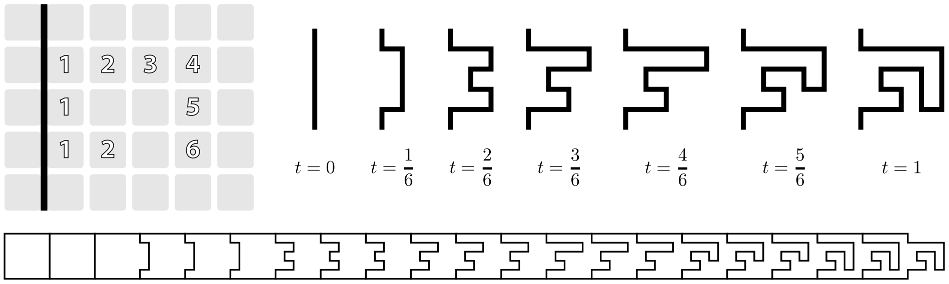

Figure 3: *A demonstration of the development of a grid-based evolving path. The initial straight path is shown with a square grid on the left. Grid squares are numbered to indicate the order in which their edits are to be applied to the initial curve (squares with the same number can be applied in any legal order). The sequence on the right shows all the paths that result from this numbering. The row of tiles at the bottom shows a use of this curve for two edges of a sample prototile of type* IH41*.*

It is now possible to construct an evolving path via a sequence of such modifications. An initially straight path P_0 is placed along n consecutive edges in a square grid. We then choose a sequence of squares $\{s_1, \ldots, s_m\}$, with the constraint that s_k be adjacent to the path obtained by applying the modifications defined by $s_1, \ldots, s_{k-1}$. As long as the path never develops loops or self-intersections, the result is m new paths $P_1, \ldots, P_m$ evolving out of a straight line segment. To evaluate this curve at time t, we simply return $P_{\lfloor mt \rfloor}$.

In practice we can offer greater flexibility and expressive power. Most importantly, several squares can be grouped together so that their modifications are applied "at the same time" (relative to t). Figure 3 shows an evolving curve with this additional property. At the bottom of the illustration, each of the six stages of the path's evolution is repeated three times. Note also that a single square can be visited more than once in a sequence of modifications; a second visit will simply reverse the effect of the first.

I have created an interactive tool in which an $n \times n$ square isohedral prototile is embedded in a square grid. The designer selects grid squares in any order, and can preview a parquet deformation based on the resulting paths. The tool restricts the designer to legal squares (those adjacent to the current prototile boundary, and which would not lead to an illegal tile shape). The system automatically propagates any local edit to all of the locations on the tile's boundary that are related by considerations of symmetry and tileability.

The resulting parquet deformations have a mechanical, rectilinear appearance. For me, the most attractive results were obtained by continuing to modify the prototile's boundary until the interior of the tile did not contain any 2×2 grid of squares. Figure 4 shows a simple example based on isohedral type IH41; more examples are in Figure 5.

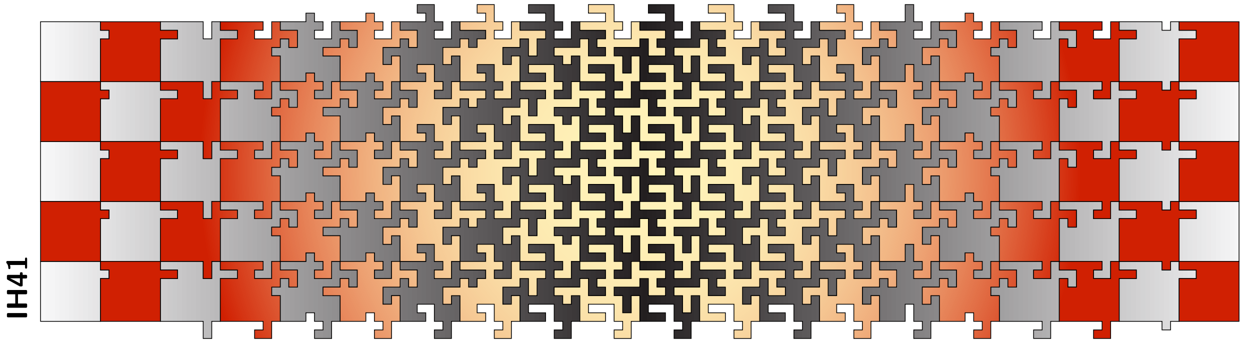

Figure 4: *An example of a parquet deformation constructed by discrete modification of grid-based paths. In this case, the time parameter is controlled by a "tent" function that is 1 in the centre of the design and 0 at the edges. The final geometry in this drawing (and others in the paper) was edited in Adobe Illustrator to adjust line widths and fill tiles with colour gradients.*

