

Computer Generated Celtic Canvas

By

Richard Brian Doyle

B.S.E.E.C.S, University of Colorado, 1975

M.S. University of Colorado, 2002

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado at Colorado Springs

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Computer Science

2008

© Copyright by Richard B. Doyle 2008.

Permission to make digital or hard copies of all or parts of this work for personal or classroom use is granted without fee provided that copies are not distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page.

This thesis entitled:
Computer Generated Celtic Canvas
written by Richard Brian Doyle
has been approved for the Department of Computer Science

Sudhanshu K. Semwal

Valerie A. Brodar

Keith L. Phillips

Charles M. Shub

Robert S. Wiener

Date _____

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentations standards of scholarly work in the above mentioned discipline.

HRC protocol # _____

Abstract

Doyle, Richard Brian (M.S., Computer Science, B.S., Electrical Engineering and Computer Science)

Computer Generated Celtic Canvas

Thesis directed by Professor Sudhanshu K. Semwal

This thesis investigates the application of complex systems methods in Celtic design. The issues of interest are aesthetic characteristics of lines seen in insular manuscripts from the 9th century and how to more closely approach them without inhibiting the creation of new work. An extensive literature review evaluates all related areas of research and reveals complex systems theory as a rare approach that has not been extensively applied to these problems.

Complex systems theory is normally associated with the study of the evolution of patterns in chaotic systems. This thesis describes a number of novel ways in which the theory can be applied to problems that are highly structured at the outset, adapted, and then finished by hand. These begin with the introduction of a robust model of curvature dynamics and genetic evolution of space curves as a foundation for line drawing and structural anatomy. Linear cellular arrays are developed to handle the complexity of integrating lines and the parameters of design, including curves with structure and crossings states, occlusion processing, depth cues, and line weighting. Lattice gas automata are developed to assist with exploration and spontaneity. We include the use

of original line art as specifications for design elements. Finally, we demonstrate the application of linear genetic programming as a means of handling the complexity of specifying, assembling, and manipulating designs under this level complexity.

The ideas are tested against examples of knotwork, plant, and animal designs from the book of Kells, the Lindisfarne gospels, and drawings from the studies of George Bain.

Acknowledgements

I would like to express my deepest appreciation for the loving support of my mother, Ruth E. Doyle, and my father, Douglas J. Doyle. Although they are not with us as this work is finished, they were my greatest examples and inspiration.

As I applied to work on this degree, my father was diagnosed with Alzheimer's disease and did not survive to see the work completed. My heart goes out to everyone who chooses to make the commitment and endure the trials of caring for a family member. My thoughts are with every researcher who has taken up the challenge to find the cure that will one day liberate our futures from this devastating disease. My gratitude goes out to all those physicians who compassionately share the struggle with their patients and their families.

I wish to thank the department and my committee for their steadfast encouragement and support.

I wish to thank my advisor, Dr. Sudhanshu Semwal for his unwavering faith in the face of the many starts and for the many discussions and thoughtful insights into the project.

I wish to thank Fr. Luke Sheffer and Mr. Ron Adair for their perspectives on the many varied and sometimes unexpected dimensions of this research.

Contents

Abstract	iv
Acknowledgements	vi
Chapter 1 Introduction.....	1
1.1 Iona	1
1.2 Reprise and Rediscovery.....	6
1.3 Hypothesis	12
1.4 Contributions of this thesis	16
1.5 Organization of the thesis	17
Chapter 2 Related Work.....	19
2.1 Introduction.....	19
2.2 Production of an insular manuscript	20
2.3 Iconography	23
2.4 Celtic Symbolism	24
2.5 Archeological design constraints.....	25
2.6 The modern methods of construction.....	33
2.6.1 The theory of plaits	33

2.6.2	A theory of points.....	35
2.6.3	A theory of grammars of elementary forms	36
2.6.4	A theory of reflections	38
2.6.5	A theory of topological assembly.....	39
2.7	Computer Generated Construction	42
2.8	Composition and creative design theory.....	48
2.9	Mathematical foundations.....	56
2.10	Complex systems theory	64
2.10.1	Complex Adaptive Systems	66
2.10.2	Cellular automata	67
1.10.2.1	Graph cellular automata	68
1.10.2.2	Lattice gas automata	69
2.10.3	Agents.....	70
2.10.4	Lindenmayer-Systems (L-systems)	72
2.10.5	Artificial Life	72
2.10.6	Social agency	74
2.11	Differential geometry.....	75
2.11.1	Curvature Dynamics	78
2.11.2	Line Drawing	79
2.12	Noise.....	81
2.12.1	Mean Zero Brownian bridges (line directional variation).....	81
2.12.2	Perlin Noise (space filling).....	82
2.13	Characteristics of the hand drawn line.....	84
2.14	Instruments and media	87
2.15	Art criticism	90
2.16	Summary	92

Chapter 3 Adaptation	94
3.1 Introduction.....	94
3.2 The modeling of lines.....	95
3.3 Curvature dynamics	98
3.4 Linear genetic programming and evolvable curves	104
3.5 Structure or the problem of grids	106
Analyzing the drawing area.....	106
Automatic grids.....	110
3.6 Posing and spontaneity.....	115
3.6.1 Perlin noise.....	116
3.6.2 Mirror curves as lattice gas automata	125
3.6.3 Mercat's labeling algorithm as lattice gas automata	129
3.7 Summary.....	131
Chapter 4 Overview of the System	133
4.1 Introduction.....	133
4.2 Organization	135
4.3 Phenotypes	136
4.4 Genotypes.....	136
4.5 Genetic algorithms and complex systems methods	137
4.6 User interaction.....	138
4.7 Evaluation.....	139
4.8 Summary.....	140
Chapter 5 Interactions Representations and Processes.....	141

5.1	Introduction.....	141
5.2	Structure.....	141
5.3	Forms and sleeves.....	146
5.4	Guide curves.....	161
5.5	Form solving and posing.....	174
5.6	Rendering.....	179
5.7	Summary.....	181
Chapter 6 Results and Conclusions.....		183
6.1	Introduction.....	183
6.2	Celtic spiral capability demonstration.....	184
6.3	Lindisfarne Gospels spiral.....	184
6.4	Knotwork.....	185
6.5	Lindisfarne knotwork (Adcock).....	187
6.6	Lozenge of MS 58 f 34r.....	187
6.3.1	Reptile (snake).....	188
6.3.2	Hairy male human.....	190
6.3.3	Beast.....	191
6.3.4	Bird.....	191
6.7	Plant.....	192
6.8	Summary.....	192
6.9	Triple spiral example code.....	193
Chapter 7 Suggestions for Further Research.....		198
7.1	Introduction.....	198

7.2	Summary of system capabilities	198
7.3	Further work.....	201
7.4	Summary.....	202
	References.....	203

Tables

Table 1: One level in the grid	146
Table 2: Linear genetic programming model operators.....	153
Table 3: Form or zoomorph design data structure.....	155
Table 4: Extracted design curve data	156
Table 5: Form anatomy image data	156
Table 6: Form symmetry image data.....	157
Table 7: Symmetry specification (left) and skeletal design data.....	158
Table 8: Segmentation specification data	159
Table 9: Study curve design data	164
Table 10: Initial curve segment template names	164
Table 11: Study curve assembly sequence	167
Table 12: Space curve with assigned curvature	169
Table 13: Segment from space curve interval table	170
Table 14 : Motion event data	170
Table 15: Solved guide curve	172
Table 16: Anatomy object for snake stroke	176

Figures

Figure 1: Lozenge of MS 58 34r, George Bain (1951) left, Board of Trinity College Dublin (2007), full size, right	15
Figure 2: Squaring a circle	27
Figure 3: The frame, circle, and cross of the Soiscel Molaise (Stevick, 1994, pp. 6-7)	28
Figure 4: Lindisfarne f26 without chevrons (knotwork, left, subdivided 11 x 11 grid, right)	29
Figure 5: Dimensions of the symmetrical loop	29
Figure 6: Dimensions of the u-bend.....	30
Figure 7: Simple plait (left), interlaced plait (right) (Allen, 1903, p. 144)	34
Figure 8: Left hand crossing (C to D), vertical break, horizontal break, right hand crossing (B to A) (Allen, 1903, p. 145).....	34
Figure 9: Elements of J. Romilly Allen with breaks (Adcock, 1974).....	35
Figure 10: Theory of points.....	36
Figure 11: Tiles joined on corner crossing constraints (Sloss, 2002)	37
Figure 12: Andy Sloss' approach, 3x4 panel (Doyle, 2005).....	38
Figure 13: Mirror curves	39
Figure 14: Mercat's trefoil and its tetravalent graph	40
Figure 15: Steve Abbott (1998) with Christian Mercat's approach.....	43
Figure 16: Stages in construction (Christie, 1929).....	44

Figure 17: Square plait derivation (Drewes, 2000)	46
Figure 18: Aidan Meehan's tri-grid (1991, 2003).....	47
Figure 19: Semi-formal structure	53
Figure 20: Golden rectangle	54
Figure 21: Celtic spiral formed using the golden ratio.....	55
Figure 22: Tangle without alternating crossings (Thurston, Fig. 10).....	56
Figure 23: Tangle with odd number of self-intersections	57
Figure 24: Tangle with odd number of self-intersections, crossed	57
Figure 25: A simple shape rule	61
Figure 26: Overhand knot (actually a tangle--black is below, white is above)	62
Figure 27: Rules of the HPP Lattice gas model, possible velocity directions (a) and possible collisions (b)	69
Figure 28: Agent Architecture	70
Figure 29: Frenet Frame	77
Figure 30: Inflated cord tile with VHDH quadrant coding (Sloss, 1995).....	80
Figure 31: Doubling knots (Bain, 1951, p. 46)	81
Figure 32: A Perlin noise image (left) and its absolute value (right)	83
Figure 33: Hermite cubic spline as half impulse	99
Figure 34: curvature impulses and their decomposition.....	101
Figure 35: schedule space curves with assigned curvature	102
Figure 36: curvature rate impulses	102
Figure 37: Space curves with scheduled curvature rates.....	103
Figure 38: curvature acceleration impulses	103
Figure 39: Space curves with scheduled curvature accelerations.....	104
Figure 40: Segmented planar space curve with assigned curvature dynamics.....	105

Figure 41: Chain code directions	108
Figure 42: automatic mesh fitting trajectories	111
Figure 43: Euclidean distance function (left) contours (right)	112
Figure 44: Kohonen grid weight distribution	113
Figure 45: A Perlin noise image (left) and its absolute value (right)	116
Figure 46: Perlin noise field grid cell.....	117
Figure 47: Perlin height field after dot products.....	118
Figure 48: Perlin blending curve	118
Figure 49: A Perlin noise image (left) and its absolute value (right)	119
Figure 50: Cross in circle.....	120
Figure 51: Single knot three cord braided border.....	126
Figure 52: Simple knot.....	126
Figure 53: Derivation of mirror curve moves.....	127
Figure 54: Example design grid	130
Figure 55: Top level system view	135
Figure 56: Genetic algorithm	138
Figure 57: Structure load and editing work flow	142
Figure 58: Use interaction for matte processing and grid design.....	144
Figure 59: Hierarchical grid table	145
Figure 60: Form load path (left) Sleeve load path (right)	147
Figure 61: Stroke extraction from anatomy file.....	148
Figure 62: Extracted features of the Celtic snake	149
Figure 63: Snake skeletal geometry.....	154
Figure 64: User interaction for form and sleeve loading.....	160
Figure 65: Work flow paths for guide curve sketching (Mercat interlace solver)	162

Figure 66: Study curve example	162
Figure 67: Guide curve generation for regular grids (left) irregular and freeform (right)	163
Figure 68: Guide curve without aesthetic enhancement	165
Figure 69: Guide curve with aesthetic enhancement.....	166
Figure 70 : Guide curve phenotype construction (without Mercat).....	168
Figure 71: Interlace solution (left) distance weights (right).....	173
Figure 72: Form selection in design panel.....	174
Figure 73: For selection dialogs.....	175
Figure 74: Phenotype of form in default display	177
Figure 75: Form symmetry binding dialog	178
Figure 76: Celtic snake in new pose.....	179
Figure 77: Render panel	180
Figure 78: Final rendering	181
Figure 79: Celtic Spiral.....	183
Figure 80: Triple spiral MS 58 f 34r (Board of Trinity College Dublin)	184
Figure 81: Lindisfarne spiral studied by Bruce-Mitford (Alexander, 1978, p. 11)	185
Figure 82: Lindisfarne knotwork folio 11 (Bain, 1992)	186
Figure 83: Lindisfarne knotwork folio 11 (Bain, 1992)	186
Figure 84: Lindisfarne folio 26 border (Adcock, 2002).....	187
Figure 85: Design files for the reptile.....	189
Figure 86: Rendering of a Posed reptile.....	189
Figure 87: Tangle of reptiles.....	190
Figure 88: Design files for a hairy human.....	190
Figure 89: Rendering of hairy male	190

Figure 90: Design files for a beast.....	191
Figure 91: Rendering of a beast.....	191
Figure 92: Design files for a bird	191
Figure 93: Rendering of a bird	191
Figure 94: Design files for a plant.....	192
Figure 95: Rendering of a plant.....	192

Chapter 1 Introduction

Iona ... a place where the "veil is thin" ...

Rosemary Power, 2006, 45

1.1 Iona

With these words, Rosemary Powers (2006) takes us into the impressions of George MacLoed as he is photographed writing in the library of the Abbey on Iona in 1943. Not long before the islanders had observed services in the ruins of this church, continuing a practice from previous centuries. In 1938 MacLoed had founded the most recent community on this small island off the coast of Scotland. Under his guidance, interrupted only by World War II, restoration of the Abbey was completed in the 1960's as a place to better equip the Church of Scotland to care for those in poverty. As with another beginning in the 6th century, Iona was again a place of learning with a mission.

What is known of medieval Iona has been gleaned from Irish Annals (Ó Corráin, 1998) fragmentary Scottish records (Driscoll, 1998), the writings of his biographer, Adomnan (Reeves, 1874), and archeological excavation of the island near the Abbey and the Nunnery (Sullivan, 1994). Iona was a place of pilgrimage, a place of high honor in Scotland, as a burial ground of kings (Faerber, 2004). MacLoed's choice of the word

“thin” finds its origin in Greek and Russian orthodox traditions being assigned to places where the material and the spiritual are only finely separated (Powers, 2006).

Here, in the year c 806, Vikings raided a vital and sizable monastic community. This was the third time, following earlier raids in c 795 and again in c 802 when it was burned. While Iona and nearby Mull were large enough to support the community, its geographical location left it vulnerable. On this occasion, the lives of sixty-eight members of the community were lost. The following year a majority of the survivors sought refuge in Kells granted them three years earlier, built a new monastery, and resumed a period of calm by c 814. St. Blathmac refused to leave Iona and was killed in yet another raid c 825 after which the monastery was mostly abandoned. The monastery in Kells remained active into the 12th century (Alexander 1978, Meehan 1994, Ó Corráin 1998).

The community on Iona was founded in c 563 by the Irish Prince-Abbot, St. Columba (Colum-cille, old Irish, meaning, “dove of the churches”), the island granted him by a relative, King Conall of Dal Riada. St. Columba, born in Donegal c 521, began his ministry in Ireland. While there remains uncertainty as to the exact chronology, by the time of the settlement in Iona, he was well established with monasteries or houses founded in Derry, Durrow, and Kells. This new mission was to minister to Irish settlers in Western Scotland and evangelize the Picts (painted people) who inhabited the region (Edmonds 1910, Meehan, 1996).

Iona became a center of learning. A library was established; and monks began the practice of writing books, as well, as gathering, copying, and preserving manuscripts. While much of the work involved copying Bibles, Psalters, and Gospels to support missionaries, there is evidence that monasteries also engaged in the preservation of history, law, science, and medicine, in other words, a cultural memory of civilization.

The work continued after the passing of Colum Cille in 597. At the request of King Oswald in 635, St. Aidan journeyed From Iona into Northumbria and founded the monastery at Lindisfarne (Edmonds 1910, Meyvaert 1989, Hayes 2005). Lindisfarne established schools comparable to those in Europe and began producing scholars. From these beginnings Iona grew in influence to head a family of monasteries stretching from Ireland to the East coast of Britain (Meehan, 1994).

From these monasteries emerged illuminated manuscripts in a style referred to as Insular. Characteristics of the style include the following (Alexander, 1978):

- An abstract decorative vocabulary
- Alternating interlace framed in the page
- Intricate kinetic forms
- Relational ambiguity
- Interpretations of natural forms as patterns
- Trumpet patterns (complex geometric spirals)
- Key patterns
- Bird interlace
- Carpet pages where form and ground are differentiated by contrasting fill patterns
- Dots incorporated into initials and ornament
- Historiated initials (pictures as fills)
- Seamless merging of script and illumination
- Insular half-uncial (mixed cases), majuscule (upper case), and minuscule (lower case) script instead of Roman uncial

- Paragraphs or sections beginning with a tall decorated initial followed by letters whose height decreases to that of the ruling over one or more lines, diminuendo (Werner 1990, Meehan 1994).

The earliest illuminated manuscript to survive is *The Cathach* Psalter c 600 (RIA, MS 12 R 33), where modestly decorated initials were penned to introduce Psalms. The first fully illuminated manuscript, the Book of Durrow, followed in c 650. This holds the symbols of the evangelists, carpet pages, and Canon tables, which we later see elaborated (Meehan, 1996). The next milestone, the Lindisfarne Gospels, was produced between c 710-725. This is a large format manuscript designed for altar use and most likely written in celebration of St. Cuthbert. Designed or commissioned by Bishop Eadfrith, its carpet pages exhibit influences of symmetry, reflection, rotation, and repetition familiar to contemporary design. Portions of these designs were so meticulously planned and carefully executed as to appear mechanical. That the decoration is unfinished may be associated with Eadfrith's passing in 721 (Swenson 1978, Meehan 1994, Brown 2004).

Interspersed among these works are others known by their scribes, including The Book of Durham, the Echternach and Maihingen Gospels, the Lichfield Gospels, a continental Gospel at St. Gall, the Rushworth Gospels and the Book of Armagh (Henry, 1974).

Then we encounter the Book of Kells, whose origin is entirely inferred. Bernard Meehan (1994, 9) relates that it is difficult to write to about this manuscript "without resorting to hyperbole." He continues

It is the most lavishly decorated of a series of gospel manuscripts produced between the 7th and 9th centuries.

The Annals of Ulster contains the following phrase in its record of the theft and recovery of a book and its cover from the Church of Kells in 1007 (Meehan, 1994):

... the most precious object in the western world ...

Alexander (1978, 73) expresses the following:

It has become one of the most famous of all medieval illuminated manuscripts and for many is the supreme achievement of Insular art.

The Book of Kells is also a large format manuscript. It was likely written in celebration of St. Columba, perhaps for the bicentennial of his passing (Henry, 1974, 221). The Book of Kells stands apart in its subtlety, complexity, and esoteric character. Its crosses are those of Iona, some forms indigenous, Pictish, other elements perhaps drawn from Lindisfarne, Canterbury, and as far away as the continent. It was guided by the vision of three and perhaps four studied scribes and artists (Henry, 1974, 217-218).

These communities were gifted with abundant resources. The vellum on which the pages of both the Book of Durrow and the Book of Kells were rendered is prepared calfskin. More than 185 animals would have been needed for the Book of Kells alone. Among its pigments is found an ultra marine made from Lapis Lazuli, whose only origin was Afghanistan, and a red made from a pregnant Mediterranean insect (Meehan, 1994, 86-88).

Property charters, copied onto what were blank pages, place the book in the Kells monastery in the 12th century. It was likely in progress as Viking began raiding

Iona and accompanied survivors to Kells as they evacuated the community. The Abbot of Iona, who may have commissioned the work, was killed in the raid of 802.

Illumination in The Book of Kells is also unfinished; and given the enormous size of the project, the interruptions, dislocation, and loss of life during this turbulent period, this is understandable.

As for other monasteries, little more is known of Durrow. The Annals of Ulster record that its library was burned in 1095 (Meehan, 1996). Vikings raided the Lindisfarne monastery in 793 leaving as few as two or three survivors. Evacuation of artifacts began around 830 as Vikings established a permanent presence in the region; and it was abandoned by 875 (Brown, 2003, 85-87).

Having followed this story, we are left with some sense of the gathering and demise of a most extraordinary culture and its art form “steeped in mystery for some thousand years” (Bain, 1992, 6). We are also left with some sense as to the artistic values, motivations, and the people brought together to realize them.

1.2 Reprise and Rediscovery

Original designs were considered impossible and were rarely attempted. Adaptation by copying was the only use.

George Bain, 1951, 15

The journey to the rediscovery of Celtic design begins with J. Romilly Allen’s (1903) survey, cataloging, and analysis of designs found on stone sculpture in Scotland. His analysis reflected on the principals of ornament; and he introduced a theory of plaits as a means of describing the evolution of many of the patterns he recorded. His work

was followed by the studies of George Bain (1951) who was interested in rediscovering methods of construction and motivated in part by the despair expressed by his teachers when confronted with the complexity of Celtic design. Inspired by these two figures, Gwenda Adcock (1974) carried out the first comprehensive survey of Celtic interlace in Northumbria (Northern England, 500-1100). As part of her study, she developed the system that is now used for the archeological description and analysis of Celtic interlace. Most notable among her accomplishments is the distillation of the extensive cataloging of interlace designs in J. Romilly Allen's survey to descriptions involving a vocabulary of nine standard patterns that describe the paths of cords (forms) as they pass through the rectangular grids (structures) proposed to organize them. Iain Bain (1986), an engineer and the son of George Bain, would describe this method of description as a grammar in his study of interlace. Iain Bain was drawn to his father's work to provide additional structure. Comments of many readers revealed a blurring of where method ended and his father's training and insight as an artist would take over. His analysis would transform many of his father's techniques into concise procedures on grids that are recognizable as algorithms. Peter Cromwell (1993) follows with his examination of frieze designs in Celtic interlace. He was the first to carefully work through the modification of these formal methods that preserve the alternating of cords (alternating knots) under transformation. He is credited with the idea that the paths traced by interlace on regular grids can be described in terms of reflections from the sides of the bounding structure. This idea would be developed further as a theory of mirror curves (Jablan 1995, Gerdes, 1997). Something more that followed from Gwenda Adcock's research was the recognition that sculptors used templates to transfer designs to stone and that most of these designs were organized on grids, with exceptions where work was done freehand. Andy Sloss (1995) developed this idea with the development of a set of

low-level tilable shapes whose assembly is guided by labeling. This was the realization of primitive elementary forms applied as a labeled shape grammar, whose theory was developed for rectangular shapes by George Stiny (1975) and started for curved shapes by Iestyn Jowers (2006). While this approach proved to be an eminently workable solution for typesetting, it provided a practical illustration of another of Gwenda Adcock observations that without a skilled artist to blend them, the simple reuse of standard templates can lead to monotonous patterns. Following these developments is a contribution by Christian Mercat (1997) that arises from the alternating cords in Celtic interlace. He observed that crossings (tangle operators) comprise an encoding of the over-under constraint and that this constraint can be annotated on the edges of a planar graph. By arguing how planar graphs create colored subdivisions of the plane, he finds a general method of specifying arbitrarily complex interlace. One further observes that these graphs and their encapsulations correspond to design grids. His method occurs as a natural consequence of his thesis (1998) contributing to the development of a discrete form of manifold theory for use in quantum mechanics. Finally, Frank Drewes (2000) introduces graph grammars as means of guiding the synthesis and assembly of knotwork based on tiling.

A survey of the web reveals that the general availability of availability of personal computers along with excellent graphics now modeled on OpenGL has lead to a growing interest in software applied to the design of Celtic knotwork. By means of these or vector-based drawing tools (Hull, 2003) a considerable body of contemporary work has evolved; and one far reaching index compiled by Peter Suber (2005) indicates that popular design resources for knots, Celtic and otherwise, have become prodigious.

Notable among those who have devoted time to computer generated Celtic design (the Knot Assistant) is Andrew Glassner (1999a, 1999b, 2000). Techniques

applied include Peter Cromwell's reflection method (mirror curves) on Iain Bain grids, citing also the considerable body of design reference materials developed by Aidan Meehan (1991, 2003), as well, as modular assembly of tiles (puzzle pieces). He is also thoughtful to provide parameters for the design of curves for realizing the geometry. Perhaps his most insightful offering is that he uses software to develop sketches as he finds (1999a) that

... getting the particular curves and patterns in a design to look just right is easier with pen and ink than mouse and keyboard ...

Steve Abbot (1998) developed a program (Knots3D) to illustrate Christian Mercat's method. A fundamental shift occurred the same year with the landmark paper by Michael Wong, et. al., (1998) introducing the world to computer generated ornament. Here we saw the use of a rule based divide and conquer strategy that implemented a small set of classical design principals. A similar milestone was realized shortly thereafter by Matthew Kaplan and Elain Cohen (2003) in their *tour de force* in computer generated Celtic design. Their implementation included Christian Mercat's specification method with artist interaction, parametric curves with new smoothing and cord inflation methods, and a means of integrating imagery to allow the inclusion of complex animal designs (zoomorphs).

For their demonstration Kaplan and Cohen include a broad range of complex designs including a partial reconstructions of the carpet page (f. 94v) from the Book of Mark in the *Lindisfarne Gospels* and the T from f. 124r in the *Book of Kells*. At this point a natural conclusion would be that the problem has been solved and we are ready to move on to new business. Being familiar with the *Book of Kells* through George Bain's book, I remembered seeing his reconstruction of the same T. As a term project for the

course in wearable computing and complex systems, there was an opportunity to examine both drawings in detail.

At this point we encounter the issues in Non-Photorealistic Rendering (NPR) studied by Tobias Isenberg, et. al., (2006). His study involved an observational assessment of NPR rendering for producing scientific illustrations after the style of hand drawn pen and ink often produced by professional illustrators for publications involving archeology, botany, and medicine. The study involved qualitative analysis of such questions as which images the participants liked best, least, the context in which any of the images should be used, whether any would be suitable for illustration in a college textbook, in a children's book, which were most likely computer-generated, which hand drawn, and any specific things that were noticed, and why. Issues determined to be important to the participants were how well shape was depicted, how well realism, and then aesthetic appeal. Opinions were divided between whether images would be used for scientific illustration or in an artistic context. Non-professionals had more difficulty describing what made something look like it was computer generated as opposed to hand drawn. Trained artists were observed to pay more attention to line quality. They observed that images likely to be computer generated appeared mechanical, that *line and mark making* was not apparent, or that patterns were artifacts of a rendering process. They described hand drawn images as

... more organic, tentative, broken up, having different line qualities or line weightings, and having little inaccuracies, as well, as inconsistencies.

Conclusions from this study included the following:

- Know your audience, in other words understand the needs of the particular context
- High quality output is important, in other words, reduce artifacts that may be the result of a particular rendering process and create a distraction
- Portray materials accurately according to the context
- Understand how a model may affect the visualization, in other words understand what to emphasize
- Avoid patterns and regularities if the work is to appear hand drawn
- Follow established rules of the traditional rendering technique, in other words how lines or dots are placed and the use of visual cues involving such things as line weight.

Gwenda Adcock (1974) cites the precedent of scholars in archeology of relying extensively on drawings for illustration and analysis when making this choice for her study. Drawings allow attention to be focused on the specific issues of investigation without distraction by the physical state of the subject, be it faded, worn, eroded, or otherwise disfigured, unless those are the questions involved. Drawings are often preferred to photography to overcome limitations in recordable detail that may result from lighting, aspect, or accessibility; and unit measure may also be more readily recorded without perspective distortion.

These issues become evident in the Kaplan and Cohen reconstruction of the *Kells* T. Rendering of lines are of uniform weight. Repeating patterns are regular. Specification of the reconstruction does not match details in George Bain's. This is even

more apparent on inspection of the near facsimile color print contained in the text by Francois Henry (1974). In places it appears that problems may also be structural. An automatic grid generation method adapted by Cameron Browne (1998) is cited in the references. Adding noise to control points in his grids give rise to organic effects on decoration. There are, however, some problems in how the method deals with changes in shape and width that are apparent in his Figure 13. These might explain comparable artifacts that appear near shape transitions in the Kaplan reconstruction. So, the question becomes, how might one go about making improvements in these areas, fully realizing that apart from the hand of the human artist, expressiveness in computer generated content will likely always suffer limitation.

1.3 Hypothesis

Any survey of technical methods in the area of Celtic design now suffers from the extensive volume of materials already published. Knotwork has been extensively studied. Its theory of alternating knots now contains existence proofs and has reached the stage of classification and enumeration (Thurston 1954, Menasco 1993, Soulie 1994, Sundberg 1998, and Jacobsen 2001). Peter Cromwell (2004) recently added the results of a topological analysis of the question of pattern restrictions. This study now forms the basis for explaining the numbers and types of patterns observed and possible for Celtic interlace in the archeological record. We may conclude that many of the properties of Celtic knotwork are now well developed.

Gwenda Adcock (2003) recently completed a study of the interlaced animal designs (zoomorphics) found in Bernician sculpture. Bernicia was a 6th century Anglian kingdom that preceded Northumbria running North from where the river Forth flows

into the North Sea and extending as far west as Whithorn in southern Scotland. Her study informs us that the design principles for animal interlace are very similar to those for knotwork. There is evidence of the use of templates for design and layout. Designs are aligned along the diagonals of grids. There is evidence that sculptors had access to design resources similar to those of artists working on manuscripts of the same period and she compares and contrasts their works with those produced in monastery at Lindisfarne.

Where Celtic knotwork strictly involves closed curves, animal designs more often involve open curves. Such curves are described as tangles in topology and significant progress has been made in their description, analysis, and enumeration with studies directed by the needs of mathematical physics (Jacobsen and Zinn-Justin 2001, 2002, Kauffman and Lambropoulou 2002, Zinn-Justin 2003, 2004, Bae and Morton, 2007).

Reviewing the recommendations from Isenberg's we find more limited development in the area of mark making. The studies of Decarlo (2003) and Sousa (2003) on line weight and cuing are cited. Searching further we find that the analysis of handwriting has lead to a standard for automated scoring of recognition algorithms (Wenyan, 2001). We also find some work on the effect of line shape in conveying emotional aspects of expression (Freeman 2003, and Shurina 2006). NPR research has also advanced to the point where physical modeling of the behavior of pencil and ink on paper (Small 1991, Zhang 1999, Sousa 2000, Chu 2005) and brush and paint on canvas (Baxter 2004) has been accomplished. Although Raman and hyper-spectral analysis has been applied to analyzing the inks and pigments used in medieval manuscripts (Burgio, et. al, 1997, Melessanaki, et. al, 2001, Gardner, et. al., 2003, Brown and Clark 2004a, 2004b, Edwards and Perez 2004), and a similar study is planned for the collections at

Trinity College (Quinn, 2007), mathematical modeling is not found for the physical properties of their behavior on animal skin vellum.

Where we find difficulty in this foundation begins with Wenyin, who notes instability in the mathematical approach that synthesizes sample patterns. We also find ourselves with the problem of the actual drawing of lines. Where the now well understood method of parametric curves is applied in each of the examples mentioned, these tend to be less expressive than the human hand. Part of this is related to the control vertices which must somehow be placed. Part of this is due to their approximating the intended shaping. Some preliminary work in the evolution of curves and self-organizing geometry is found in the Ph.D. thesis of Kurt Fleischer (1995). Modeling diffusion reaction equations and a limited set of behaviors observed to be present in the walls of biological cells, he was able to demonstrate the evolution of structural groupings and chains. Searching further we find an experiment where a trefoil was observed as a structure organized by constraints in a lattice gas model of a bistable chemical system (Malevanets and Kapral, 1996). While the natural formation of these constraints remains an open question, it is interesting that such a complex structure was observed to form. More recently we find work combining the notions of shape grammar applied to the derivation of cellular automata that model lattice gases (Speller, 2006).

With these ideas in mind, the question presented itself of how complex systems methods and differential geometry might provide advantage in Celtic design that involves animal forms. In particular our attention was drawn to the challenge posed by George Bain's reconstruction of the chaos in the lozenge that rests at the crossing of strokes in the Chi in folio 34r of the *Book of Kells*, the Chi Rho or XPI monogram page:

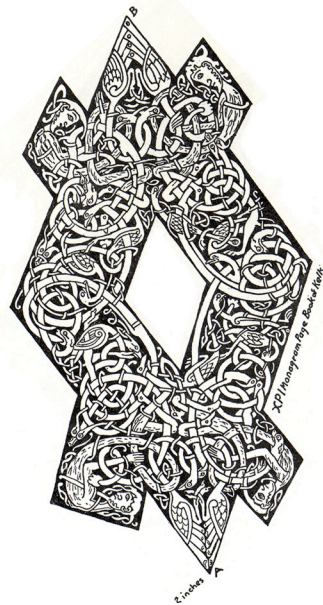


Figure 1: Lozenge of MS 58 34r, George Bain (1951) left, Board of Trinity College Dublin (2007), full size, right

By his count there are four humans, four reptiles, four beasts, and 12 birds contained within this tiny space. On casual observation we observe almost none of the regular structure found in much of traditional Celtic interlace. We are confronted by what appears to be a mass of tangles. What we might hope to discover are structures from complex systems theory that help us organize this kind of dynamic and from differential geometry, techniques that are similarly capable of generating the complexity of line. Thus we find ourselves with the following set of questions:

- How do we compose?
- How do we accomplish spatial organization?
- How do we decide what fills the spaces/frames?
 - How do we choose unit forms?

- How do we decide which interlace to use?
- How do we control crossings?
- How do we draw lines?
- How do we draw cords and anatomy?
- How do we provide the means for a designer to be expressive?
- How do we facilitate the creative process?

As with the archeologists on whose research much of this work will rely, the results will be presented in the form of sketches which hopefully will reflect qualities of the hand drawn line.

1.4 Contributions of this thesis

What we believe to be the contributions of this thesis are as follows:

- The use of lattice gas automata to explore the design of curves
- Genetic specification of planar space curves as the foundation for line drawing and structural anatomy
- The use of linear cellular arrays to handle the complexity of integrating the solution of curves and parameters of design-- these include the association of curves with structure and crossing states, the coding of boundaries to enable the processing of occlusion, the coding of distances to facilitate the incorporation of depth cues and line weighting, and the placement of design elements in a composition

- The use of original line art as specifications for design elements
- The use of linear genetic programming to handle the complexity of specifying, assembling, and manipulating a composition under this level of complexity

1.5 Organization of the thesis

The remaining chapters of this thesis are organized as follows: chapter two presents a general review of the related research. This includes summaries of illuminated manuscript production in the context of 9th century and how it influences the scope of this research, known design constraints from the archeological record, the modern design methods and their expression as algorithms, creative design theory, drawing, mathematical foundations (complex systems theory and methods, differential geometry, and noise), physical media, and the issues in art criticism that influenced what was developed.

Chapter three summarizes the development associated with answering the questions in section 1.3 and describes the experiments conducted to determine which ideas work. This includes the development of a curvature dynamics model that facilitates the genetic description of space curves, the use of Perlin noise as a path design method and accompanying difficulties, a lattice gas model of reflection, morphological analysis of the canvas, the design of hierarchical grids for design structure, experiments with automatic methods from finite element analysis and self-organizing maps and their attending difficulties, morphological analysis of unit forms (cords and animals) with advantages and disadvantages, the derivation of the characteristics of hand drawn lines,

analysis of crossings and resolution of self intersections, and finally rendering and occlusion mapping.

Chapter four provides an overview of the system and its interfaces. This describes the general organization of the system, the workflow, and artist interaction.

Chapter five presents details of the design in terms of its biological analogies. This includes a description of phenotypes and genotypes, as well, as the linear genetic programming model.

Chapter six presents results and conclusions of the research.

Chapter seven provides suggestions for further research.

Given the diversity of the audience for this type of research, the general philosophy guiding the development of discussions is that the material should be self-contained and comfortably readable without attending the technical details. There will be places where understanding of the technical detail will require familiarity with the calculus.

Chapter 2 Related Work

Background and General Review of Related Work

You develop very compelling techniques—visual techniques—that are interesting. But the final outcome seems to be based on all the wrong models of visual art. It's not boring for the practitioner, but it's extremely boring for the viewer.

Bob Riley
San Francisco Museum of Modern Art
(Spencer, 1989)

Generation, however masterful, without evaluation just does not count as creativity.

Matthew Elton
(Elton, 1995)

... these systems do not model creativity because they cannot successfully recognize the creativity of their own work.

Rob Saunders
(Saunders, 2002)

If it was procedural, Sam didn't like it.

Peter Nofz
Digital Effects Supervisor, *Spider-Man 3*
(Scoredos, 2007)

2.1 Introduction

What we hope to discover are methods that give rise to the complexity of design that we observe in insular manuscripts in the period c 500-900. This chapter will review and summarize the substantial body of interdisciplinary research that we have to draw from.

Materials relevant to the questions may be divided into twelve subject areas:

- (i) Illuminated manuscripts
- (ii) Archeological design constraints
- (iii) The modern methods
- (iv) Computer generated construction
- (v) Composition and creative design theory
- (vi) Mathematical foundations
- (vii) Complex systems theory
- (viii) Differential geometry
- (ix) Noise (bridges and procedural)
- (x) Characteristics of the hand drawn line
- (xi) Instruments and media (physical characterization)
- (xii) Art criticism

Additional reviews involving details of specific problems will be addressed as they are encountered in later chapters.

2.2 Production of an insular manuscript

Accounts as to how the great insular works came into being, vary. It may be that these differences arise from contrasts between the enterprises later formed for commercial gain and those of these earlier times formed as celebrations of faith; although the two may be found to be linked in both Hebrew and Christian traditions (Barker, Proverbs 16:3). In the present case, the findings have actually affected the way in which the research was done.

Christopher de Hamel (1994, 2001) offers a glimpse into the basic structure of the undertaking. Commissioned manuscripts were entered by contract. As with contemporary practice these contracts defined what was to be produced, although the costs of some items may have been negotiated, as we find in a rather pointed letter regarding a vellum order several centuries later by Planudes, head of the school at Khora (Abt, 1989). Production would involve resources ranging from a single copyist to a pipeline engaging a large number of people. The content and layout of a manuscript would be meticulously planned for text and illumination. The relationship between this process and the obtaining of vellum is not entirely clear. The final stages of layout and planning are observed to have taken place on the vellum itself. Since vellum is expensive even now, one might infer that some effort would go into determining the amount that would be needed.

Next would come the making of pages from calfskin. Since a manuscript involved a large numbers of pages, it is believed that this process would begin in butchery with the carcasses of animals taken for meal preparation. de Hamel (1994, p. 86) notes that monasteries that did not consume meat would also be without vellum. Skins would be put through a process of washings and soakings. Soakings would be in a mixture of water and lime after which hair could be easily removed. Processed skins would then be stretched on frames for drying and further preparation by rinsing and scraping. A finished skin, now vellum, would be folded (*folia*) according to the dimensions of the manuscript of its intent and trimmed to the size of the codex (page).

Scribes would next take the *folia* and nest them into quires (signatures). Registration of the pages in a quire would be maintained by boring holes along the common fold with an awl. Final layout of content would begin by ruling the pages. Places for art and ornament would be framed and annotated. Pages would be numbered and the quires

dismantled so that the individual folia of the book could be distributed to scribes and artists for rendering.

Production of artwork involved a multiple step process that would begin with under drawings rendered or traced into allocated spaces. These would be pressed by hard point or sketched with a form of graphite—early pencil sketches. As revealed by partially completed manuscripts, under drawings were often as compelling as line drawings as the finished paintings (Alexander, 1992, p. 38). Artists would paint these designs in layers beginning with the pressing of gold leaf (if used). The painting was commonly done with egg tempera, where the yolk of an egg is separated, pierced, and mixed with a small amount of pigment. Amy Baker (2005) describes the pigments identified in medieval manuscripts as coming from many different sources including animals, plants, and minerals. She divides them into four categories: earths, minerals, synthetic, and organic. Examples of earths would be iron, clay, and manganese dioxide (a black brown) for reproducing earth tones. Minerals would include ultramarine, azurite (dark blue), and orpiment (yellow). Some colors were synthetic obtained by processing of other materials. These would include lead white, red lead, and vermilion (red). An example of a rare organic pigment is the kermes (crimson) prepared from the dried female insect.

Problems with these materials included toxicity (Baker, 2005) and if the vellum was not properly prepared, a tendency to flake with age (Abt, 1989). Artists would apply these colors background to foreground. Application would be done with brushes as fine as a single hair.

2.3 Iconography

In the insular period of the first millennium manuscripts were prepared primarily for the collections of monastic libraries. In reviewing the photography and illustrations provided in the analysis by Françoise Henry (Sheffer, 2007), one is drawn to common forms and repeated figures across the insular manuscripts. The portraits and symbols of the apostles, the portrait of Mary, the images of Christ and the Chi Rho page, and the Liber page and more are rendered in a manner expressing a common origin of design. It is this that reflects the deeper structure and subsequent culture of iconography and the iconographer. The style of abstract representation of human and animal forms emerges near the time of the iconoclastic crisis where the role, purpose, and representation of symbols and likenesses came under challenge. The central question involved the differentiation of these images from idols. The issue would subsequently be resolved in favor of icons and images through precise reconciliation of their purpose, form, and meaning (Barrow, 1975, p. 11).

An icon is an instrument of prayer (Forest, 1997, p. 19) that symbolizes a specific Saint in the history of the Christian faith. The icon itself is most strongly associated with orthodox traditions (Greek and Russian) where it represents a window into heaven. The form of an icon is fixed. Its basic structure is always the same. Still within this constraint individual iconographers have found room for the expression of inspiration accompanying individual renderings.

The painting of an icon is a labor of faith involving a skilled artist, the iconographer, a tradition of methods, materials, and an accompanying journey of prayer, fasting, and meditation. Monks working on manuscripts are frequently depicted as working alone in private quarters (de Hamel, 1994, p. 64). The final passage of an icon is from its maker to its recipient through a process of association and Church blessing.

That these symbols are present in the insular manuscripts may be interpreted as telling us that the entire undertaking was one of faith and dedication in long solitary hours as work proceeded.

What this mean to us is that when work on a manuscript was completed its quires were reassembled, a binding was sewn, and in case of a gospel, a cover likely fixed to adorn and protect it. From there a gospel may have followed the path of icons to a monastery altar (citation).

One clear purpose of the decorated pages is to mark the sections of books. Illustrations may have also provided visual references intended to assist with its public reading (Farr, 1997, p.45). The numbering of verses as an organizational method would not be devised until the early 13th century (Barrow, 1893, p. 73).

2.4 Celtic Symbolism

Insular manuscripts are characterized by the incorporation of Celtic forms whose meaning, beyond lavish decoration, likely remains lost. There is some analysis that associates placements with the meaning or implications of stories on the page, e.g., if ignoring some lesson leads to chaos, a vignette depicting chaos may be present (citation). Others placements appear entirely whimsical. Christianity applied new meanings to the historical symbols of the Celts. Some common interpretations are as follows (Meehan, 1994):

- The snake, with the shedding of its skin representing rebirth or renewal, may be associated with Christ's resurrection.

- An ancient myth surrounding the still birth and resurrection of lion cubs may also translate the lion into a symbol of Christ's resurrection.
- Christianity is historically associated with the symbol of the fish.
- The peacock may be associated with Christ's incorruptibility and immortality.
- There appear otter, moths, cats, and mice, whose symbolism is less clear.
- Then there are fanciful characters, including birds with multiple sets of wings, mermen, and griffin. The griffin may symbolize evangelical elements such as the Eucharist.
- Continuous lines symbolized eternity to the Celts. Knots will often contain the longest possible cords.
- The complexity of knotwork in Celtic tradition is associated with perfection.

2.5 Archeological design constraints

This section provides a summary of design concepts, dimensions, and attributes of pose that guided the design of the insular manuscripts. These dimensions establish proportions and limits for design elements. In describing and illustrating these features, we will be following the precedence established by scholars in archeology of relying on line drawings. For purposes of illustration, this overcomes any limitations associated with the condition of subjects under study (Adcock, 1974, xiii-xiv).