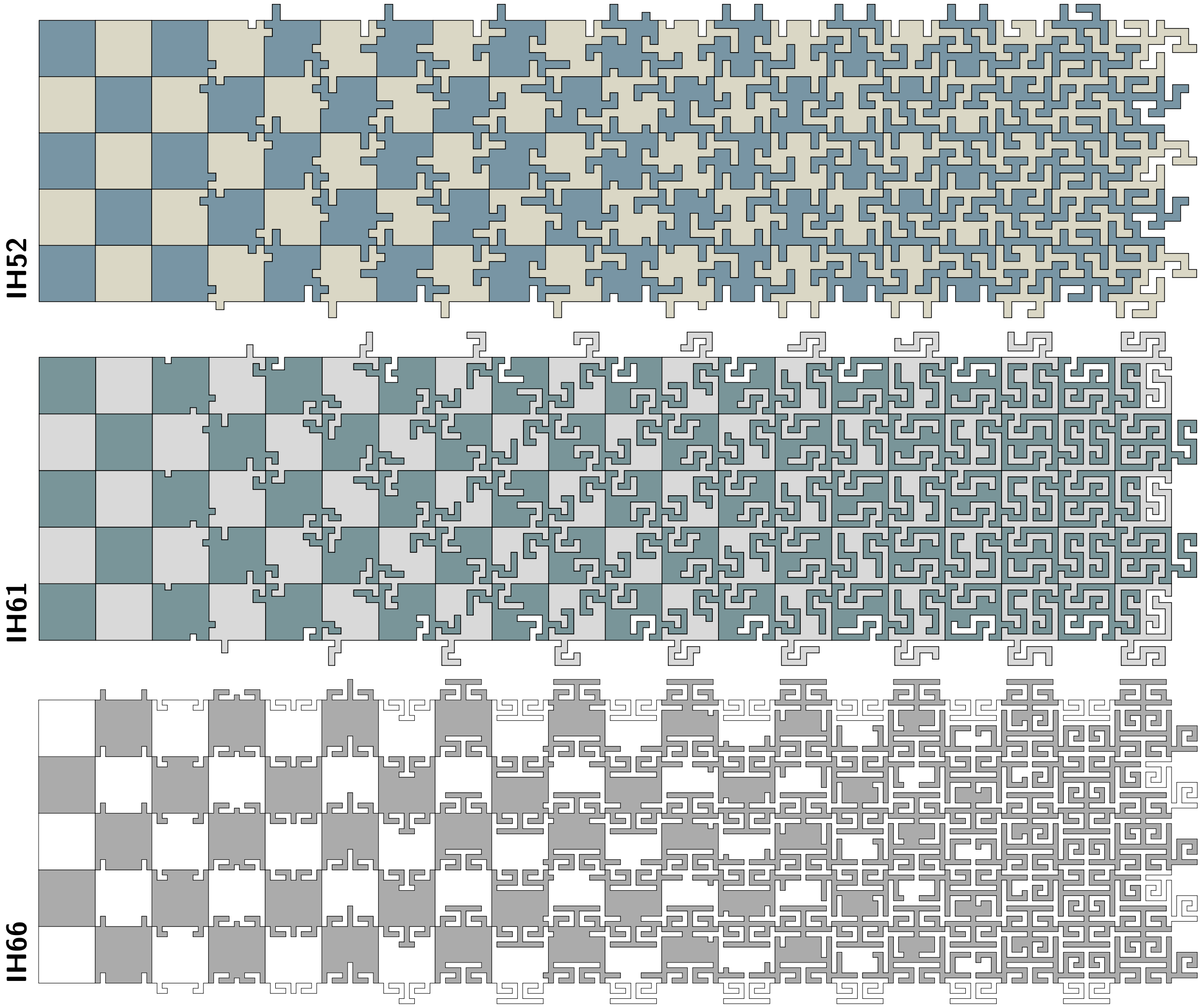

Figure 5: *More examples of parquet deformations constructed via discrete modifications of square grid paths.*