The insular manuscripts form part of the early history of the departure of writing from scrolls to the codex (page). The page defines a frame having specific dimensions and aspect. Within this frame space is allocated to borders, text, and illuminations. These spaces, their dimensions, and the designs within them, have been subjected to intense

examination by archaeologists and art historians. Guilmain (1985, 1987) focused on the structure, design, and method of the spirals, interlace, and animal designs contained within the panels of the cross carpet pages of the Lindisfarne Gospels. Where grids provide structure for interlace in square panels, there are pages such as folio 26v where the design is based on patterns of circles. Guilmain provides a reconstruction of a detailed development of geometry within this page first carried out by Bruce-Mitford for the preparation of the facsimile volume, *Evangeliorum quattuor codex Lindisfarnensis*, for the Lindisfarne Gospels. Similarly detailed developments have been carried out to describe the distribution of spirals in folio 33r of the Book of Kells (Hull, 2003, p. 145-147). Reconstructions by Bruce-Mitford go still further in finding similar complexity within individual ornaments (Alexander, 1978, p. 11).

The studies of Robert Stevick (1994) demonstrate that designs on these pages result from methods based on classical measures and proportions. He presents constructions four panels from folios 2v, 94v, and 138v of the Lindisfarne Gospels based on golden ratio subdivisions. He presents constructions of layouts for the apostle pages of St. Mark and St. Paul in the St. Gall Gospels and the cross carpet page 220 from Lichfield Gospels based on 4 x 3 rectangles. He finds a geometric ratio in a similar construction of the apostle pages for St. Matthew and St. Luke.

The structures associated with the layouts of these pages, by themselves, resemble architecture. Megan Hitchens (1996) observes this parallel in scripture references involving houses, builders, foundations, and stones, the temple, dwelling places, and analogies to the person of Christ. She then finds the Pythagoreans writing ... *number and geometry equate with each other at the level of spiritual meaning*. Stevick views each page as a study in graphic computation. His studies frequently find at the heart of these constructions the squaring of a circle:

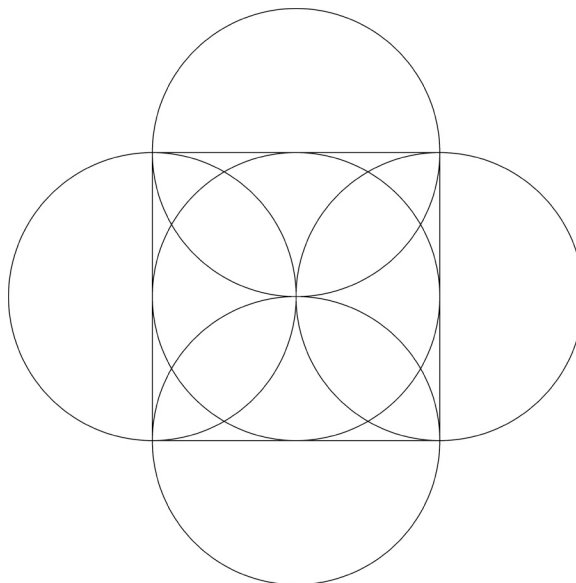


Figure 2: Squaring a circle

Hitchens cites Stevick's assessment of this form as *perfect* in that it *evolves from the circle and the cross, signs of perfection themselves*. Within the numbers and the dimensions, the derivation, and the positioning of every structural element, spiritual meaning appears to be inferred or assigned. The work of the illuminator was translated into fulfillment of spiritual purpose. The squaring of the circle facilitates the systematic subdivision of the page. An example of the construction of the frame, circle, and cross of the Soiscel Molaise book shrine (a manuscript cover) following an analysis of Robert Stevick that begins with the squaring of a circle is shown below:

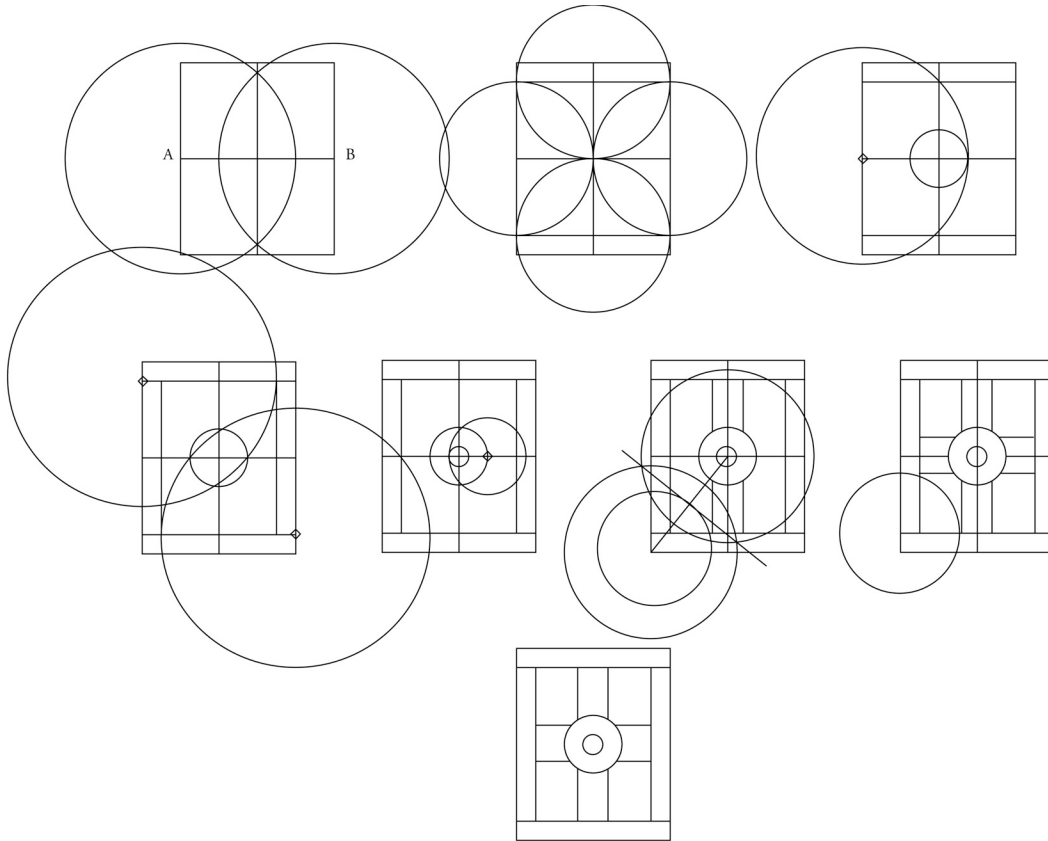


Figure 3: The frame, circle, and cross of the Soiscel Molaise (Stevick, 1994, pp. 6-7)

The effect of these methods under the guidance of insular imagination was a layering of geometric structure with semantic symbolism from the grand program of a piece to the most intricate frame. Within the panels that emerge from this process we find the various elements characteristic of insular design: interlace, human and animals forms, spirals, plants, and key patterns. The attributes of interlace, human, and animal forms are essential elements of the present study. We next review the relative proportions of these elements and their supporting structures.

Interlace has been the subject of a number of now famous archeological surveys and studies (Allen 1903, Bain 1951, Adcock 1974, Bain 1986/1992, Budny 2001). Design is typically set out on a square grid with a single subdivision. Characteristics include the

numbers of strands that can occupy a square, the width of the strands, the proportions of the symmetrical loop and the u-bend or chevron.

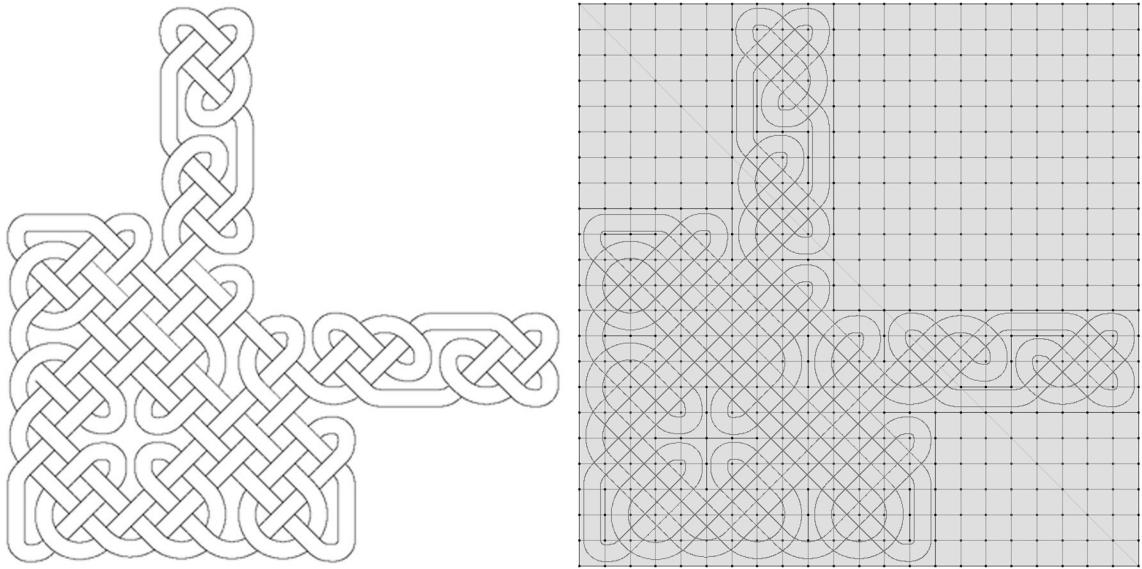


Figure 4: Lindisfarne f26 without chevrons (knotwork, left, subdivided 11 x 11 grid, right)

Measures have been taken for both cords and their medial lines. The medial line of a symmetrical loop is observed to have the following dimensions (Adcock, 1974, Figure 6):

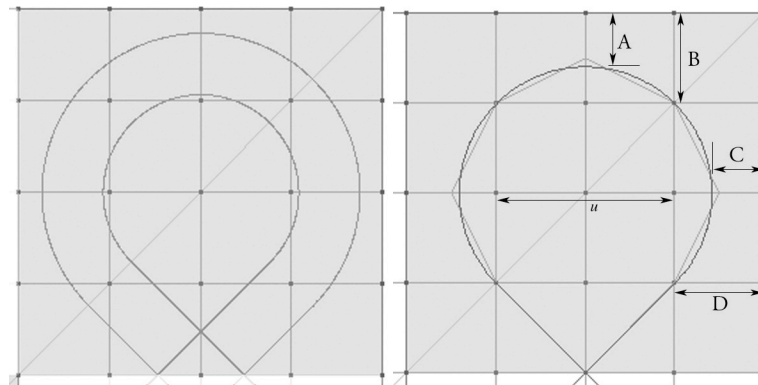


Figure 5: Dimensions of the symmetrical loop

where $A = \frac{1}{4}u$, $B = \frac{1}{2}u$, $C = \frac{1}{4}u$, and $D = \frac{1}{2}u$. In the example shown here with circle and line, the radius was set to $u\sqrt{2}$. The dimension given for C requires an adjustment either in the start and end of a circular arc or the use of a blending curve. The width of the cord is scaled as $\frac{1}{3}u$, which is common in manuscripts.

The medial line of the u bend loop typically has the following dimensions:

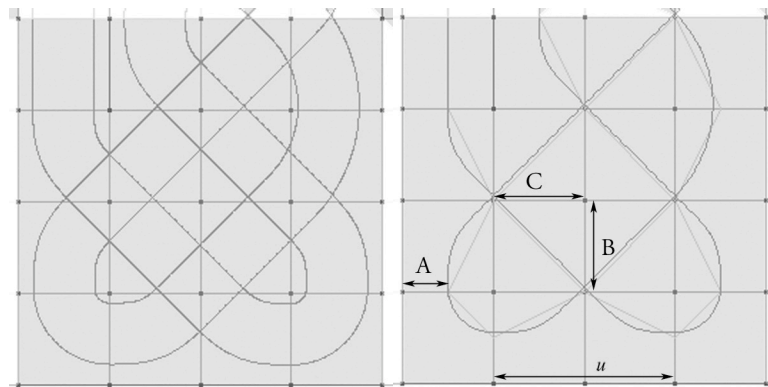


Figure 6: Dimensions of the u-bend

where $A = \frac{1}{4}u$, $B = \frac{1}{2}u$, and $C = \frac{1}{2}u$. This type of reversal is commonly interpreted with a chevron and extends to closely fit the corner of the square.

Interlace patterns have been observed to be set out with one strand passing per unit of the grid, as well, as others with two. The widths and spacings of cords have been observed to vary with a maximum width being $\frac{1}{2}u$. Cords having a width of $\frac{1}{3}u$ are common in manuscripts and set out with two strands per grid unit.

Iain Bain (1992) notes that the grids associated with insular design are not always square and are shaped according to the needs of the designer. As with J. Romilly Allen, he notes the use of rectangles, rhomboids, trapezoids, and annular segments. The rectangular

grid with an aspect of 3 x 4 is common. In each of these cases the dimensions of the passing strand are skewed or distorted proportionally. While Iain Bain devotes time to the discussion of the geometry of curves, the secrets of the ideal remain shrouded. There are many examples in which the basic drawing of a curve by means of a compass leaves room for sweetening. In these cases we observe that abrupt changes in curvature from compass to line can be softened by a blending curve that provides an easement into the straightening of the line. In other cases the distance to the next turn may be such that the blend provides easements between the turns removing the line altogether. Deciding when to fit such blends becomes an art form.

When searching for similar characteristics in animal designs, the diagram of folio 26v of the Lindisfarne gospels by Bruce-Mitford (Guilmain, 1987, p. 43) suggests order without necessarily a grid. From a level of detail perspective this would appear to be a reasonable assessment. However, layout studies by Adcock (2002, p. 5) plausibly argue that grids also apply as the framework for animal design. Because of their size and complexity, this becomes an exercise in forensic visualization, where one must infer a skeletal framework around which a creature's anatomy is organized. It is this framework that forms the basis for interpreting the design of interlace. Once done, what is observed as normal in animal design is for bodies and limbs to move diagonally and curve in the same manner as interlace. While paws or noses may be tucked for closure or to support the interlacing of limbs with other animals, what is noted as a lack of continuity due to unterminated extremities is the evolution of tangles from the open curves of the medial lines or skeleton.

Extremities are used creatively. Ear lappets, tails, wings, legs, and torsos are adjusted to a space or draped over forms any letters. Every part of the anatomy can be extended and turned into knotwork interlaced with other creatures or to fill spaces between them. Wings can evolve into spirals. Heads can be posed in any direction. Patterns are

formed by repetition and the adaptation of frieze motifs and we often find these organized around vertical and horizontal symmetries in a manner similar to knotwork. The use of these features varies from manuscript to manuscript, bearing the signature styles of individual artists. As with stonework designs it is possible that some designs were first worked out and then transferred. The similarity of shapes also suggests the existence of collections that served as starting points from which creature designs could be adapted. Pattern books are certainly known from the fifteenth century (de Hamel, 2001, 62).

Some work has been devoted to describing and organizing poses (Marx, 1995) using abstractions formed of line drawings; however, without more tangible connections to the creature being designed, such as those found in the intermediate drawings by George Bain, the abstractions are not easily interpreted. There are also a number of poses illustrated at the end of the paper that are impossible to describe; and their denotation would resemble scribbles. Adcock suggests that the number of animals found in Celtic design also limits what can be done to organize them.

Returning to the lozenge in the XPI monogram page we notice that the constructions by George Bain do not include a grid and that the poses of the creatures appears freeform. This becomes a point of contrast between designs in the Lindisfarne gospels and those in the Book of Kells. Animal designs in the Lindisfarne gospels appear orderly and almost sedate. Their form has been so carefully planned that their appearance is at times almost mechanical, reminiscent of Lichtenstein in recent times, with the grids and compass marks clearly visible on the reverse of some pages (Browne, 2004, p. 11). Animal designs in the Book of Kells are wild and at times seeming uncontrolled almost suggesting that their designers were having difficulty in keeping them from taking on a life for their own.

2.6 The modern methods of construction

At the time of this most recent survey there are about five modern theories covering the construction of Celtic interlace:

- A theory of plaits (Allen, 1903)
- A theory of points (Bain, 1951)
- A theory of grammars of elementary forms (Adcock, 1974, Sloss 1995)
- A theory of reflections--mirror curves (Gerdes 1989, Jablan 1995)
- A theory of topological assembly (Mercat, 1998)

This section will provide a brief summary of each of these theories.

2.6.1 The theory of plaits

In the analysis that accompanies his survey, J. Romilly Allen (1903, pp. 142-145) observed that interlace appears to be similar to plaiting. He further observes that this is likely a later adaptation by Celtic designers, as it is not found in earlier indigenous metal work, which serves as the basis for many of the figures we find on the pages of insular manuscripts. A plait consists lines set out at 45 degrees as in a lattice. If we bend the edges back towards the center; and then carefully order the over and under relationship at each crossing so that it alternates, we find the most basic interlace pattern:

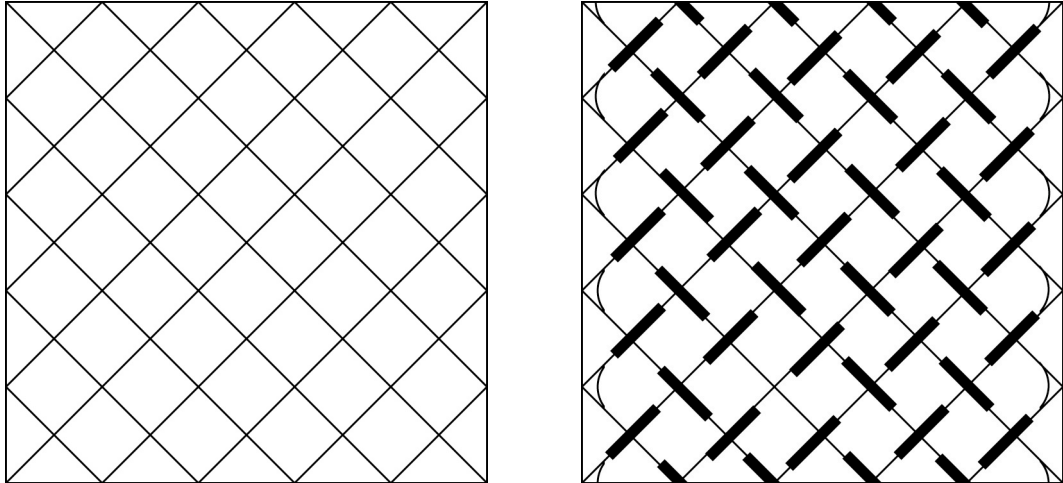


Figure 7: Simple plait (left), interlaced plait (right) (Allen, 1903, p. 144)

Allen describes that each of the intersections may be further treated by making one of the following substitutions as long as the resulting interlace remains alternating:

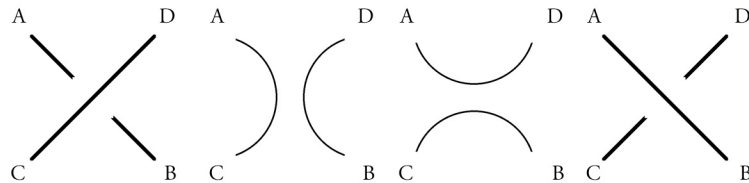


Figure 8: Left hand crossing (C to D), vertical break, horizontal break, right hand crossing (B to A) (Allen, 1903, p. 145)

These substitutions are recognized as the tangle operators from modern knot theory (Conway, 1970). Closure is accomplished by turning back appropriately at the bottom and top of the pattern. The shape of the lattice is apparent. The grid appears when recognizing that the points of these diamonds bisect the edge of an enclosing square. We then recognize that the breaks may be places on any of combination of these edges to steer the cord to the needs of the design. The number of variations accommodates the imagination.

Adcock summarizes nine movements that are observed to accompany the use of these breaks. Constructions of them are included here, as they formed the basis of a pattern search experiment described in Chapter 3:

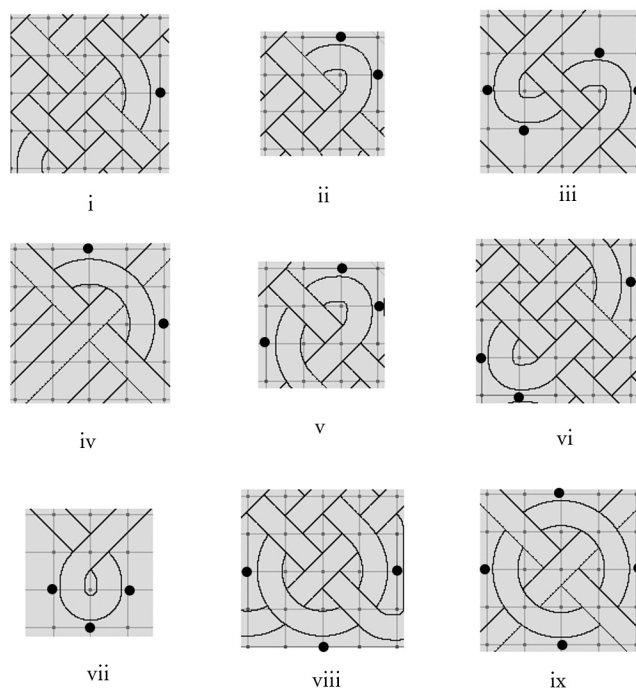
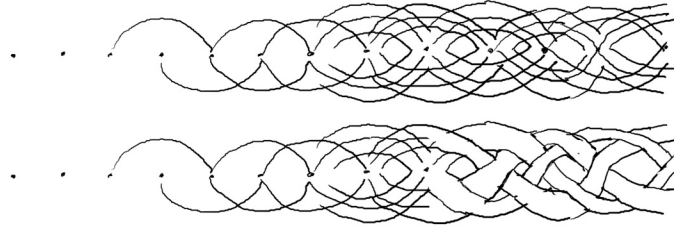


Figure 9: Elements of J. Romilly Allen with breaks (Adcock, 1974)

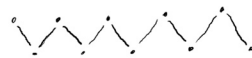
2.6.2 A theory of points

Iain Bain's research was stimulated by the need to explain the method developed by his father, George (Bain, 1951). The difficulty arises in that the imagination must be fully engaged. We begin with a row of points. This row of points describes all patterns that can be constructed from a single row of right or left hand crossings, where each point represents an intersection:

One Point:



Two Points:



Three Points:



Figure 10: Theory of points

Diamonds re-emerge as rows of points are added and again we can generate a grid by bisecting edges of the enclosing squares with these points. Once the idea makes sense, and assuming one has a much steadier hand than mine, one can do pretty much any kind of Celtic interlace with this basic schema. Adcock (1974) provides the clearest explanation of this method.