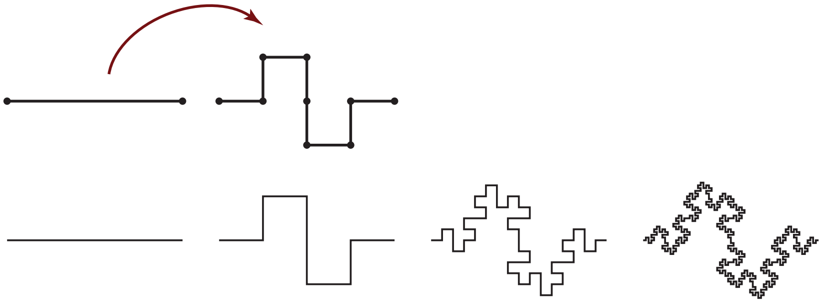

Figure 6 : *A simple iterated function system. The top row shows the rule for replacing a line segment with a path made out of eight equal-length segments. The bottom row shows the paths that result after zero, one, two and three applications of this rule. This rule is used to construct the top parquet deformation in Figure 7.*

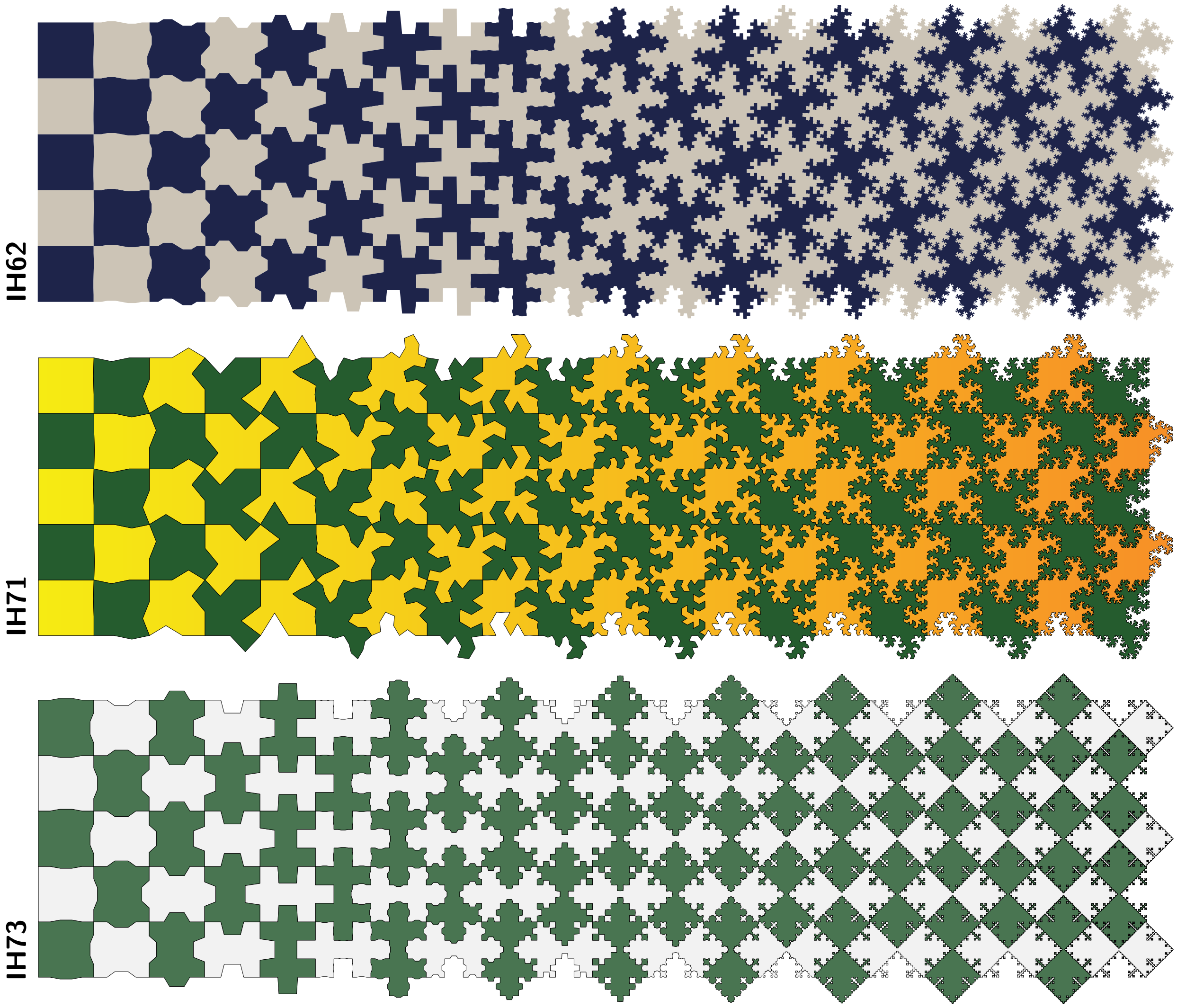

Figure 7 : *Three examples of parquet deformations based on linear interpolation between successive generations of iterated function systems.*

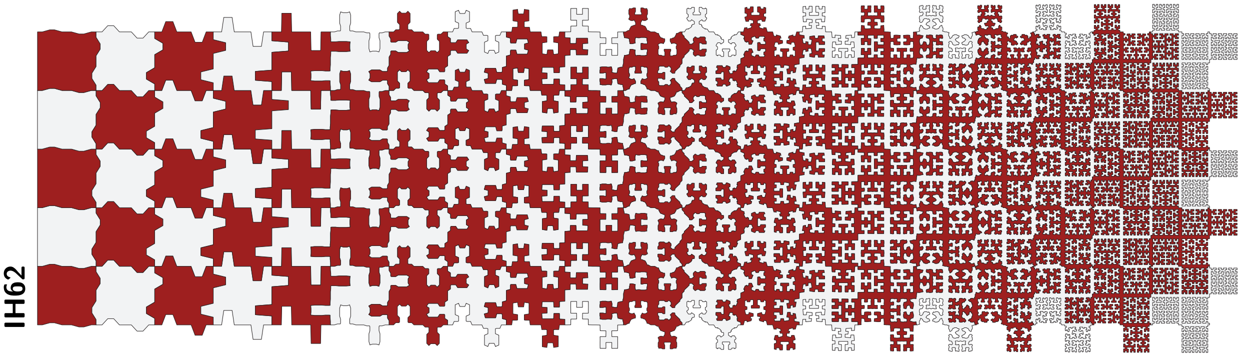

Figure 8 : *A parquet deformation based on steps in the generation of a Hilbert curve.*

4 Iterated function systems

A popular and attractive technique for evolving a curve is to use an iterated function system (IFS), which adds progressively finer detail to a curve, producing a fractal in the limit. In this context, an IFS is simply a piecewise-linear path from $(0,0)$ to $(1,0)$. The path can be elaborated to any number of generations by repeatedly replacing each of its segments with a suitably transformed copy of the entire path (see Figure 6). The transformation can optionally incorporate a reflection across the segment, or the replacement can be suppressed for that segment altogether.

It seems natural at this point to take a sequence of paths, each constructed via replacement from its predecessor, and use them as discrete steps in a single evolved curve, as in the previous section. However, I found that the transition between generations in an IFS is far too abrupt to produce an aesthetically pleasing parquet deformation. A simple compromise achieves a satisfactory result: we space the sequence of paths evenly from $t = 0$ to $t = 1$, and linearly interpolate between adjacent keyframes for intermediate values of t. Although the results from linear interpolation are acceptable, there likely are interpolation algorithms that are better suited to this particular task. A more intelligent algorithm would take into account the local relationship between a segment and its “descendants” in future generations, and for any value of t might incorporate exponentially decreasing detail from several (or all) future generations.

Figure 7 shows three examples of parquet deformations generated by assigning a single iterated function system to all the edges in a prototile. Figure 8 shows an additional example based on a Hilbert curve; this required additional effort, since its replacement rule is not as simple as those of ordinary iterated function systems.