2.6.3 A theory of grammars of elementary forms

Now that Large catalogues of interlace patterns have been carefully assembled and analyzed, Iain Bain (1986) credits Gwenda Adcock (1974) with the idea that patterns of this kind can be generated grammatically as assemblies of elementary forms or templates. The concept engages the notion of tilings of the plane (Grunbaum, 1986). What we seek is a set of tiles that can be assembled, lock and key, after the manner described as “jigsaw pieces” by

Andrew Glassner (1999b). When the tiling is finished, the patterns on the pieces form interlace. The method was most dramatically demonstrated in a tile set designed by Andy Sloss (1995).

In his design Sloss first relocates the centers of a two strand per square grid to the edges of the traditional arrangement, thus placing most intersections at the center of his squares. Each square is then painted with one of the possible cord traversals. Each of the four corners of is then coded by the way the cords enter and leave, vertically, horizontally, or diagonally. This gives rise to constraints as shown in the following figure:

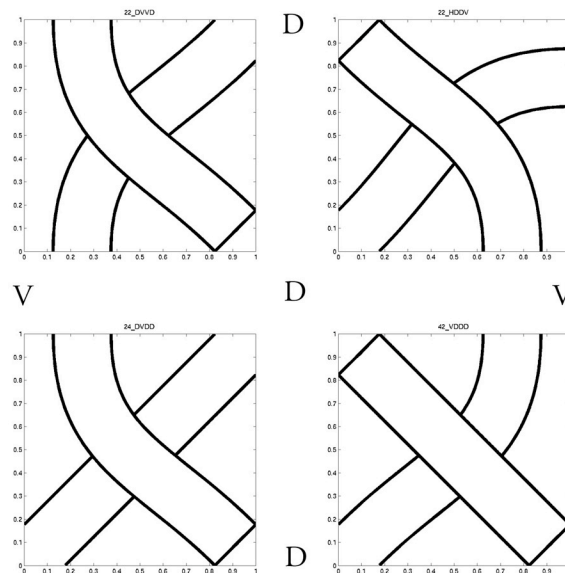


Figure 11: Tiles joined on corner crossing constraints (Sloss, 2002)

If we assemble pieces from the tile set on a grid, making sure that diagonal labels are always matched edge to edge, the result is a panel of interlace. An example from one of the student's early experiments on constraint propagation is shown in the following figure:

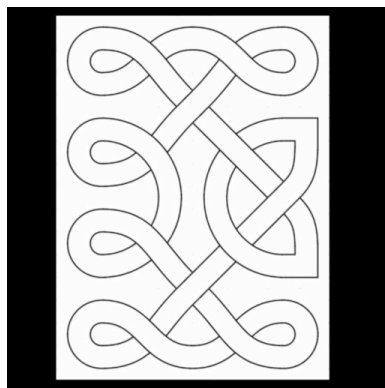


Figure 12: Andy Sloss' approach, 3x4 panel (Doyle, 2005)

The reader may recognize this method as a form of labeled shape grammar, the theory for which was developed by George Stiny (1975). The primary drawback of this particular implementation is that the tile set is closed and after a time the patterns will be found to become monotonous, predictable, and repetitious.

2.6.4 A theory of reflections

In exploring Tamil threshold designs in South India, Paulus Gerdes (1989, 1997a, 1997b) observed that these patterns could be generated by tracing the path that a ray of light would take reflecting at an angle of 45 degrees assuming that the boundaries of the designs were mirrors and other mirrors interior to the boundary were carefully placed. At some point these became known “as mirror curves.” Extending his studies he noticed that Celtic interlace could be generated using the same principle. Jablan (1995) has begun the development of a formal theory for the method.

To see this a bit more clearly, we carefully crop the plait we used examined earlier to its boundary and observe the trajectories of the diagonals:

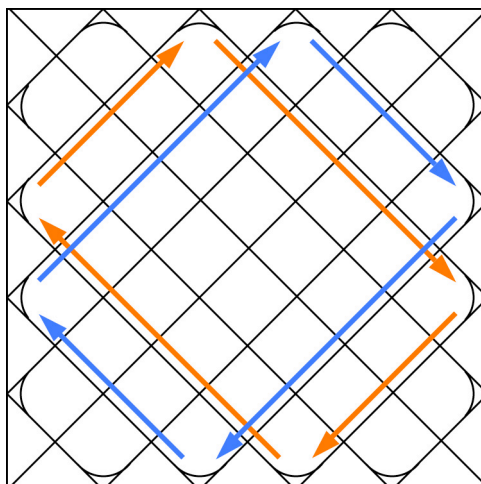


Figure 13: Mirror curves

One will notice how the diagonals “bounce” off the boundaries of the plait. The breaks of J. Romilly Allen’s plait theory are interpreted as mirrors by this method. After the fashion of turtle geometry, cords may be inserted along the path taken by the ray and through its reflections. Again the grid emerges, this time in the places where one hangs the mirrors. This would appear to be a very simple and elegant idea.

2.6.5 A theory of topological assembly

This last theory forms a general method for assembling knotwork on a grid. When looking at a trefoil, Christian Mercat (1997) happened to see a central triangle and its dual, the Voronoi diagram of the bounding polygon. He noticed the way in which the cords crossed the edges of the Voronoi diagram. Each cord was associated with a specific direction. If one treated the cord as a flow and placed a bidirectional arrow in its center, he noticed the opposing directions of the arrows as each flow crossed the boundary. The cord is split into six pieces by the Voronoi graph. What one needs to do is to reassemble the cords at each edge. So, there are a total of four connections that need to be made on each

edge. The bidirectional arrows provide four places for these attachments. What he realized was that the Voronoi graph together with the arrow labeling system formed what one would technically call the projection of a tetravalent graph into the plane. In other words, a labeling system in the plane capable of forming four bonds the result of which is the connecting of the four pieces that make up the two lines that cross an edge:

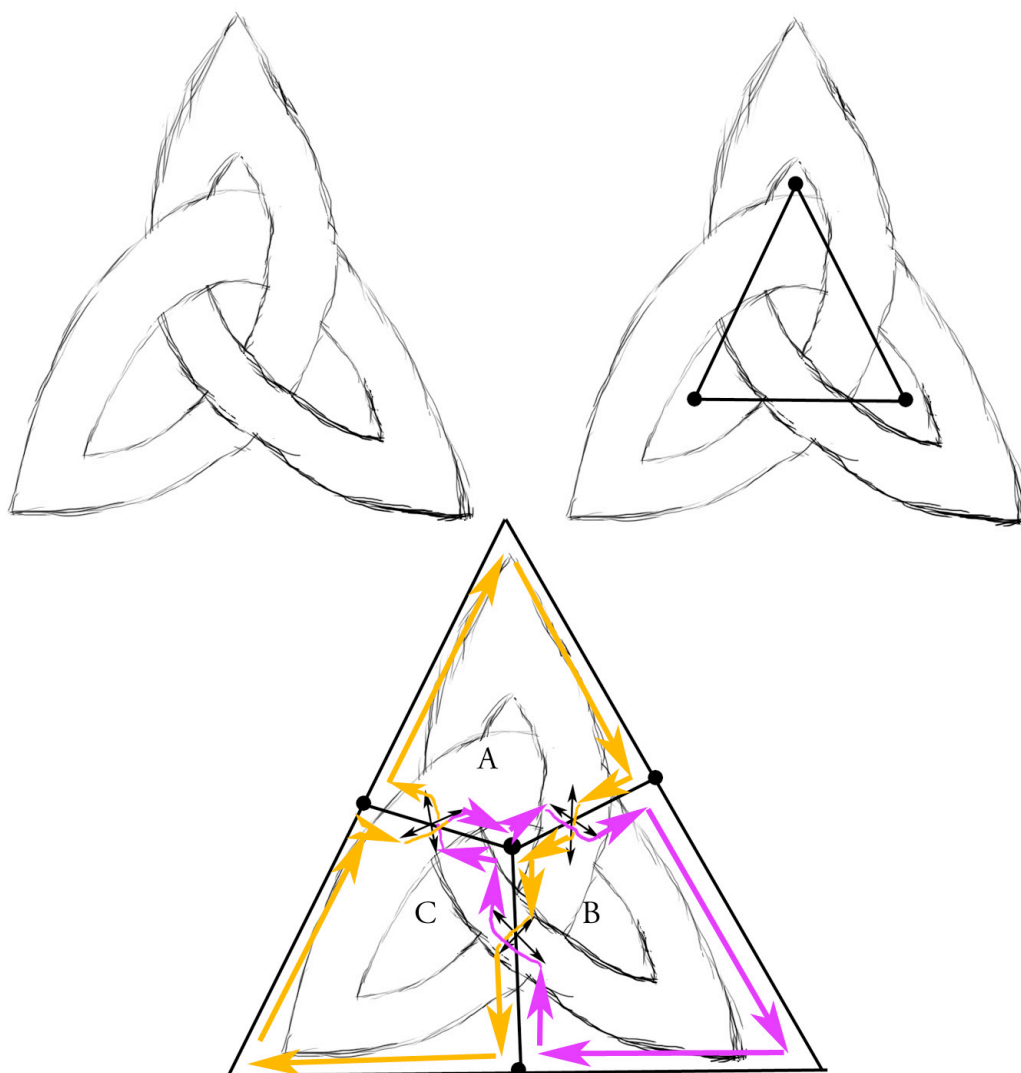


Figure 14: Mercat's trefoil and its tetravalent graph

So, begin with one cord fragment at one edge of the Voronoi graph, suppose the first of the orange arrows in A, at the bottom left of the polygon. Follow the edge of the graph

until you reach the next connector and the arrow opposite your direction of travel. Dock the cord there. Follow the connector across the edge. Pick up the cord fragment docked there. Continuing in the same direction, follow edge of the graph until you reach the next connector and its arrow opposing your direction of travel. Dock the cord there. Follow the across the edge ... and repeat until you arrive back at the beginning. I changed the color of the arrow part way through to try and keep things clear.

If we had something more complicated, we might have breaks. A break would be indicated by replacing the connector and its four arrows by a break mark. Break marks behave like a mirror. Rather than continuing across the respective edge following the arrow we expected and continuing in the direction of travel, one would cross the edge and reverse direction until arriving at the next connector.

Suppose we now label the arrows so that they alternate between the states *above* and *below* according to the path we expected the cord to take. By reassembling the cord fragments as we have we have now produced the crossings needed to generate a Celtic trefoil knot.

The method works with any planar graph. A planar graph is one having edges that meet only at vertices. In other words it is not self-intersecting. The only difficulty with it may be that one must become adept at visualizing graphs by their intersections. One must also be meticulous in placing breaks and crossings on the edges of the graph so that there is always an entry and an exit for each of its compartments. There need be only a single crossing denoted for a compartment to be used.

In the survey of complex systems methods we will find something called a graph cellular automaton capable of supporting this type of procedures by using a particle to mimic what we just did in following the edges of the graph. As this was being written an interpretation based on gas lattice automata was also uncovered.

If we look at the preceding figure carefully, it is very curious to notice that the way the bounding polygon has been drawn, we have a triangle with a quadrilateral partitioning. There are a number of people who have worked very hard, Matthias Muller-Hannemann (1997) for example, to make these very well behaved for applications involving finite element simulation.

So, modifying the plait, point sets, tilings, reflection, and topological assembly by means of a tetravalent graph, are the five modern methods of constructing Celtic interlace. I have not included the evolution of filaments in bistable media for the reason that we do not yet have a general procedure for designing interlace by this method. Each time we noticed that we arrive back at a grid. Everything we have examined so far supports the thesis of Gwenda Adcock that the grid is the basic structure for designing Celtic interlace. Next we review at what has been done in computer generated Celtic design.

2.7 Computer Generated Construction

The general availability of personal computers with excellent graphics modeled on OpenGL has lead to a growing interest in software for exploring Celtic interlace. By means of these or vector-based drawing tools an index assembled by Peter Suber (2005) shows that a prodigious volume of resources and work centered on the study and practice of knots has sprung up in print and across the world wide web. In this section we will review what has been done.

An interesting hardware implementation is by Louis Kaufman (1994) who presents the design for an integrated circuit that implements tangle operators. A collection of these circuits is able to represent knots for experimentation and problem solving.

Doug Zongker (Zongker, 2000) implemented computer-generated Celtic knotwork based on Peter Cromwell's study of plaits (Cromwell, 1993). He presents a standard quadrilateral grid that allows the placement of breaks. A tile is inserted into each cell that satisfies the boundary conditions for that cell. This method Cromwell was able to illustrate Frieze groups and their modifications for Celtic interlace.

The shape grammar developed by Andy Sloss was used to develop a font.

Malevanets, Anatoly, and Raymond Kapral (1996) have presented an example of a trefoil that evolves as a filament from constraints on the solution of a bistable system simulated by lattice gas automata. This would be the first example of a complex systems process used in Celtic design. The derivation of spatial constraints for general design work was left as an open problem.

Steve Abbott implemented Christian Mercat's method in a program named Knots3D. A Celtic circle designed using Knots3D, cleaned, textured, and rendered by the author, is shown here:



Figure 15: Steve Abbott (1998) with Christian Mercat's approach

Robert Scharein's dissertation on topological drawing is most commonly referenced when describing how Celtic knots differ from those in pure knot theory (Scharein, 1998).

Celtic design belongs to the specialized area of alternating knots and tangles. Theory in this area continues to be developed for mathematical physics as described in the introduction.

A most notable contribution by Michael Wong, et. al., (1998) has stimulated interest in the broader question of computer generated ornament. His is the first work that autonomously develops designs based on fundamental principles of ornament. These are encoded as procedures invoked as rules by a production system (symbolic artificial intelligence). This method was chosen for its ability to sequentially control evolution during design, similar to constraints often imposed on shape grammars. It was believed that this was more manageable than parallel elaboration by an L-system (Prusinkiewicz, 1990, 1994). L-systems have become a mainstream tool for simulating the growth of plants and plant-like objects in computer graphics

Inspiration for Wong's method came in part from studying the work of Archibald Christie (1929):

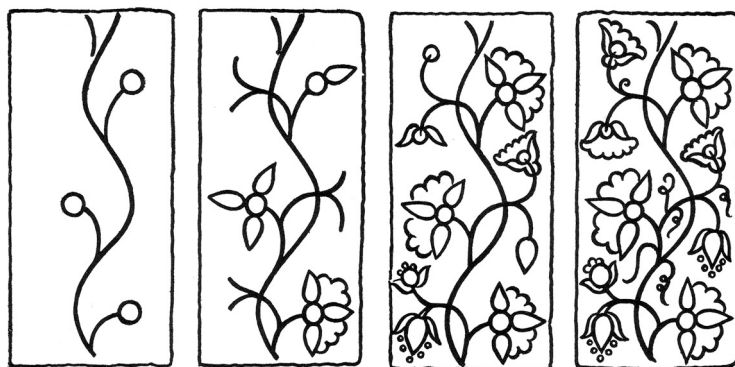


Figure 16: Stages in construction (Christie, 1929)

Design employs a divide and conquer strategy that begins by partitioning a canvas using a medial axis transform. The result is used to locate the major symmetries. The main stem of ornament is drawn using this as a guide. Circles are spawned left and right of the stem as proxies for subsequent ornament. These boundaries are inflated to the largest size

that still fits the spaces. As spaces reach sizes suitable for each level of refinement, decisions are made as to what best fits, detail is generated, and the process is repeated, growing branches, and eventually terminating with flowers, fruit, leaves, or whatever terminus is to be placed. This work encouraged the thesis of Craig Kaplan (2002) on geometric ornament. By means of operations over Frieze and wallpaper transformations on geometric shapes, his work algorithmically generates many of the intricate and beautiful patterns in Islamic architecture.

Andrew Glassner developed a method similar to Peter Cromwell's. This involved a grid layout and constraint directed assembly of tiles "puzzle pieces." Glassner indicated that the *Knot Assistant* is intended to assist someone with exploring design ideas. He goes further in expressing that the rendering of the knotwork is much easier to do by hand than by computer (Glassner, 1999a, 1999b, 2000).

Another approach to the problem of arranging tiles that uses graph grammars was taken by Frank Drewes (2000). He developed a set of rules that generates a planar graph that codes the plait of J. Romilly Allen (1903). The rules are implemented for parallel elaboration by an L-system. The following is an example of his production rules:

$$\begin{aligned}
 C &\rightarrow d[L,C,R] \\
 L &\rightarrow l[L] \\
 R &\rightarrow r[R] \\
 C &\rightarrow c \\
 L &\rightarrow e_l \\
 R &\rightarrow e_r
 \end{aligned}$$

C is the center (starting point) of a plait, $d[L,C,R]$ expands this by another center tile flanked left and right. $l[L]$ continues left development. $r[R]$ continues right development.

C is eventually constrained to a corner to become a chevron, c . L is eventually constrained to a left edge tile, e_l . The analogous termination on the right side is a right edge tile, e_r . Derivation of a complete plait follows from its center with a graph acting as the organizing structure as opposed to a grid/array:

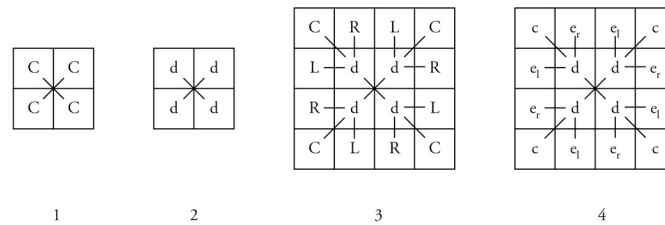


Figure 17: Square plait derivation (Drewes, 2000)

When development is finished, codes are replaced by their corresponding tiles. Rotations of subassemblies can be coded, as can holes. This leads to some variety in design. However, expressiveness is again limited by the fixed size of the tile set.

Aidan Meehan illustrates the assembly of a grid and its subdivision for knot design with what appear to be computer-plotted drawings in his latest book (Meehan, 2003). The software is not identified. As shown in Figure 18, the tri-grid method begins with the grid in step 1, adds a subdividing grid in step 3, and then diagonals to form lozenges as in Step 4. One draws and closes curves on the diagonals. Lines drawn on the subdivision grid denote breaks. We are not informed of the underlying representation(s) if this is in fact computer generated. Development of his version of the grid is done as shown here:

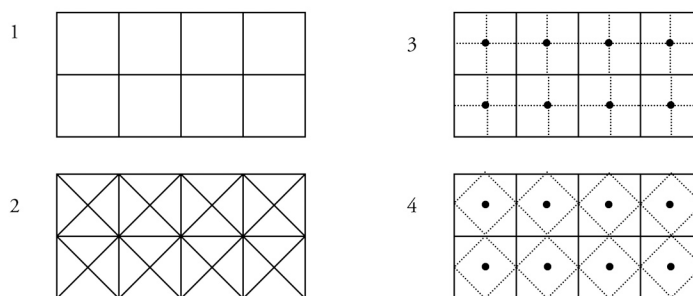


Figure 18: Aidan Meehan's tri-grid (1991, 2003)

Mathew Kaplan and Elaine Cohen presented an implementation based on Christian Mercat's curve crossing labeling method, new algorithms for curve smoothing, and means of integrating imagery for more complex features such as the anatomy of zoomorphs/animal designs (Kaplan, 2003). An artist first designs a graph for the knotwork after which the program performs a modular installation of interlace and then smoothing to improve aesthetic appearance. Kaplan demonstrated reconstruction of some of the most complex interlace designs with a portion of a carpet page (folio 94v) from the Lindisfarne Gospels and the T from folio 124r in the Book of Kells. Mark making remains an issue and editing tools are required to allow an artist to intervene on details of curve generation to resolve specific defects that occur during inflation of guide curves in regions of high curvature.

The Masters thesis by Hunter Parks (2003) developed a method based on tileable spline patterns. This was done in Maya from which he builds Renderman shaders that can be applied to surfaces.

An Honors thesis by Walmisley (2004) implements a tiling approach.

The first major extension of George Stiny's work on shape grammars comes in the Ph.D. thesis of Iestyn Jowers (2006), where he begins development to support freeform curves. He includes a small example of Celtic knotwork synthesis by means of a shape

grammar. The method sequentially applies curve segment recognition and replacement rules and is very constrained at this early stage of development. [A summary of the complete method begins on his thesis page 156.]

The Ph.D. thesis of Keith Wiley (2006) presents an alternative labeling method to that of Christian Mercat for representing interwoven surfaces. While being a crossing labeling method, it works in 2 ½ D allowing occluding edges of surfaces to be correctly depth labeled. Because it operates on the edges of crossings surfaces, it requires four annotations to describe a crossing compared with one in the Mercat method. The additional complexity unloads some of the hidden line removal burden from the renderer and allows the interweaving of complicated surface structures.

For his Honors thesis Fung (2007) implemented a reflection algorithm that credits Peter Cromwell. This paper leads indirectly to Gerdes and preliminary work on the formal foundations of mirror curves by Jablan.

So, we have a collection of implementations that are consistent with the five modern methods. Matthew Kaplan and Elaine Cohen's work represents the most advanced of these integrating new curve smoothing methods with. Preferences seem to favor tiling.

2.8 Composition and creative design theory

We see with the mind.

Igor Yevin (2002) presents the conjecture that aesthetics might be encoded as critical brain states. He argues that adding to or removing anything from a work of art that gives rise to such a brain state results in its collapse. In looking for a quantitative way to measure brain state, Yevin records the results of an eye movement (saccade) study of observers of an impressionist painting. Along with variations from subject to subject,

merging their results revealed a common sub-path. In other words, each of us shares a common basis for interpreting what is presented at our visual cortex.

Laurent Itti (Itti, 2004) reported on the use of a computational neurobiological model of visual attention to synthesize realistic eye and head movement for a synthetic actor. Based on a verified neurobiological model of the primate vision system implemented by means of artificial neural networks, Itti's application processes the foveal field of the retina according to the transforms performed by the optic nerve and the fusion of these results that occurs at the visual cortex. His model of visual attention produces a salience map based on these data.

In the early '90s, Remko Scha was confronted with mathematical models proposed to assign a "beauty coefficient" to visual patterns (Scha, 1993). Yielding to reductionism, this single number was intended to describe how much aesthetic pleasure was evoked by any given pattern. Scha invoked Kant to observe that this isn't so easy. He observes Kant telling us that aesthetic judgment is more likely "an intersubjectivity." An evaluation related more closely to similarities between the cognitive machines that each of us possess than to an objective measure. He proceeded to evaluate three theories for measuring aesthetics, one by Birchhoff (harmony), another by Bense (information theory), and a third by Leeuwenberg and Prangnanz (gestalt). All of the theories provide a measure of orderliness. This interpretation of the Gestalt provided a measure in terms of the length of a turtle code required to draw a result, with the shortest code winning. Scha returns to Kant to say that all three of these models are too static and that something concerned with the perceptual processes themselves is needed. It would also be convenient for the encoding to capture the sense of ambiguity. Scha concluded that the semantics of these things can't be worked out and that the primary benefit of these measures is what they may suggest about what a robust model contains. The hope for a neurobiological model comes closest to what Scha was

looking for. A salience map provides an encoding that represents a fusion of measures believed to be relevant to a given task. Interpretation and use of this map remains an area of active research since we do not yet understand the deep structure of what it means when attention is directed to something in such a map.

The connection to Yevin's conjecture continued with Anthony Santella's Ph.D. thesis (2005) on abstraction in non-photorealistic rendering. The problem centered on how one decides which details to keep and which to discard when interpreting photographic images and visualizing scientific data. The importance of abstraction to scientific visualization is how to effectively communicate only what is relevant without becoming lost in detail.

In reviewing the literature, Santella found studies comparing brain activity and eye-movement in skilled and novice artists (Solso, 2001), eye and hand movement in skilled artists (Miall, 2001), and eye-movement in radiologists reading tomography for breast cancers (Mello-Thoms, 2002). These studies report that brain activity and eye-movement in expert performance is different from that of novices. Expert eye fixations are precisely targeted. In the case of radiologists, even when cancers were missed, eye fixations were recorded at the locations of these cancers in the imagery.

Santella's idea was to use the eye fixations of skilled artists to identify the most important details in a given photographic. Detail everywhere else in the photograph would be reduced to produce an abstracted image. What was observed was that images produced using the eye fixations of skilled artists were very similar and effective in producing the desired result. Correlation studies on the data collected in Santella's evaluation experiment included Itti's neurobiological model as an independent observer and this is why we understand that cognitive interpretation of the salience map remains an open issue.

The best predictor of human visual attention was found to be another human with similar expertise in the given task. We may conclude that the present thesis will need to rely on human judgment as to whether artistic objectives are achieved. The algorithms themselves will be without an automated means of judgment or introspection.

A study conducted by Peter Stebbing (2004) was designed to identify *the core components of a visual organization grammar*. This question arises in what to teach in art school. In surveying 50 textbooks on visual composition, the frequency of occurrence of specific terms was observed to be as follows (p. 63):

-	Contrast and synonyms	212
-	Rhythm and synonyms	131
-	Balance and synonyms	57
-	Movement	57
-	Proportion and synonyms	48
-	Expression	41
-	Unity and synonyms	37

The author concludes that there are four fundamental organizing principles involved in visual composition: contrast, rhythm, balance, and proportion. Intercultural studies further suggest that these concepts may be universal.

To this point we have used the term art and design almost interchangeably in regards to things visually Celtic. At this moment we need to differentiate the terms. Craig Elimeliah (2006) provides a simple rule for telling the two apart. Art is intended to produce a purely aesthetic experience and doesn't adhere to any specific rules. Design on the other hand, is intentional. To design something means to go about the process of creating

something very specific with a certain outcome or effect in mind. To create something Celtic means that we are working with intent. What we are about is design. Note that this is a contemporary view and there are those in the visual arts who disagree with this distinction and base their defense on classics (Donis, 1973). We will assume that what we must engage in is creative design and search out its principles. The only important thing that I might stress is that we are not about the process of designing toasters, GPS navigators, skateboards, A380's, or anything of that sort.

This review will summarize discussions by Donis, who we have already mentioned, Robert Lawlor (1982), Wucius Wong (1993), Kimberly Elam (2001), Gail Hanna (2002), and Christian Leborg (2006).

There is a notion of space, positive and negative, white on black and their reversals, and a notion of ground, where one color is recognized as the background. Whether there is stress perceived in an arrangement depends upon whether or not symmetry and balance are aligned with the vertical (felt) axis. Horizontal and vertical directions suggest stability, peace, or well being. Diagonal arrangements are the most unsettling. Graceful curves may suggest movement and warmth. Jagged ones may suggest anger or chaos. Tones may suggest light or darkness, texture, or provide depth cues. A middle gray may suggest rest. These elements together with scale can be used to create contrast by establishing relative relationships. Repetition if carefully crafted can create a sense of rhythm. All of these methods are focused on the objective of invoking an emotional response on the part of the viewer.

As for what is being arranged, we have familiar elements. There are points or dots as the most basic unit. As we add points the mind begins to assemble them, searching out relationships. From chaos may emerge pattern and shape. It is this effect of black on white that is used to produce gray tones for printing. It is also the property that gives recognition

to illustrations drawn in the old Wall Street Journal. It is this process of self-assembly that is known as gestalt.

In contemporary design we have settled on two organizational notions, the unit form and the structure. A collection of shapes all the same is referred to as unit forms. They are most often arranged in rectangular, circular, or radial grids. These grids are referred to as structures. The purpose of structures is to assist in establishing repetition or the implementation of other operations over these unit forms. These can include such things as overlap, interpenetration, union, subtraction, and occlusion. Progressive change in the direction or some other characteristic of a form or structure, such as size, shape, or color, is referred to as gradation. Forms or structure departing from a common origin is referred to as radiation.

Structures can interact with forms. Some grid cells may be opaque and cause forms to be cut or occluded. When this happens the structure is called active. Passive structures simply help guide arrangements. Structure can be visible or invisible. Structures can be formal being composed of exact squares all the same size. Semi-formal structures have slight irregularities. A semi-formal structure might appear as follows:

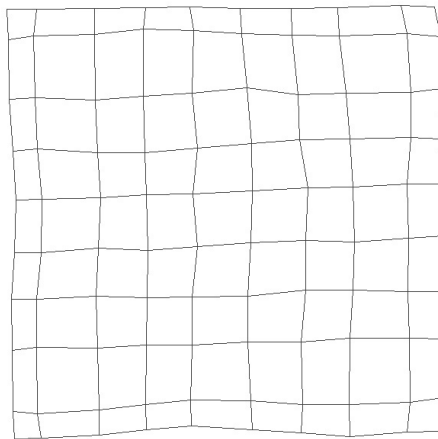


Figure 19: Semi-formal structure

Of all the ways we can arrange forms within a structure, include translating, rotation, scaling, skewing. Of course we can also do stretch and squash. The most important distillation comes from patterns and tiling: translation, rotation, reflection, and glide reflection, from which we can create all of the classical arrangements known as wall paper patterns (Alexander, 1975). Glide reflection means that not only do we reflect a form across a line, we also translate the reflected form along the direction of the line by some amount. From this point it is very much up to the imagination as to what is or needs to be expressed.

What we recognize at this point is that Celtic design most commonly involves a formal invisible structure called the grid. It is active in that it determines the pattern of interlace. Forms are recognized as the curve fragments of a tiling. Forms are also of animal design or tilings whose shapes are distributed in pose over a structure.

The creation of forms is also closely linked with classical ideas. One such idea is the golden mean/ratio, considered to be a measure of idyllic proportion. The golden rectangle is found by bisecting a square and drawing a circle as in the following diagram:

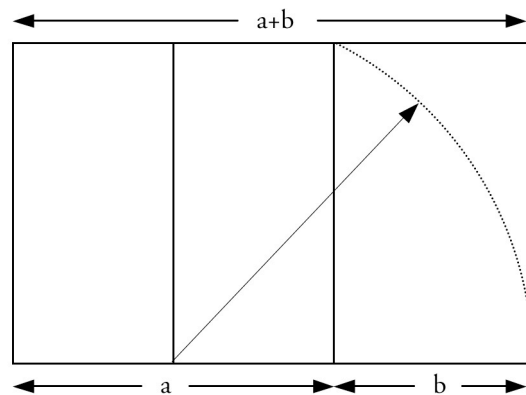


Figure 20: Golden rectangle

The ratio of the side of the square to the width of the extension equals the ratio of the rectangle base to the side of the square. The Greeks used this proportional relationship

extensively in their architecture. As we mentioned earlier, Stevick has discovered that many pages in the insular manuscripts have proportions guided by this number:

$$\frac{a+b}{a} = \frac{a}{b} = \varphi = \frac{1+\sqrt{5}}{2} \approx 1.61803\dots, \text{ an irrational number.}$$

What we next discover is that in addition to nested curves of diminishing radius, as shown us by George Bain, spirals are also found to be assembled using the golden ratio or progressions based on it. The following Celtic spiral is one such example:

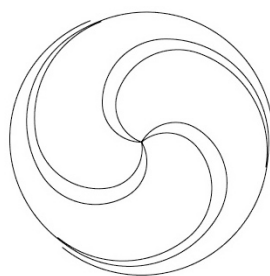


Figure 21: Celtic spiral formed using the golden ratio

Formed from a geometric progression of quarter circles based on the golden ratio (Elam, 2001, p.25), this spiral is terminated at the outer intersections and the narrow bands later filled to complete the design. The basic lines consist of genetically generated spiral curves from a genome that solves the intersections and finishes the outer spiral.

More exotic curves are found in the arcs of the Chi on folio 34r. Retreating to the studies of Robert Lawlor (1989), we discover that a varied assortment of complex compound curves can be shaped using the golden ratio as a starting point. The shapes we seen in the Chi-Rho page should be eminently approachable with this guidance. What we would need is a foundation for accurately designing such curves.

We next turn our attention to what is formally known about Celtic knots.

2.9 Mathematical foundations

This section provides a summary of what is formally known about Celtic design. Several straightforward extensions are made for tangles.

As we return to the lozenge in folio 34r of the Book of Kells, the thing most striking about it is that it contains no knots. If we examine the design, we will notice that all of it is made of open curves, the snakes and beasts most notably, and thinking about birds and humans being modeled as star like structures with extremities forming open curves emanating from central cores. In the vocabulary of topology, the design consists of tangles and not just any kind of tangles. These are alternating tangles satisfying the consecutive over-under requirement imposed on Celtic design. The interlace appears to work. But does it always?

This question was formally posed by H. A. Thurston (1954) about Celtic interlace in general. As long as the interlace is composed of knots or links, both of which consist of closed curves, the answer is that an alternating interlace always exists. To show that this does not hold for arbitrary tangles, his Figure 10 is shown below:

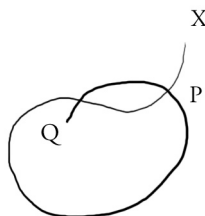


Figure 22: Tangle without alternating crossings (Thurston, Fig. 10)

For closed curves there is a further restriction in that crossings may consist of only two curves at a time. The curves must also cross rather than just touch at a single point.

Assuming these conditions and inspecting Thurston's Figure 10, one might conjecture that alternating crossings might exist if the number of self-intersections of the tangle is odd. We can present this problem in terms of Conway's tangle operators as shown in the following figure:

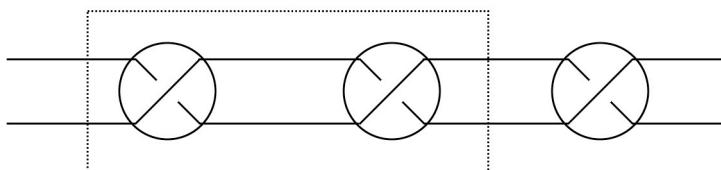


Figure 23: Tangle with odd number of self-intersections

We notice that an alternating solution exists for three self-intersections. By induction on the pair of operators on the left enclosed in the dotted rectangle, we know that an alternating solution exists for tangles containing an odd number of self-intersections.

We next have the question of how this is affected by other links and tangles in a design. For this we examine the following figure:

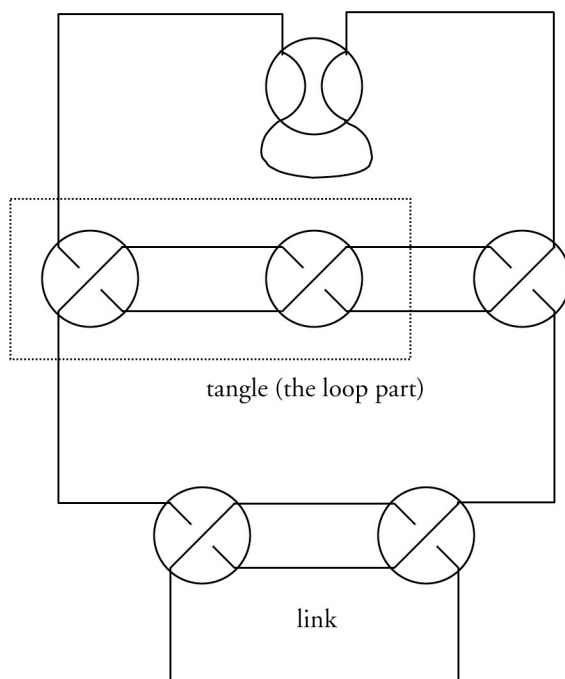


Figure 24: Tangle with odd number of self-intersections, crossed

We clarify the problem a bit at this point by closing the tangle so that we don't confuse it with anything else. We then show a crossing by another tangle, again closed to prevent confusion. As long as this second tangle threads the loop of the first in the same way that a knot or link would, we see that an alternating solution exists.

Another point by Thurston is that the alternating solution is found by sequential solution of the crossings. One has to have the knotwork in place before one can solve for the alternation.

Something for which I have only a conjecture at this writing is that if a link fails to solve, one need retry the solution one more time with the opposite initial crossing state. An initial failure can occur due to other curves in the design that have not yet been solved being mixed with ones that have. If the retry fails, then there exists no solution. The reason for the failure will be the presence of a tangle that has not been properly designed or crossed.

Peter Cromwell (1993) studied symmetries in Celtic knotwork. This is made possible by the regular lattice (grid) on which most designs are based. In borders and thin strips, patterns are formed where break markers appear. This occurrence is often repeated and regular intervals. He describes such patterns as locally periodic. This gives rise to translational symmetry. Transition of this to idealized friezes is interrupted by the interlace.

His analysis was carrying out in terms of symmetry operations, namely, the four isometries of the plane. For planar frieze patterns, he found only twofold rotation, reflection in the centerline, reflection in a line perpendicular to the centerline, and glide reflection along the centerline. He also notes that compound motions can also serve as symmetries. A screw is a rotation followed by a translation along its axis. A rotary reflection is a rotation followed by a reflection in the plane. He found that both of these can act as symmetries of an interlaced pattern.

His analysis reveals only 10 classes for interlace, some of which are not found in the field. The alternating crossings eliminate a majority of cases normally associated with Friezes. He also examined the question of whether the placement of breaks could provide a means of classification. He found that break markers associated with Friezes were discrete but not necessarily transitive.

Cromwell (2004) also presented the results of an enumeration study of the distribution of small knots derived from plait work. He provides tables of the results and diagrams of the knots. He also observed that only about six of the knots appeared to be used on a regular basis. The enumeration analysis showed that the way in which crossings were split in plait-work formed a constraint that was consistent with this observation.

Paulus Gerdes (1990) is credited with recognizing that interlaced patterns can be generated by reflection or what is now called *mirror curves*. As part of continuing research, Slavik Jablin (1995) summarizes the numbers of prototiles required for the construction of all such curves given the number of sides of the enclosing polygon:

-	Regular triangle	3
-	Square	5
-	Hexagon	11

The two central questions on square tilings were (1) the placement of interior generating mirrors for a specific curve and then (2) the classification of the resulting curves. An important feature in Celtic design is the unbroken cord. The cord generation method of the enumeration study provides rules for finding single curves/knots:

- A mirror introduced into the crossing of two distinct curves connects them into single curve.
- A mirror introduced into a self crossing, depending on its position, will cause one of two things happen, either the curve will remain unchanged or the curve will be broken into two closed curves.

Adding a single cell to the enclosing grid preserves single curves, in other words while the curve becomes extended it will remain unbroken. It is noted that there are other transforms that also have this property. They are not described.

The symmetries of curves having strikingly similar forms to those in Celtic design that are explained by the same method have been found in the threshold patterns of the Tamil in South India (Gerdes, 1989).

Christian Mercat's (1997) tetravalent graph provides a general labeling method and algorithm for discovering a valid ordering of alternating crossings in Celtic interlace. Properties of tetravalent graphs are developed in his Ph.D. thesis (Mercat, 1997).

John Merne (1931) presents an ontology of shapes with derivations to different forms observed in the archeology of Celtic design. He then proceeds to demonstrate the derivation of knotwork by transformations and compositions on forms in this ontology. The derivations may be formally described in terms of a shape grammar.

Developed by George Stiny (1975) a shape grammar is a formal production system where a design is represented by a form and a set of shape replacement rules is applied to modify the design. Developing shapes by this approach is referred to as shape computation. The simplest form of shape grammar is a fractal, where the initial shape may be a line or polygon. One can then define a rule that replaces an edge with a different shape:

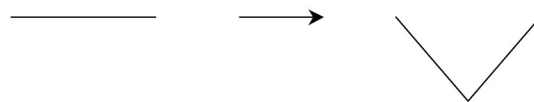


Figure 25: A simple shape rule

This specific substitution evolves through repetition into the Harter-Heighway dragon curve. Formally we would write that a shape grammar is the four-tuple $\langle V_T, V_m, R, I \rangle$, where V_T is a finite set of shapes (terminals), V_m is a finite set of shapes such that everything derivable from V_T , V_T' (terminals), is distinguishable from everything derivable from V_m , V_m' (non-terminals), R is a finite set of production rules $U \rightarrow V$ where U and V are in V_T' or V_m' , and I is a shape formed from the composition of shapes in V_T' or V_m' . Note that this is modified definition slightly to accommodate Jower's thesis.

The research of Gwen Fisher and Blake Mellor (2004) begins the study and enumeration of Celtic knot and link patterns in plaits. A $p \times q$ panel where p and q have no common factors will contain only a single strand; otherwise the number will be $\gcd(p, q)$. If a plait is treated as a band and bent into a circle/border, the number of strands will be $\gcd(2p, q)$. The concluding result is that the number of strands in a $p \times q$ L shaped frame of width 2 (cells) will be $\gcd(|p - q|, n)$. A study by Jacobsen (2002) addresses the computerized enumeration of alternating tangles.

Peter Suber (2004) presents a formal predicate notation for describing the tying of knots. This development was motivated by the difficulty in describing these procedures clearly where existing notations fail. For an overhand knot the procedure would be expressed as follows (p. 60):



Figure 26: Overhand knot (actually a tangle--black is below, white is above)

1. Hold(StandingPart, LeftHand), Hold(RunningPart, RightHand) -- Start with the standing part in the left hand and the running part in the right hand.
2. MakeLoop(RunningPart) = LP.1, CrossOver(RunningPart, StandingPart), Plane(LP.1:HorizontalPlane)-- Make a loop in the running part, name it loop #1. The running part crosses over the standing part. Loop #1 lies in the horizontal plane.
3. Reeve(End.RunningPart, LP.1:Down-Up) -- Reeve the end of the running part through loop #1 from down to up.
4. Move(StandingPart^:Left, RunningPart^:Right) -- Pull the standing part to the left and the running part to the right.

The abbreviated notation for this appears as follows:

1. GP(SP, LH), GP(RP, RH)
2. ML(RP)=LP.1, CO(RP, SP), PN(LP.1:HP)

3. $RV(E.RP, LP.1:D-U)$

4. $MV(SP^{\wedge}:L, RP^{\wedge}:R)$

While the notation is immediately attractive and suitable for exploration by genetic programming, we can also visualize a pair of cords having a conversation with this notation forming the basis of planning a local interaction. However, Suber is careful to tell us that the notation is not complete. It is not capable of forming decorative knots such as sinnets, beackets, and turks head. The turks head knot is similar to the Celtic circle pictured earlier.

Keith Wiley (2006) provides a labeled knot diagram capable of representing the interweaving of 2.5D surfaces. 2.5D surfaces are those that extend above and below the plane without having thickness, such as found in vector drawing applications such as Adobe Illustrator. The method is similar to Christian Mercat's in handling the edges of surfaces. Mercat's work is not mentioned. The reason for this is that the research focused on improvements that might be made to existing illustration programs rather than on Celtic design. The advantage of this method is its formal development.

Cameron Browne (2005) proceeds into the esoteric extreme with an examination of how to construct Cantor knots. The Cantor set resembles the anatomy of an animal and this paper examines compromises in the alternation that may be required in some instances. His follow on study (2006) involves the fractal expansion of Celtic knots.

We have briefly examined the questions of the existence of alternating solutions to interlace and tangles, labeling methods and algorithms for finding these solutions, and some of the fundamental results from the theory of alternating knots and tangles.

2.10 Complex systems theory

In this section we provide an overview of Complex Systems theory and its associated methods in light of what we now know about Celtic design. Complex systems theory is engaged in the search for patterns in what would otherwise be considered chaotic systems. It is often described as a search for order at the edge of chaos. The modeling of these processes characteristically follow a unifying concept of simulating the interaction of behaviors over assemblies of discrete elements. The organization or topology of these assemblies, element mobility, connectivity, communication, and synchronization, all affect the possible outcomes.