5 Organic labyrinthine curves

In the context of constructing labyrinths and mazes, Pedersen and Singh developed a geometric simulation that evolves simple initial curves into convoluted organic forms [10]. In their algorithm, points on a curve undergo Brownian motion and geometric smoothing, and are subject to physical attraction and repulsion forces from other points. These forces, inspired by the Lennard-Jones potential in chemistry, cause nearby points to seek a “rest distance” in which attraction and repulsion cancel out. This algorithm produces curves reminiscent of the shapes in TSP Art [8] and Hart’s Growth Forms [3], though Pedersen and Singh offer a large number of (spatially varying) tuning parameters that can produce more distinctive results.

I have adapted Pedersen and Singh’s algorithm to produce isohedral prototiles with labyrinthine boundaries. The prototile boundary evolves according to their simulation, with some important exceptions. First, the

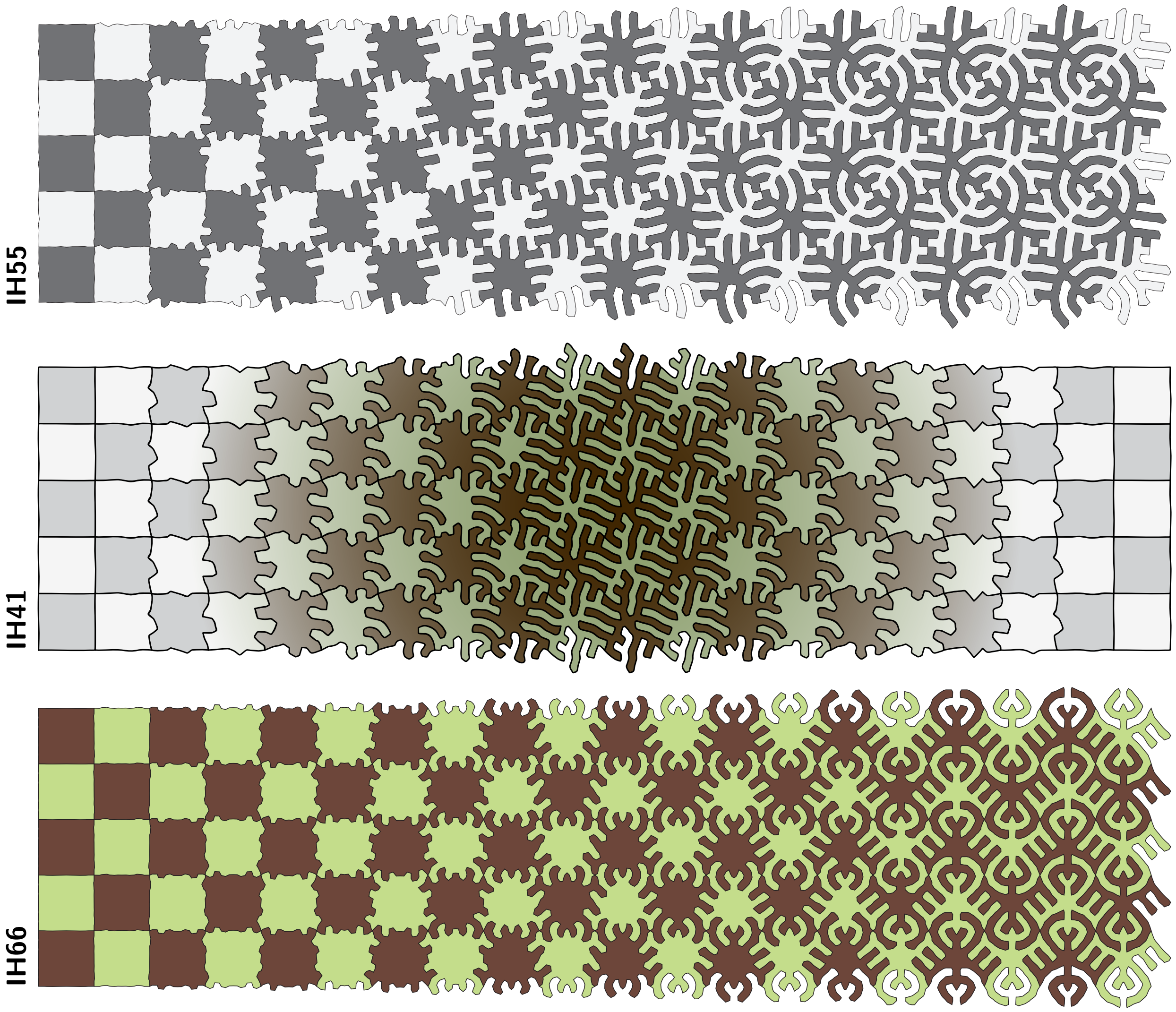

Figure 9 : *Three examples of parquet deformations based on the organic labyrinth growth algorithm of Pedersen and Singh.*