The key idea in complex systems theory is that local interactions give rise to organized global behavior. This quickly leads us to recognize that many of the things around us are exotic. A waterspout that forms over lake Pontchartrain (NOAA, 2007) evolves from the unstable dynamics of aerosols (dust and water vapor) interacting with direct and reflective solar heating of earth, water, and atmosphere. Rogue waves arise by means of constructive reinforcement and with unexpected frequency in the oceans and great lakes. These waves present serious challenges to marine engineering. Time series analysis of more than six years of observations where they often occur still leaves us with no clear understanding of how they behave (Liu, 2005). In the life sciences the ribosome is a molecular machine that reads nucleotide sequences to assemble the various proteins needed by the cell. While the structures that the ribosome assembles get it right, current methods in computational molecular biology predict the correct folding of these structures only about 70%-80% of the time (Jones, 2007). Turning to structures assembled from proteins, the embryo begins as a single cell then subdivides. As subdivision repeats, by means that are still not understood, cells begin to differentiate to assume specialized roles in what will

eventually become a larger functioning organism. Each cell behaves in a non-trivial way, from how they respond to chemical messages from to how they use molecules internally for regulation and communication. Considering larger organisms, ants will organize themselves around cracks in pavement and dig elaborate tunnel systems. They use chemical tracks to guide others in the colony to the latest morsel and will improve the design of these paths (Bonabeau, 1999, p. 26). Birds flock in elaborate formations and guide their flight using the Earth's magnetic field for reference (Davis 1981, Toner 1998). Bees dance as part of describing how far it is to a fresh patch of flowers, use sun angle for directional navigation, and recognize flowers by the way they reflect UV light (Brisco, 2001). Social systems involving competition and cooperation are common among living organisms, fish, insects, mice, and the most complicated of all, people. Every time the Federal Reserve Chairman voices a position on the economy, investors and analysts form opinions based on how they perceive that position might affect their strategies. We observe the tangible effects of these perceptions in how the markets move. This behavior may not be repeatable, as something special goes on when people are involved. We learn and we apply what we learn to what we do next. In testimony before Congress, former FOMC Chairman Dr. Allen Greenspan, cited this as a source of frustration. His comment was that every time a new model was developed to predict the behavior of some part of the economy, investors and analysts would incorporate that understanding into their own analysis and the behavior would change (citation). In computational neuroscience, Laurent Itti and others have discovered that if one accurately models the microscopic characteristics of the neurology in the primate vision system, the resulting simulation naturally exhibits many of the expected behaviors of early vision. Anthony Santella's thesis on the relationship between eye movement and the creation of visual abstraction shows us that this modeling of early vision is only the beginning of how an artist perceives what is seen (Santella, 2005). What all of these examples share in

common are individual components that at some scale are governed by local rules extending from simple logic, to message or symbol passing, to local supports for non-linear dynamics. The process of interaction between these components results in organized behavior. We notice that outcomes are not predictable and that the order we find may be fragile as it emerges from the chaos. We also see that some of these systems have richer capabilities than others.

2.10.1 Complex Adaptive Systems

Complex systems capable of exhibiting learning and adaptation form a special subclass called Complex Adaptive Systems. Organisms that are aware of their surroundings and exhibit the ability to make conscience choices are clearly in this category. The ability of organisms to adapt to changes in their environment includes a broader class. Adaptation includes the ability to make adjustments to immune systems to overcome disease. More fundamental changes take place at the genetic level over longer periods of time accounting for such things as physiological adaptations.. These robust characteristics of biological systems would lead to an obsession with searching for similar capabilities in computer science. von Neumann was able to find self-replicating automata (Ganguly, 2003). Cellular automata became a center of attention when rules for a simple Sun Microsystems screen saver called the game of life were observed to exhibit the ability to replicate patterns and eventually Turing equivalence (Heudin, 1999, pp. 6-16). Computational analogies to biological genetics allow evolutionary study of abstract structures. Genetic algorithms are now commonly used for optimization in engineering analysis where advantage is found in their ability to escape local minima.

2.10.2 Cellular automata

Cellular automata are represented as n-dimensional arrays of machines. Each machine within the system handles changes of state of some process based on assessments made within a limited radius of its location within the array. Toffoli (1987) describes the simulation of a number of physical systems involving spatial and field relationships such as diffusion, fluids, and gasses, with these systems. In Toffoli's example machines are programmable although homogeneous on the CAM-1 architecture. Characteristic of these models are that they represent locally restrictive finite element simplifications. Often, however, as in the case of Margolus neighborhoods--2x2 blocks of cells that model particle collisions and phase representation such as in lattice gases--they may also contain elaborate topological constructions and groupings. While macroscopic behaviors of gases and liquids may resemble their physical counterparts, unexpected properties such as the anisotropy that occurs in simulations of fluids on square grids may also emerge as side effects of these simplifications (Buick, 1997). In such cases, refining the topology of the array, allowing local behaviors to be more robust, may provide relief.

Melanie Mitchell (1996) shows us the evolution of machine programs to explore behaviors. Given a fitness function, evolutionary computation can be used to search for programming that exhibits specific behavior. We recognize the combination of a genetic algorithm with an evolvable computing element as being adaptive and capable of exhibiting complex behavior.

Niloy Ganguly (2003) provides a survey of cellular automata. While finding many of approaches to characterizing these machines, he arrives at the conclusion that they can be divided into two broad categories: (1) additive or linear CA where the next state is defined in terms of simple logical expressions and (2) nonlinear CA which are not. Stephen Wolfram summarizes a wide variety of specific machine types and control strategies that can be

applied (Wolfram, 2002). While a careful ontology is important when the objective is to attempt to prove specific properties of specific machines, in the present instance we assume that cells of arrays are as powerful as they need to be for the modeling of a process in a given experiment.

1.10.2.1 Graph cellular automata

A simplified model of the graph cellular automata (GCA) is introduced by Jidong Chen, et. al., (2006) as a means of modeling the behavior of traffic flow. Formal properties of the representation may be found in O'Sullivan (2000).

In the Chen study a road network was modeled. Nodes of the graph represented intersections and edges represented road segments without intersections. The general model was modified such that each edge is a cellular automaton whose length consists of a finite number of cells connecting the distance between two intersections and whose width is equal to the number of lanes. Vehicles were represented by symbols whose movements were defined by the rules of the simulation.

The analogy would be, of course, for the Celtic design grid to be modeled as a GCA for behaviors that may be modeled by the movement of symbols along the edge of a graph. This immediately reminds us of Christian Mercat's method.

Note that we can implement graph cellular automata in terms of a heterogeneous collection of behaviors (vertex, edge, and face) defined on Margolus neighborhoods of any sufficiently large array. Support rules would limit symbol behaviors to the region of the array simulating the graph.

1.10.2.2 Lattice gas automata

Stephen Wolfram (2001) describes lattice gas automata as cellular automata fluids and traces their origin back to molecular modeling of gasses in the late 1800's and the early discoveries of large scale fluid behaviors by Alder and Wainwright using idealized molecular dynamics in 2-dimensions.

The lattice gas automata (LGA) was formally proposed in 1976 by Hardy and de Pazzis. It was based on the reduction of molecular movement in a square cell to sixteen discrete possibilities. Rules governing the modeling of particle collisions within a cell could be implemented as simple logical operations on the boundary conditions for that cell. The basic interactions allowed in HPP model are summarized below in a depiction distilled by Medvedev (2002, p. 9):

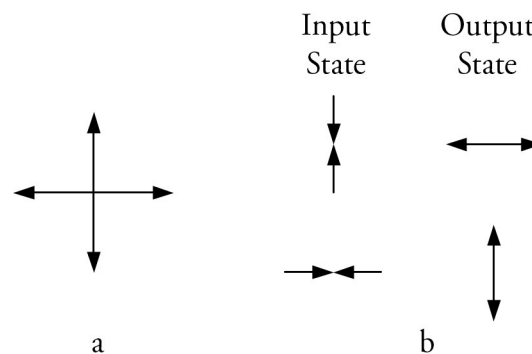


Figure 27: Rules of the HPP Lattice gas model, possible velocity directions (a) and possible collisions (b)

In the present research we are able to use the lattice gas automata to model design behaviors that can be simulated by a particle interaction. The case that immediately presents itself is the reflection model by Gerdes. Rules for the reflection model are derived in chapter three. In some cases the behaviors of graph cellular automata can be reduced to

lattice gas automata. A model of Christian Mercat's method will also be derived in Chapter three.

2.10.3 Agents

Where mobile cellular automata are able to approximate mobility, an agent is able to simulate structures with unrestricted spatial mobility. Rather than attempt to define the notion of an agent in terms of sophistication of behavior, I defer to Stuart Russell's first and most basic definition, that an agent is something that acts (Russell, 2002). We can then describe how this comes about in terms of architecture. An agent is a system that receives percept information about its environment by way of sensors. To this information it applies means for analysis and interpretation to derive an internal state, conclusions, and courses of action. It then carries out a course of action on the environment by way of actuators. We can think of agents abstractly as models of systems or individuals interacting with the world around them. For example, a population of agents constrained to the vertices of a lattice and allowed to communicate with each other only as a function of an adjacency relationship are able to simulate the behaviors of cellular automata. The agent architecture can be represented schematically as shown in Figure 28:

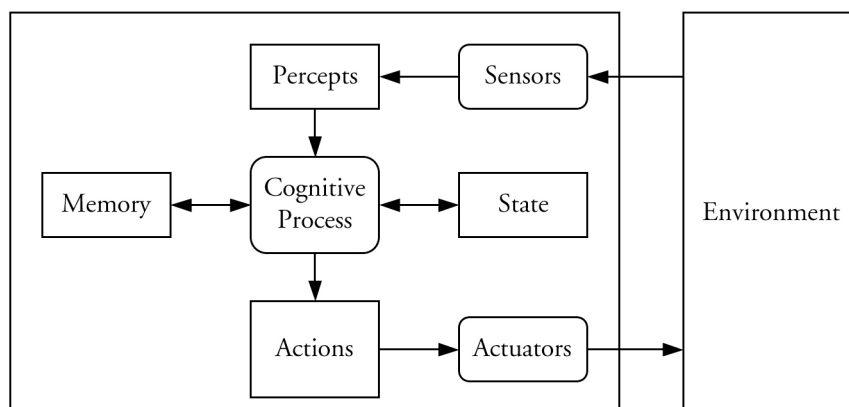


Figure 28: Agent Architecture

Stuart Russell defines categories for agents in terms of the power associated with their internal models and processing:

- Table Driven Agent – a system where a percept or percept sequence is used to index actions in a table.
- Simple Reflex Agent – a system that employs simple rules to derive the next action from the current percept.
- Model-Based Reflex Agent – a system that adds a model of the environment that assists a reflex agent by constructing internal state descriptions of the environment from a percept stream.
- Goal-Based Agent – a system that adds goal seeking to a model-based agent allowing it to derive plans involving actions that can be carried out in multiple steps. This involves predicting and interpreting the meaning or consequences of actions as part of determining what to do next.
- Utility-Based Agent – a system that adds an affective measure of effectiveness to a goal-based agent that allows the evaluation of plans in terms of satisfaction with their outcomes.

While Russell organizes these basic classifications around symbolic processing, the internal components associated with the cognitive process are intended to include machine learning and statistical methods such as Bayesian networks, Markov processes, fuzzy logic, and artificial neural networks, as appropriate, for modeling percept pattern learning and response association. The agent architecture offers a flexible and systematic way of thinking

about the modeling of individuals, systems, or systems of systems, involved in studies of interaction.

2.10.4 Lindenmayer-Systems (L-systems)

The L-system (Prusinkiewicz, 1989) is based on the idea that complex objects are assembled by the successive replacement of parts of an initial simpler one. The idea is basically that you have a production system that successively replaces substrings beginning from an initial string. What is novel about the L-system is that it performs this replacement in parallel. This simple process is able to generate strings that describe the complexity of plants and other complex organic structures such as corals. The way this is done is the application of turtle geometry, where a given letter in an evolved string is given geometric interpretation. The union of all of these pieces of geometry form the structure of the evolved object.

2.10.5 Artificial Life

While learning clearly represents one form of adaptation, limitations in an individual's ability to accomplish a functional objective can also be structural. If we view the structures in a model as forms of artificial life, the use of analogs to biological genetics and competition has emerged as a means of exploring the capabilities of new individuals derived through evolutionary modification.

Genetic algorithms are now routinely applied to optimization problems in engineering. Peter Bentley (1996) demonstrated their application to the design of physical structures through the exploration of parametrically trimmed cube stackings. Karl Sims (1994) shows us how to combine motors with articulated limbs to realize synthetic machines with mobility, Rafal Kicinger (2005) shows us how to evolve bridge trusses,

Temesgen Mengistu (2004) shows us the optimization of cascades of gas turbine blades, Peter Wonka (2007) shows us land use analysis, design of urban areas, and the assembly of synthetic cityscapes for feature film.

The process of life is organized around two primary structures, the genotype and the phenotype. The genotype is an allele level coding of the specifications for an organism. In a biological organism this is its DNA. The phenotype is the physical form of the organism assembled by interpreting its specifications.

Genetic algorithms follow a life cycle that begins with the seeding of a population of individuals with initial genotypes. The phenotypes for these individuals are assembled and their behavior in performing a desired task is evaluated by means of a fitness function. The best performers are then sent through a procreation cycle, where the genotypes for new individuals are generated from the genotypes of parents selected from the best performers. Production of new genotypes is done through the operations of crossover and mutation. Crossover assembles a new genotype by copying and recombining genotypes from the parents. Mutation introduces random changes in the coding of new genotypes. New phenotypes are assembled from this newly created pool of genotypes; and the process repeats with various possible halting conditions. For example if a problem has a specific answer, the process can halt when the answer is found. If the problem is to find a good solution to a problem, the process can halt when a desired level of performance is exceeded. If the problem is optimization, the process can halt when the rate of improvement per generation diminishes below some level.

In order to prevent super individuals from biasing a search, an age limit is usually imposed. This allows diffusion of traits from this individual's genotype throughout the population without the individual dominating the solution forever.

Simulated population sizes are typically between 100 and 200 individuals with the population being trimmed to this level for each cycle. Population management strategies and related parameter settings that are observed to produce good operating conditions may be found in the thesis of Peter Bentley (1996).

2.10.6 Social agency

Ricardo Medina (2005) discusses the idea of design in terms of creativity, innovation, and social agency. Social agency provides a structure that offers an opportunity to create and observe the effects of tension between conformity and behavior. Modeling individuals as cellular beings, the dynamics of the model are produced by picking an individual at random to be active and then another at random from amongst its neighbors. With probability equal to their similarity, these two cells interact by choosing at random a feature on which the active cell and its neighbor differ (if one exists). The active cell's feature is then changed to match that of its neighbor. Random cells may revise a feature at every discrete time step according to the following rules:

- Select a neighbor at random
- Let G be the difference of features between the two individuals
- Let H be the intersection of features between the two individuals
- Given G and H change all of the differing features of the neighbor.

Features may consist of such things as occupations, preferences, opinions, awareness, mobility, health, ancestry, goals, possessions, and so forth.

The reason this becomes interesting are the tangles in the lozenge of the Chi Rho page. Each form occupies space. Each form is entangled with another. The question that

naturally rises is whether this pose evolves from a perception of purpose or interaction. Certainly the interlace represents a relationship as does the physical space occupied by the form. Were there other sources of symbiosis or discord competing as social behaviors in creating the pose? Are creatures flocking? Why are others moved to entangle? These ideas might also involve the implementation of some form of intelligence on the part the forms.

2.11 Differential geometry

In this section we will review some properties of differential geometry that we need for working with lines. Alfred Gray (1998) provides a comprehensive development of the subject. This discussion will be guided by his work. A basic understanding of calculus is assumed.

As one guides a pen across a writing surface, its nib or sharpened point traces out a line. The point of contact may be thought of as a particle moving along a space curve, α , in the plane of the surface; and we can describe a number of properties related to its motion. At any moment our particle is moving in a specific direction and speed. We refer to this as velocity and we will measure it along the length, s , of the curve:

$$\alpha'(s) = \frac{d\alpha(s)}{ds}$$

We describe the speed of this motion independent of direction as the magnitude of $\alpha'(s)$, $v(s) = \|\alpha'(s)\|$. We can describe its direction as unit tangent vector:

$$\mathbf{T}(s) = \frac{\alpha'(s)}{\|\alpha'(s)\|}$$

In terms of these two properties, velocity may also be written as $\alpha'(s) = v(s)\mathbf{T}(s)$.

A space curve will be regular if we can find its velocity everywhere (i.e., it is differentiable) and this velocity is non-zero. If $v(s) = 1$, α will be called a unit speed curve. Unit speed curves are important to us, as every regular curve can be reparameterized as a unit speed curve (Gray, 1998, p. 11).

The length of a curve from where it starts to the point of the pen drawing it is called its arc length. The length of a curve is described by the arc length function defined as

$$s_{\alpha}(t) = \int_0^t \|\alpha'(u)\| du,$$

where $u = 0$ is the place we started to draw and $u = t$ is where the nib is on the writing surface. For unit speed curves the arc length function simply describes the distance along the curve from where we began drawing. The term arc length is also used to refer the length of any non-empty sub interval of $s_{\alpha}(t)$.

The amount of turning left or right of the curve at the nib of the pen is called curvature. When the speed of a space curve is non-zero it is defined as follows:

$$\kappa(s) = \frac{\|\alpha'(s) \times \alpha''(s)\|}{\|\alpha'(s)\|^3}$$

For a unit speed curve, notice that the curvature reduces to $\kappa(s) = \|\alpha''(s)\|$, which is the magnitude of the acceleration of the curve. The normal of a curve provides a reference direction for its curvature.

$$\mathbf{N}(s) = \frac{\mathbf{T}'(s)}{\kappa(s)}$$

The property that causes a space curve to leave the plane is called torsion. When the velocity and acceleration of a curve are non-zero and positive the torsion function can be defined as

$$\tau(s) = \frac{\alpha'(s) \times \alpha''(s) \cdot \alpha'''(s)}{\|\alpha'(s) \times \alpha''(s)\|^2}$$

The last thing we need is the binormal. It can be defined in terms of the cross product of the unit tangent and unit normal functions:

$$\mathbf{B}(s) = \mathbf{T}(s) \times \mathbf{N}(s)$$

Together, $\mathbf{T}(s)$, $\mathbf{N}(s)$, and $\mathbf{B}(s)$ define the Frenet-Serret frame. For a unit speed curve a set of differential equations called the Frenet formulas hold (Gray, 1998, pp. 185-186):

$$\begin{aligned} \mathbf{T}'(s) &= \kappa(s)\mathbf{N}(s) \\ \mathbf{N}'(s) &= -\kappa(s)\mathbf{T}(s) - \tau(s)\mathbf{B}(s) \\ \mathbf{B}'(s) &= -\tau(s)\mathbf{N}(s) \end{aligned}$$

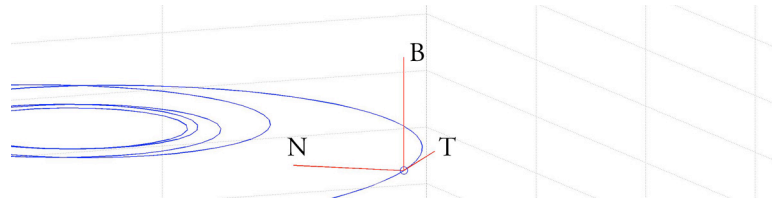


Figure 29: Frenet Frame

We next wish to employ something special called the existence property from the fundamental theorem of space curves.

Theorem: Given differentiable functions $\kappa(s) : (a, b) \rightarrow \mathfrak{R}$ (curvature) and $\tau(s) : (a, b) \rightarrow \mathfrak{R}$ (torsion), with $\kappa(s) > 0$, there exists a unit speed curve $\beta(s) : (a, b) \rightarrow \mathfrak{R}^3$ having this curvature and torsion. $\mathbf{T}(s_0)$ and $\mathbf{N}(s_0)$ may assigned as long as $\|\mathbf{T}_0\| = 1$, $\|\mathbf{N}_0\| = 1$, and $\mathbf{T}(s_0) \cdot \mathbf{N}(s_0) = 0$ (Gray, 1998, p. 221).

This tells us that a unit speed model of the curve we have been drawing can be expressed as a curvature function and a torsion function parameterized in terms of arc

length. Notice that the curvature function in the theorem is strictly positive. Our pen can change directions both right and left as we draw it across a writing surface; so, our intuition tells us we need something that releases this restriction. Since our curve is being drawn in the plane, its find space curve is also confined to the plane, $\beta(s): (a,b) \rightarrow \mathfrak{R}^2$, and its curvature functions are not restricted in this way (Gray, 1998, p. 137). To limit a space curve to the plane, we set its torsion to zero, $\tau(s): (a,b) \rightarrow 0$.

With an initial position, tangent, and normal of \mathbf{X}_0 , \mathbf{T}_0 , and \mathbf{N}_0 , and assigned curvature function $\kappa(s)$ and torsion function $\tau(s)=0$, we can render the corresponding curve by solving the following system of differential equations:

$$\begin{aligned}\mathbf{X}(s_0) &= \mathbf{X}_0 \\ \mathbf{T}(s_0) &= \mathbf{T}_0 \\ \mathbf{N}(s_0) &= \mathbf{N}_0 \\ \mathbf{B}(s_0) &= \mathbf{T}(s_0) \times \mathbf{N}(s_0)\end{aligned}$$

$$\text{with } \|\mathbf{T}_0\| = 1, \|\mathbf{N}_0\| = 1, \text{ and } \mathbf{T}(s_0) \cdot \mathbf{N}(s_0) = 0$$

$$\frac{d}{ds} \begin{bmatrix} \mathbf{X}(s) \\ \mathbf{T}(s) \\ \mathbf{N}(s) \\ \mathbf{B}(s) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \kappa(s) & 0 \\ 0 & -\tau(s) & 0 & \kappa(s) \\ 0 & 0 & -\tau(s) & 0 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{X}(s) \\ \mathbf{T}(s) \\ \mathbf{N}(s) \\ \mathbf{B}(s) \end{bmatrix}$$

2.11.1 Curvature Dynamics

If we are to assign curvature, we need a quantitative means of modeling its dynamics. A starting point for this would be the linear model commonly used in civil engineering to specify road layouts as compound curves composed of lines, circles, and clothoid splines (Dickmanns 1992, Corridori, 2004):

$$C = \frac{1}{R}$$

$$C = C_0 + \frac{dC}{dl} \cdot l = C_0 + C_1 \cdot l$$

$$C_1 = \frac{1}{A^2}$$

where

C is curvature

R is radius

l is distance

C_0 is the initial curvature when $l = 0$

C_1 is the rate of change of curvature, held piecewise constant

A^2 is the clothoid parameter

A curvature dynamics model for this research is developed further in Chapter 3.

2.11.2 Line Drawing

When we know where and how we are to draw lines (we might approximate these by piecewise assemblies of Hermite cubic splines) we need to render these such that they resemble human performance. The following method is derived from earlier student work. Our innovation in this area will be the following technique. Again our math follows the development in Gray (1998).

In the process of inflating a guide curve to produce a cord, a parallel line in the plane at a distance s to the guide curve is found as

$$c_{c//\alpha}(\alpha, s, t) = \frac{sJ\alpha'(t)}{\|\alpha'(t)\|}$$

This says we take the tangent/velocity of the guide curve at t , normalize it, rotate it until it is normal to the guide curve and scale it by the desired distance.

To inflate a cord of interlace we do this both in the direction of the curve's normal and in the opposite direction and then scale by the radius of the cord. This gives us two edges that describe the boundaries of the cord, as shown in the student's drawing of a center tile from the Sloss grammar in Figure 30.

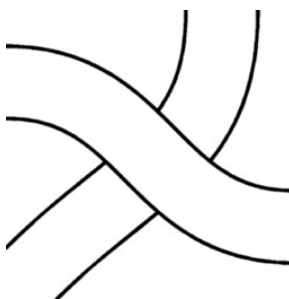


Figure 30: Inflated cord tile with VHDH quadrant coding (Sloss, 1995)

Drawing lines in this way also means we can set the radius, $s(t)$, of the cord differently for each edge. This can be used to break the symmetry between the two lines. We can also introduce an additive width to create a depth cue between two cords. The cord diameter of the upper cord can be varied to be greater than that of the lower cord. One can also apply spatial assumptions to this idea and determine a mean cord diameter based on how the cords would appear if they were 3-dimensional and being imaged in perspective projection. We can also imitate what might be seen if the cords were squashed.

For knotwork often associated with the *Lindisfarne Gospel*, the two edges derived from the guide curve are themselves treated as separate cords. This is called doubling. We can apply the same parallel curve method again to first split the guide curve

and then inflate them to create these smaller cords. The rules of interlace must also be revisited as the crossings of the new smaller cords are opposite as shown in

Figure 31.

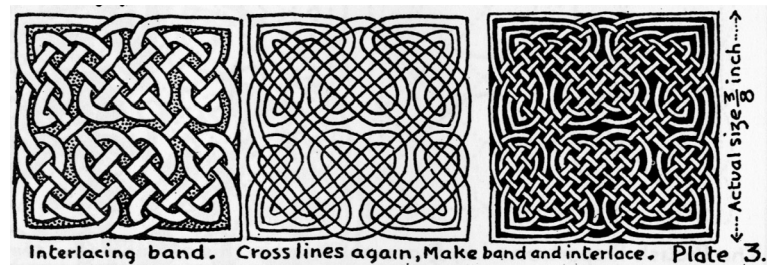


Figure 31: Doubling knots (Bain, 1951, p. 46)

2.12 Noise

Noise can be used in a variety of contexts. For example, we can use it to introduce models of natural variation. In the case of pseudorandom noise, we can employ it to synthesize artifacts simulating the effect of spontaneity. We can also use models of natural dynamical systems that evolve relevant shapes. Here we will review two models of noise, zero mean Brownian bridges and Perlin noise, as well, as measures of self-similarity.

2.12.1 Mean Zero Brownian bridges (line directional variation)

Brownian motion is used to describe random walks. A Brownian Bridge is a random walk that connects two points. Scaled between two points, the modeling of intermediate points along the path takes the following form:

$$W_t \sim W_0 + \frac{t}{T}(W_T - W_0) + \sqrt{\frac{t(T-t)}{T}} \cdot N_{0,1}$$

$$0 < t < T$$

Thijs van den Berg (<http://www.sitmo.com>)

where

W_0 known fixed starting value of the Brownian motion

W_T known fixed end value of the Brownian motion

$N_{0,1}$ standard normal (Gaussian) distributed random variable

Edwin Gerber (2005) provides another model and enumeration of random walks of length N , where a bridge is formed when N even and the number of steps to the left is the same as the number of steps to the right over the course of the walk.

2.12.2 Perlin Noise (space filling)

Perlin noise (Perlin, 1985, 2002) is a method of creating space filling textures. In two dimensions it works by forming a weighted height field for each location in an image region.

The algorithm defines a regular grid (composed of square cells or tiles) over the image region. We assign a plane at the corner of each cell. We represent the orientation of each plane by its normal. The directions of these normals are chosen at random. Perlin's algorithm restricts their distribution to a filtered Gaussian. Given a pixel in the image, we find the cell in which it is located. We calculate a weighted average of distances from the position to each of the four corner planes of the cell. The weights are based on the value of a cubic function that implements a smoothly decaying local loft that limits the effect of any one plane to a distance of 1 cell. The effect is to approximate the appearance of a cubic surface with randomly varying height.

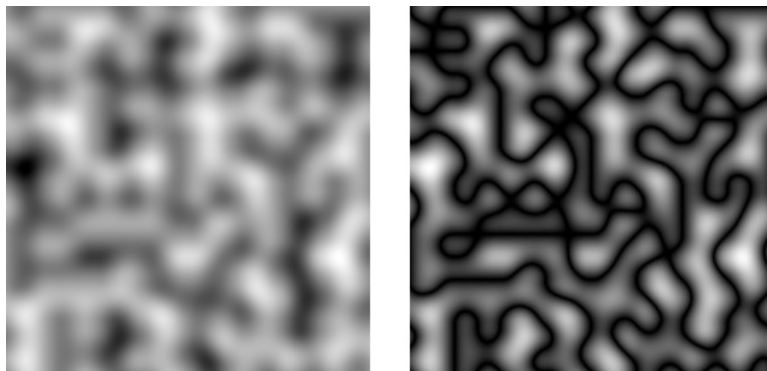


Figure 32: A Perlin noise image (left) and its absolute value (right)

The problem with noise is interpreting what it means. When one looks at a Perlin noise field with the intent of recognizing interlace, it is sometimes possible to see suggestive patterns. At this point we need to be cautioned that the reason this is so has to do with processing in the human vision and system. It is well known that a tremendous amount of work is done by this neurology (Itti, 2004), after which results are presented to higher cognitive functions for further interpretation. Thus, we are confronted with the problem of needing to provide a means of interpreting the noise, if we are to use it as a means of generating pattern possibilities for original interlace.

To this end we need at least a means of recognizing crossings. One approach would be to train an artificial neural network to recognize candidates. We could then examine alternatives that arise from associating tangle operators and connectivity with regions local to these candidates.

If this method of generating structures were successful, we would also need a means of scaling the results and placing them in compositions with appropriate orientations.

To build a training set we would need a human operator to examine a series of generated patterns and mark possible crossings interpretations. If we could build a sufficient number of these samples to employ a Probabilistic Neural Network (Augusteijn, 2002), we

might also be able to use novelty detection to determine when a particular sample is not suitable for interpretation.

Another approach with Perlin noise is to set up a gradient field whose orientation is evolutionary. The filaments in the absolute value of the noise image are zero crossings of the surface. So, if one thinks of peaks and troughs in a terrain, the problem posed for the evolutionary system becomes how to design contours at the zero crossings that appear as knotwork.

Experiments with this latter approach are explored in Chapter 3.

In emulating the characteristics of specific work, Taylor (2007) has been studying the issue of authenticating fractal drip painting by Pollock. What he noticed was that the box counting method for measuring fractal dimension (Voss, 1998) produces a number that is uniquely characteristic of Pollock's work and may be used almost like a fingerprint. Of course Pollock's paintings look like fractals. Returning to the lozenge in the Chi Rho page causes one to wonder if the writhing of the miserable beasts is also associated with a characteristic fractal dimension. We also know that wavelets can be used to provide measures of texture and pattern correlation at multiple resolutions.

2.13 Characteristics of the hand drawn line

The findings of Isenberg inform us that the quality of mark making is a differentiating factor between computer-generated lines and those that are hand drawn. In this section we review research on characterizing the appearance of handmade marks.

The interpretation of handwriting and hand drawn lines is a central problem in automated document recognition and analysis. The motivation for work in this area is the need to transfer paper-based engineering drawings and data for online access. Developing

systems that perform these transfers involves on-going research in recognition algorithms. As part of a protocol that guides the synthesis and use of test examples for automated evaluation and scoring of these algorithms, standard models of the noise types associated with hand drawn engineering drawing have been developed (Liu 2001, Zhai 2003).

One of the problems is the segmentation of circular arcs (Wenyin, et. al., 2001). Generating test images is similar to non-photorealistic rendering. Various types of noise at various levels are added to drawings chosen from a repository. The performance of each algorithm can then be evaluated and scored against each noise type. The following types of noise are used:

- Pixel noise (Gaussian, high frequency, and hard pencil simulation)
- Vector noise: taking into account the geometry of a particular shape
- Context noise: taking into account where a line is drawn

The Gaussian and high frequency components of pixel noise result from the performance of an imaging sensor arrangement, document transport, or naturally due to document wear. Hard pencil simulation may be useful before implementing a more complex model of pencil on paper. Vector noise is introduced by modulating the curvature of line. By simulating a line as a space curve with assigned curvature, vector noise can be accomplished by additive curvature noise while solving the Frenet frame. Vector noise can be used to simulate imperfections and asymmetries in line drawings. Context noise is related to the simulation of a cognitive recognition process.

Isenberg recommends work by Doug DeCarlo (2003) and Mario Sousa (2003) for improving the quality of mark making. DeCarlo studies the characterization of lines that help convey shape in non-photorealistic rendering. His method searches for zeroes in

curvature that appear occluded between surfaces in the line of sight of the viewer. This method is particularly effective in the NPR rendering of 3-dimensional characters as line drawings. The method presented by Sousa also addresses the NPR line drawing of 3-dimensional characters. Contours based on occlusions and silhouettes are extracted and user interaction adjusts thresholds to establish the numbers and types of contours that are aesthetically appropriate for conveying the shape of the subject. Line weighting may find application in the present research.

Other artifacts we might expect to see are depth cues and asymmetry. In this case a cord is drawn narrower when further away and wider when it is closer. Seen more often in Celtic design is the effect of natural cords being soft. When pressure is applied to a cord, it deforms by becoming flatter. In other words, being squashed. When this happens the cord presents more surface area and is perceived as becoming wider. Also, when we look at cord we will notice that the right-hand side does not have exactly the same shape as the left hand side. Because each was drawn independently, each has its own unique variations, most notably in terms of vector noise. It should also be noted that some artists can draw perfect curves. So, while we may be able to measure the variations in a drawn line, we may find that it contains no noise.

There is also the matter of choosing the lines to draw. Gail Hannah (2002) presents studies involving lines in space. In the preparation of this study she has identified a series of eleven lines that evoke specific emotional responses. They are grouped into three categories: slow curves, fast curves, and directional curves. Each stroke is chosen for its specific shape. In studying the lines of Celtic interlace some of these lines look familiar. Incorporation of these shapes may also be important as suggested by a study by William Freeman (2003) on the effect of line shapes on the perception of emotion or style. Next we turn to drawing instruments and media.

2.14 Instruments and media

At this point we know that the insular manuscripts were rendered on calfskin vellum with egg tempera paints and iron gall inks. This section summarizes a survey of research available for simulating media and identifies those based on complex systems methods.

This is a topic that belongs to the area of non-photorealistic rendering methods. Traditional drawing and rendering instruments include the graphite pencil, pen and ink, brush and ink, and brush and paint. Contemporary surfaces are paper and canvas. Physically based models were located for the following combinations of instrument, pigment/vehicle, and surface:

- Brush and watercolor on paper (Small 1991)
- Graphite pencil on paper (Sousa 1999, 2000)
- Brush and ink on paper (Zhang 1999, Chu 2005, Huang 2001,2003)
- Pen and ink on paper – non-of the methods found in this combination were physically based.
- Brush and paint on canvas (Baxter, 2004, Drago 2004)
- Brush and watercolor on canvas (Laerhoven 2004, 2005)

No modeling of animal skin vellum was located. It is believed that this is an open question. Raman spectra of parchment and vellum have been recorded for characterization and authentication of materials (Edwards, 2001). These would be suitable in part for developing color models based only on spectral reflectance. A follow on study (Edwards,

2004) provides results on the effects of biological and chemical deterioration on spectral properties. As mentioned in the introduction, Raman and hyper-spectral analysis has been applied to identifying inks and pigments in medieval manuscripts (Burgio, et. al, 1997, Melessanaki, et. al, 2001, Gardner, et. al., 2003, Brown and Clark 2004a, 2004b). Again these would be suitable in part for developing color models based on spectral reflectance.

The model presented by Mario Sousa and John Buchanan (1999, 2000) for graphite pencil and paper is the most complete. They divide the problem into three parts:

- Low level (drawing materials) – a low-level simulation of wood-encased graphite pencils and drawing paper
- Medium level (rendering methods) – this addresses drawing primitives for pencil stroke and mark making built on top of the low-level simulation
- High level (drawing composition) – partial control of composition through the ordering and scheduling of drawing steps

The low level model simulates the following properties:

- Pencil hardness – this addresses the rate of flow of graphite from the pencil core to the pencil point
- Pencil points – this addresses the way in which a pencil has been sharpened and models its contact with the surface of the paper
- Papers – this addresses how the surface of a paper accumulates lead
- Pencil and paper interaction – this addresses the deposition of lead on the paper surface.

There are a large number parameters involved in the model. Rather than listing the equations at this point I will attempt to describe what is being modeled and the manner in which this is being done.

Pencil point—three types are defined (typical, broad, chisel)

Tip shape—polygon outlines are defined for each point type. One of these is chosen and scaled for the thickness of the lead.

Pressure distribution--this describes the pressure applied across the surface of the tip polygon. This represents a distribution scaled according to the amount of the tip that is in contact with the paper. This corresponds to the shape of the sharpened tip as it is pressed against the surface of the paper. If the pencil is very sharp, the distribution will be concentrated at the location of the point within the tip polygon.

Paper grain – a cellular model is set up for the paper with one cell per pixel over a surface area up to 1280 x 1024. The grain at each cell is modeled 3-dimensionally as a clipped voxel, where the clipping plane is defined by four height values. This creates a simple shape that looks like a chisel.

Pencil and paper interaction – pencil strokes are created by friction modeled between the lead and paper as the pencil is drawn across the surface. For each position of the pencil tip as it passes over the surface of the paper, the following calculations are performed:

1. Evaluation of the shape of the polygon tip
2. Setting a local lead threshold volume for the paper
3. Calculation of the pressure applied by the pencil to cells covered
 - a. Calculate the porous threshold
 - b. Calculate the grain biting the lead

- c. Calculate the damage caused by the lead to the cell
- 4. Calculate the reflected intensity of the lead, meaning the color of each cell as a function of the amount of lead deposited

To summarize this section, we were unable to find a physics based model for animals skin vellum; and this modeling appears to remain an open question. We were able to find a physics base cellular model for a pencil.

2.15 Art criticism

The mainstream view is that art is an intimate personal expression shaped directly by an artist's hand in some physical medium. An example of this is on display in my orthopaedic's waiting room. A three dimensional doorway to an adobe hut and a pot are both rendered in relief in white paper. The process most likely involved making the paper over a relief fabricated in another medium. From a distance, though, there was ambiguity in what I was seeing. I was unable to determine if I was viewing a sculpture or a rendering until I made close inspection and was able to determine that the shadows were physical. This leads to a further dilemma in that ambiguity is considered a bad thing. So, while these pieces were quite beautiful, they were breaking a fundamental rule ... in that I was unable to readily determine if they were sculptures or paintings. Even in writing, to be art, we might require a text be written in the author's own hand, as with one fellow who spends endless hours composing elegant poetry on handmade parchment at Starbucks or the latest fantasy by J. K. Rowling scribbled on lined pad with pencil.

With this in mind, it was with tremendous revulsion that George Stiny's paper on algorithm aesthetics (1975) was met ... *the mechanistic tenor and extreme reduction of art to*

mathematical principles was objectionable to the mainstream art community (Taylor, 2004). A central issue remains a perception that when a computer is involved in the creation of something there is a sense of depersonalization, *human-vs.-machine*. Work in the computer sciences on the question came to an almost immediate halt with only a handful of papers braving the waters since that event.

Today one of the roles of the technical director in feature film is to set up the digital paint environments for artists so that they can take the pen and place the pixels where and how they need to go on all those surfaces. The pen is not yet a brush. It is as close as we can get the artist to being in direct contact with the digital medium. There is still software where software needs to be in the process to help get artists go where they need to go and produce the results they need to produce. What is understood is that a trained artist possesses a different way of looking at the world than others, whether this be solely a byproduct of their training or an artifact of genetic predisposition.

Still there remains widespread controversy in the arts regarding the critical disposition of products of artistic processes that involve computers. This places the idea of improving the quality of line drawing in a difficult light. Why should we? Why don't we just leave this to the artist and be finished? So, Andrew Glassner's Knot Assistant provides a means of planning something rather than completing it because it is easier to finish with pen and ink than with the mouse or any render engine.

In the case of this research, the same rule applies. If it is relevant to what someone needs to do, that artist can do whatever they feel is most appropriate in applying what is learned.

2.16 Summary

In this chapter we surveyed the interdisciplinary research on Celtic design. We reviewed the history of manuscript production to understand what was technically and culturally involved. We examined the archeological findings to help define the boundaries in the designs of forms and structures. We reviewed the five modern methods of construction and the work that has been done with computer implementations. We have worked on elements of creative design theory. We have gathered what is known of the mathematics of Celtic design. We reviewed complex system theory and many of its methods. We have introduced differential geometry to help us understand the line. We have taken a look at some properties of noise that may be able to help. We have established essential characteristics that we expect to see in lines that are drawn by hand. We have searched what is currently possible with the modeling and simulation of the physical media. Finally, we have struggled with the questions of the computer in the arts. In the next chapter we will summarize the derivations and experimentation performed in determining feasible directions for the present study

Chapter 3 Adaptation

Adaptation of Related Research

3.1 Introduction

In this chapter we develop the adaptations of related research needed to find feasible solutions for the questions posed in this study. The reader will recall that we posed the following questions towards the end of Chapter one:

- How do we compose?
- How do we accomplish spatial organization?
- How do we decide what fills the spaces/frames?
 - How do we choose unit forms?
 - How do we decide which interlace to use?
- How do we control crossings?
- How do we draw lines?
- How do we draw cords and anatomy?
- How do we provide the means for a designer to be expressive?
- How do we facilitate the creative process?

Our review of issues in art criticism leads us to understand that significant aspects of questions one and three rest exclusively with the artist/designer. While we will begin the preparation of the foundation needed to support the automated exploration of questions such as these, we will first focus our analysis on capabilities needed to study the problem of improving the quality of mark making. Our study will examine the following issues:

- The modeling of lines
- Structure or the problem of grids
- Posing and spontaneity

Although these are the pivotal questions, they will help us resolve which programming language to use, the definition and analysis of the drawing space, and the solving of interlace.

3.2 The modeling of lines

The method of line drawing we saw in Michael Wong's work (1998) was based on *skeletal-strokes*, an extension reported by Siu Chi Hsu, et. al., (Hsu, 1993) of a method, developed earlier by Richard Beach and Maureen Stone (Beach, 1983), that involves warping a texture to follow the path of a spline. This method provided the first and simplest means of emulating a brush stroke and relies on an image of the brush stroke for its realism.

In the present research we seek a model for lines whose evolution is closely matched to the way lines are drawn by the human hand. As in the study by George Cellick (1991), a common choice of deformable curve primitive in evolutionary design is the Hermite cubic spline. Its appeal comes from the simplicity and control afforded in its specification, by means of two points and the tangents at those points. If we describe a curve in terms of a

sequence of Hermite cubic splines we find several disadvantages over the space curves having assigned curvature from differential geometry. First of all we need to specify the positions of all the control points and the tangents at those positions for each segment of the curve. This creates a problem if we don't know this. Delving further we find that the speed of the curve is not constant, which would develop if the physical motion of the hand is found to be an integral part of the appearance of the lines it draws. If we attempt intrinsic specification, in terms of arc length and curvature, the derivation of these in Mathematica reveals unmanageably complex expressions for the parametric form.

At this point it was reasonable to ask if it was possible to work directly with space curves of assigned curvature. Gray (1998) structured his course around the use of Mathematica to help the student quickly grasp an understanding of the depth of possibilities in the representations of differential geometry. We also find curves such as the prolate cycloid that resemble the curling snakes in folio 34r of the Book of Kells. We also notice (Gray, 1998, p. 190-191) that curves of arbitrary speed can be reparameterized as unit speed curves. The fundamental theorem of space curves together with the fact that both positive and negative curvatures can be assigned to curves in the plane, allow us to conclude that this approach to modeling lines provides us with a capable foundation.

The significant way in which unit speed curves of assigned curvature differ from splines is that we no longer have the luxury, convenience, and safety provided by interpolating polynomials. Interpolating polynomials provide the means by which splines are able to infer the shape of the curves they model between explicit control points. With an implicit model for space curves with assigned curvature, we need to specify the curvature at every point along the arc length of the curve. The advantage is that we can do anything needed. The disadvantage is that we must describe it

In Chapter two we introduced a model for clothoid splines used in civil engineering to design curves for roads and highways (Dickmanns, 1992). Formal development of problems involving interpolating clothoids has been carried out by Meek (1991, 1992). What Meek shows us that designing with clothoids involves a model-based approach. In other words models of curves will take the place of interpolating polynomials. The first paper in this series presents a segmented approach to modeling space curves as a function of arc length. An exit ramp that merges with an crossover is shown modeled in terms of lines, circles, and interpolating clothoid splines along the length of the roadway, and this affirms, [after the fact] our intuition that segmented representations are not unreasonable.

One of the most immediate needs we have is the practical matter how we are going solve these curves. Autodesk formally Alias, Maya, does not offer access to its physics engine. Further examination reveals that data abstractions in the Maya Embedded Language (MEL) at this juncture have apparently been considered unimportant, which is a significant disadvantage when considering application development. We also have advice from Kurt Fliescher's thesis (1995) to use a development environment that includes common things such as graphing tools. In the present case almost half of the student's masters thesis, written in C, was devoted to language specification and interpretive evaluation. This and the present hypothesis suggests that a first class language may provide advantage. Using examples provided by Gray (1998) the solution of the Frenet frame was timed in Mathematica and then developed and timed in Matlab, the two tools available to the student that provide built-in support for solving differential equations. It was found that Matlab provided the fastest solution for the differential equations of the Frenet frame. An additional advantage is that the Matlab programming language, m-code, provides a syntax familiar from mainstream languages, a flexible system for object programming, and a powerful dynamic schema system for organizing information. Given that both Matlab and

Mathematica provide low-level access by means of Java, choosing the m-code language of Matlab over Mathematica would not preclude the participation of Mathematica, should it be found to be the most appropriate solver for some problems in this study. The language system from the student's masters thesis, earlier ported to Perl, was ported from there and optimized for Matlab to be available as needed.