corners of the square must remain fixed, as these will become tiling vertices in the final parquet deformation. Second, the boundary contains redundant information because of matching edges and internal tile symmetries. The new position of a point on the tile boundary must take into account attraction and repulsion forces exerted not just on that point, but on all its redundant copies. These separate forces can be added together to obtain a final displacement for the single shared representation of a point and its copies.

The simulation is permitted to run until a desired final form is reached (typically a few hundred to a few thousand iterations, based on a timestep that is sufficiently small to avoid numerical instability). A snapshot is saved of every tile outline created during the simulation. Once the simulation is complete, a subset of the snapshots, evenly spaced in time, is extracted and used as keyframes. Because of the inherent smoothness of the simulation, no interpolation is necessary between keyframes.

Three examples of organic parquet deformations are shown in Figure 9.

6 Conclusions

This paper explores a small slice of the design space of parquet deformations. I have deliberately curtailed the degrees of freedom available in the tiling that acts as the scaffolding for the design, and concentrated on three independent algorithms for evolution of curves in this context. Even within this domain a wide range of aesthetic opportunities can be uncovered, suggesting that the larger world of parquet deformations can serve as an inexhaustable source of inspiration for art.

The software framework described in this paper can be extended in many natural ways. The most obvious path for future work would be to explore other algorithms for curve evolution or transformation. In computer graphics, linear interpolation of curves has long been understood to produce unattractive results. More principled approaches such as that of Sederberg et al. [11] or as-rigid-as-possible methods [1] might be more robust and more aesthetically pleasing. I would also like to experiment with curve simplification algorithms such as the standard Douglas-Peucker method [2] or the more recent abstraction technique of Mi et al. [9].

Given a richer toolkit of curve evolution strategies, it should also be possible to juxtapose multiple algorithms spatially (on different edges of a single tile) or temporally (in sequence on a single edge, over time).

References

[1] Marc Alexa, Daniel Cohen-Or, and David Levin. As-rigid-as-possible shape interpolation. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 157–164, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[2] David Douglas and Thomas Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *The Canadian Cartographer*, 10(2):112–122, 1973.

[3] George Hart. Growth forms. In Craig S. Kaplan and Reza Sarhangi, editors, *Proceedings of Bridges 2009: Mathematics, Music, Art, Architecture, Culture*, pages 207–214. InType Libra, 2009.

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

[5] William S. Huff. The Parquet Deformations, from the Basic Design Studio of William S. Huff at Carnegie-Mellon University, Hochschule für Gestaltung and State University of New York at Buffalo from 1960 to 1980. Unpublished.

[6] Craig S. Kaplan. Metamorphosis in Escher's art. In *Bridges 2008: Mathematical Connections in Art, Music and Science*, pages 39–46, 2008.

[7] Craig S. Kaplan. *Introductory Tiling Theory for Computer Graphics*. Morgan & Claypool, 2009.

[8] Craig S. Kaplan and Robert Bosch. TSP art. In *Bridges 2005: Mathematical Connections in Art, Music and Science*, pages 301–308, 2005.

[9] Xiaofeng Mi, Doug DeCarlo, and Matthew Stone. Abstraction of 2d shapes in terms of parts. In *NPAR '09: Proceedings of the 7th International Symposium on Non-Photorealistic Animation and Rendering*, pages 15–24, New York, NY, USA, 2009. ACM.

[10] Hans Pedersen and Karan Singh. Organic labyrinths and mazes. In *NPAR '06: Proceedings of the 4th international symposium on Non-photorealistic animation and rendering*, pages 79–86. ACM Press, 2006.

[11] Thomas W. Sederberg, Peisheng Gao, Guojin Wang, and Hong Mu. 2D shape blending: An intrinsic solution to the vertex path problem. In James T. Kajiya, editor, *Computer Graphics (SIGGRAPH '93 Proceedings)*, volume 27, pages 15–18, August 1993.