3.3 Curvature dynamics

Save for quantum mechanics, nothing macroscopic involving kinematics in nature is instantaneous, even in the most abrupt movement. In order to more closely emulate physical behavior, we would introduce a model for space curves with assigned curvature that results in the ability to smoothly change that curvature down to acceleration. We begin by thinking about how a reaction control system works in space. We fire a small motor (chemical or ion) in one direction to generate an impulse that pushes us in the opposite direction--hopefully, the one in which we wish to be going. To check that motion, we need to generate another impulse to cancel the first.

Notice that we are not interested in producing a specific curve in an analytic sense. In other words, we will be satisfied with whatever curve is generated by our process and be willing to adjust it by means of the same process as needed. This removes the problem of worrying about things such as the elliptic integrals involved with the generation of transcendentals (e.g., a sine wave).

The first thing we need to do is model the shape of an impulse function. We observe something that looks familiar as a piecewise symmetric assembly of two Hermite cubic splines. We can adjust the underlying area to interpolate the accelerations needed to produce specified curvature velocities. This model gives us smooth differentiable transitions

to and from zero. The half impulse is modeled in the unit square of the first quadrant with the following parameters:

$$P1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad P4 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \quad R1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad R4 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

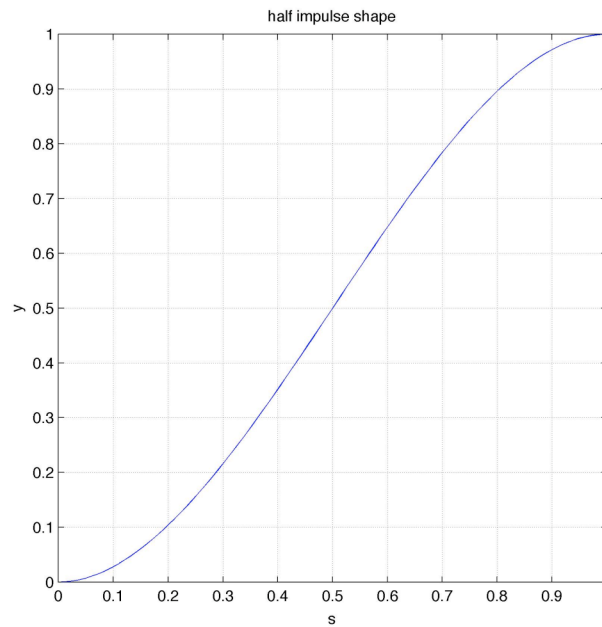


Figure 33: Hermite cubic spline as half impulse

Given the arc length interval $[s_0, s_1]$ over which an impulse is to be applied and the height of the curve $[y_1, y_4]$ that corresponds to half the desired area, the corresponding analytic curve is found as

$$f(s) = y_1 - \frac{(s - s_0)^2 (2s + s_0 - 3s_1)(y_1 - y_4)}{(s_0 - s_1)^3}.$$

For a specified change in curvature, one must determine the changes needed in curvature velocity and then the changes in curvature acceleration that induce the velocities. We assume that the change in curvature will have the shape of a scaled easement function over an interval of arc length. This change must equal the area under a velocity curve. We assume that the velocity curve takes the shape of a symmetrical impulse. If we think of the Hermite curve shape as an easement, we will want to use an ease-in for the head of the impulse and an ease-out for its tail. We then form the head of the velocity impulse:

$$v(s)_{head} = f(s), \text{ where } [y_1, y_4] = [0, a_{vel}] \text{ and } [s_0, s_1]_{velocity} = \left[s_0, \frac{s_1}{2} \right]_{curvature}.$$

We solve the following curvature velocity integral for the amplitude, a_{vel} :

$$\kappa_{change} = 2 \int_{s_0, velocity}^{s_1, velocity} v(s)_{head} ds.$$

We next observe that the velocity impulse was induced by two acceleration impulses. The first acceleration impulse begins the movement and a second one stops it. The maximum velocity corresponds to the area under an acceleration impulse. We again assume that the acceleration is a symmetrical impulse. One acceleration impulse is assigned to the first half of the velocity impulse and another acceleration impulse of opposite amplitude is assigned to its second half. We repeat the previous calculations over these intervals to determine the amplitudes of the acceleration impulses:

$$a(s)_{head} = f(s), \text{ where } [y_1, y_4] = [0, a_{acc}] \text{ and } [s_0, s_1]_{acceleration} = \left[s_0, \frac{s_1}{2} \right]_{velocity}$$

We solve the corresponding curvature acceleration integrals for amplitudes:

$$v_{change} = 2 \int_{s_{0,acceleration}}^{s_{1,acceleration}} a(s)_{head} ds$$

We finish by scheduling a motion event for each acceleration. We derive the velocity and curvature functions on each acceleration interval by integrating the acceleration function. We solve the boundary conditions by evaluating the acceleration, velocity, and curvature functions at the beginning and end of each interval.

We implement the mathematics analytically in Matlab and use motion events to schedule equations in corresponding arc length intervals by means of an interval table. A family of curvature impulses is shown as follows:

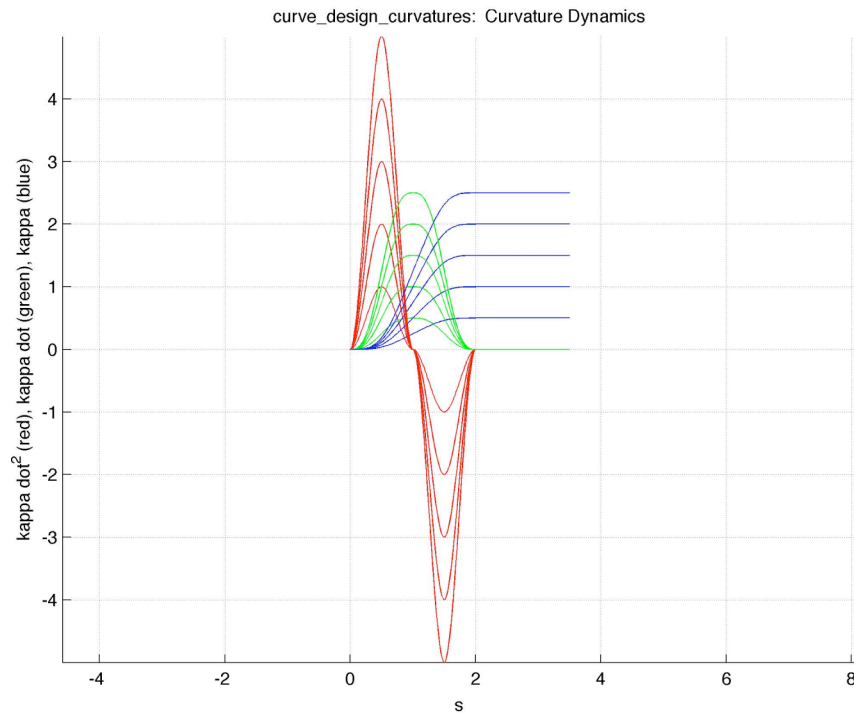


Figure 34: curvature impulses and their decomposition

Solving the differential equations of the corresponding Frenet frames we find the following family of curves:

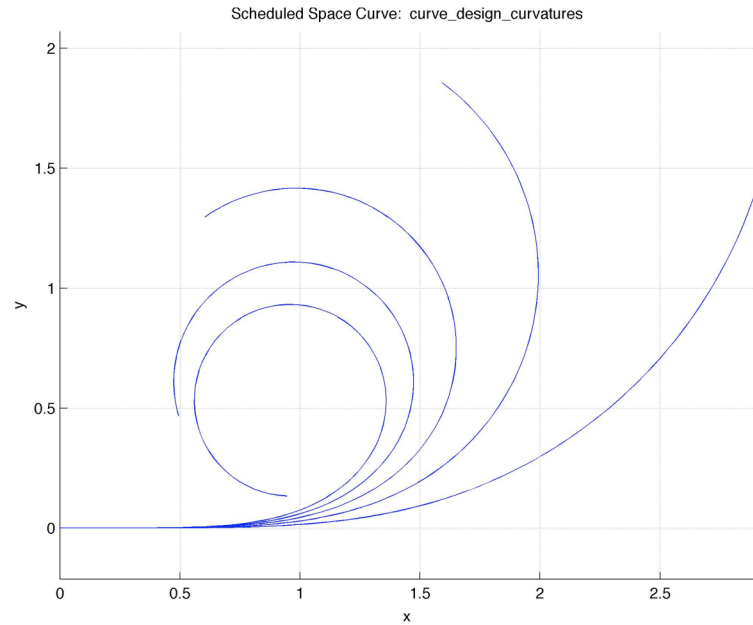


Figure 35: schedule space curves with assigned curvature

As another example, we apply curvature rate impulses:

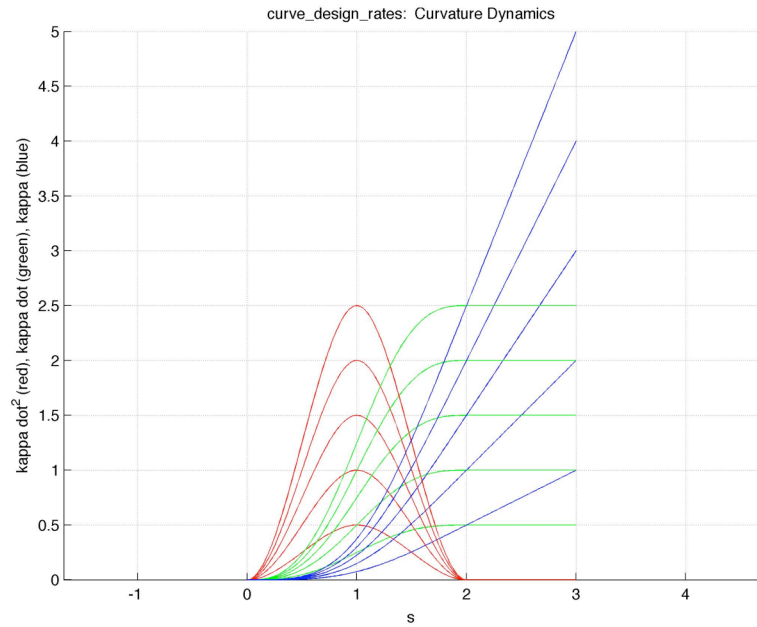


Figure 36: curvature rate impulses

Solution of the corresponding Frenet frames give us the following curves:

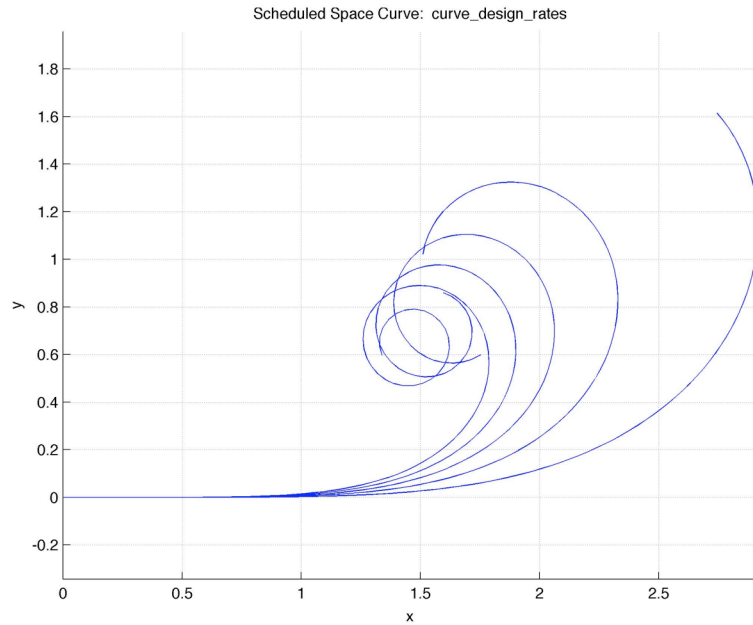


Figure 37: Space curves with scheduled curvature rates

Last we repeat the exercise with assigned curvature accelerations:

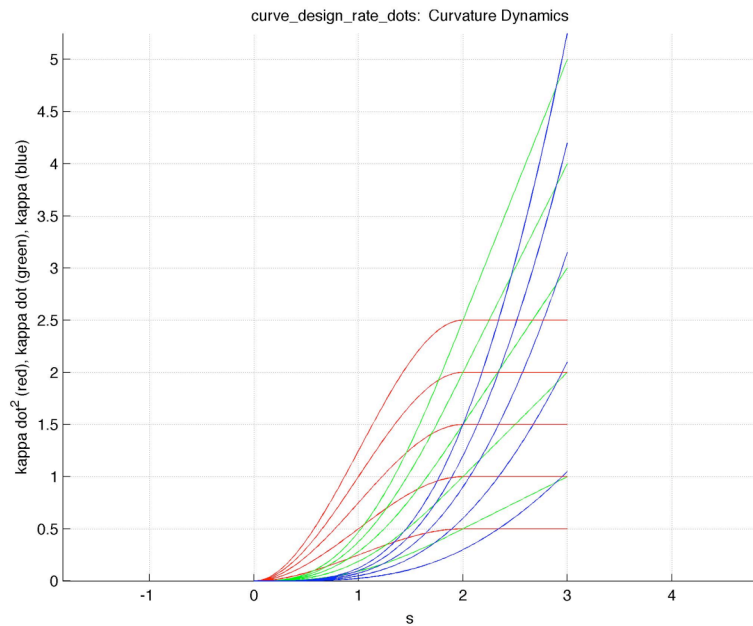


Figure 38: curvature acceleration impulses

From which we obtain a family of corresponding curves:

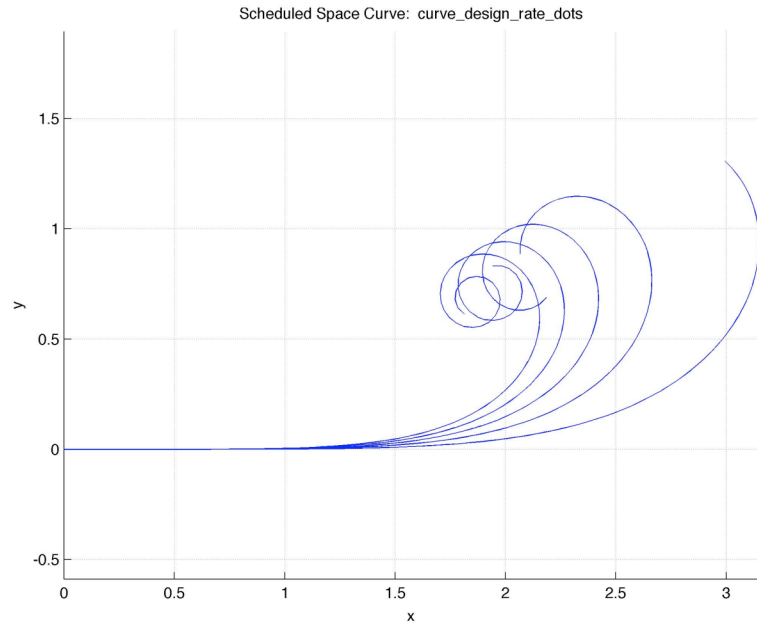


Figure 39: Space curves with scheduled curvature accelerations

We find, as we expect from intuition, constant curvature goals give us curves that evolve into circles. Velocity goals give us more aggressive curves, where the curvature changes at a constant rate. And finally acceleration goals provide the most extreme changes in curve shape, where the rate of change of curvature is changing. Note that this method is able to accommodate the linear curvature velocity changes needed to generate clothoid splines.

3.4 Linear genetic programming and evolvable curves

In order to have evolvable curves we need a bit more than being able to schedule equations on intervals. We need a modular means of denoting their specifications, in other words, a way of stating initial conditions and then defining curvature as a function of arc length. For example, we wish to be able to describe the following curve

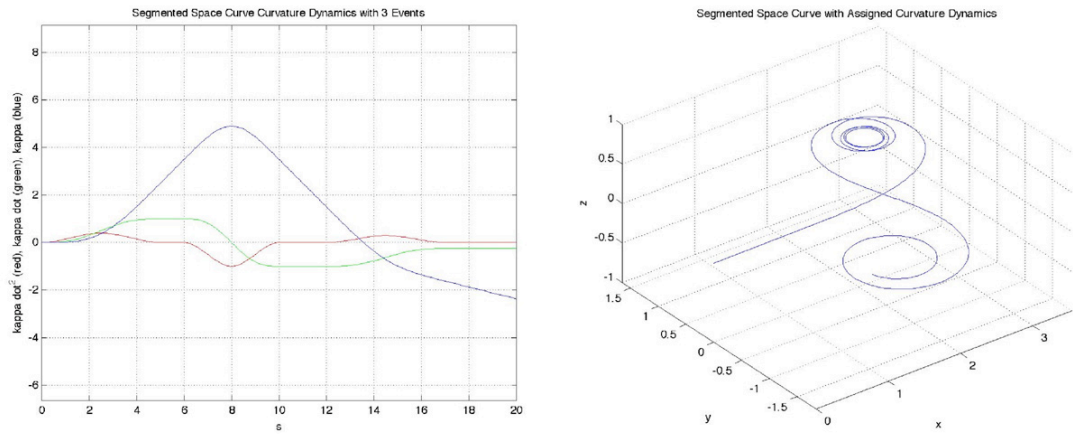


Figure 40: Segmented planar space curve with assigned curvature dynamics

in a compact way such as:

```
{ 'new' 'intrinsicCurve' 'myCurve' }
{ 'impulse' 'myCurve' 5 'parm_rate' 1 }
{ 'extend' 'myCurve' 1 }
{ 'impulse' 'myCurve' 4 'parm_rate' -2 }
{ 'extend' 'myCurve' 2 }
{ 'impulse' 'myCurve' 5 'parm_rate' 0.75 }
{ 'extend' 'myCurve' 3 }
```

This representation is a form of linear genetic programming, where a nucleotide of the sequence consists of a single imperative statement. This is developed further by adding expressions in prefix notation, a conditional, and relative branching. Operators that facilitate sequence marking and rewriting, as well, as the assembly and management of forms have been added. Variables and scoping allow the behaviors of sequences to respond both to the individuals to which they belong, as well, as to the environmental factors presented by a composition and user interaction. The translation behavior of the ribosome becomes the interpreter. As with its biological counterpart, no restriction is placed on the length of a sequence, with the intent of investigating the issue of essential grouping of nucleotides, a question that remains open. The complete instruction set is given in Chapter 5.

A recent study of theory and practice with this representation is presented in the Ph.D. thesis of Markus Brameier (2004). Although the focus of his research was program

evolution and basic structural issues, his results demonstrated some advantages of linear representations over tree-based ones. These included the compactness of solutions under unrestricted crossover, the diversity of effective programs (structural difference distances), suitability to be varied in smaller steps, efficiency in register utilization, and reduced code growth with a specific intervention that blocks neutral variations. In practical terms the student found the representation easy to work with.

3.5 Structure of the problem of grids

The structure component in Celtic design is the grid. It is assumed to be formal and invisible. In this section we will discuss the challenges and experiments associated with the design and installation of a grid.

Analyzing the drawing area

It will be assumed that the design is represented as a digital image and is arbitrarily large. Due to practical limitations in the size of memory it will also be assumed that the designer divides the work into smaller parts. These parts will be referred to as regions. Associated with each of them will be a matte. It will be assumed that the matte is anti-aliased. If there is an under drawing that the designer would use for reference, it is assumed that the matte would cover the area of the design corresponding to the under drawing and would be the same matte used for its extraction. It is assumed that the designer works with a single one of these regions at a time. For the purpose of this discussion it is assumed that the designer has selected one of these regions. It will be referred to as the active region, we will assume that it is blank, and that blank corresponds to white. One final thing is that we will

assume that the shape of the active region is the same as the lozenge in folio 34r of the Book of Kells

Prominent features of this region are its borders and corners. At this point these are represented only by changes in pixel color. We need to associate them with the vertices and edges of the bounding polygons, inner and outer, since the active region contains a hole. Examining the remainder of the folio, it is clearly a coincidence that the region is bounded by straight lines. As a consequence, it is important to note that the following analysis applies primarily to regions bounded by simple polygons. If the region contains holes, they must also be bounded by simple polygons. We also rely on the boundary of the active region to be anti-aliased; otherwise, due to their sensitivity, the extraction procedures will fail.

We calculate the gradient magnitude and angle (Gonzalez, 1992)

$$|\nabla f(x,y)| = \sqrt{\left(\frac{\partial f(x,y)}{\partial x}\right)^2 + \left(\frac{\partial f(x,y)}{\partial y}\right)^2}$$

$$\alpha(x,y) = \tan^{-1}\left(\frac{\frac{\partial f(x,y)}{\partial y}}{\frac{\partial f(x,y)}{\partial x}}\right)$$

of the active region and extract the boundary using a boundary tracking turtle/bug (Castleman, 1996). Canny edge operators (1986) are used to calculate derivatives:

$$g_x = \frac{\partial}{\partial x} = \begin{bmatrix} -0.325532 & -0.536711 & -0.325532 \\ 0 & 0 & 0 \\ 0.325532 & 0.536711 & 0.325532 \end{bmatrix} \quad g_y = \frac{\partial}{\partial y} = \begin{bmatrix} -0.325532 & 0 & 0.325532 \\ -0.536711 & 0 & 0.536711 \\ -0.325532 & 0 & 0.325532 \end{bmatrix}$$

Calculation of these operators was adapted from an implementation by John Koo (1996).

When finding or being placed on a pixel having a non-zero value in the gradient magnitude image, the turtle walks in the direction of the local maximum to find a peak and uses this as an initial point for boundary tracking. Moore connectivity is assumed. The track of the turtle is recorded as a chain code sequence.

Because array indexing in Matlab begins with 1, chain code directions on the Moore neighborhood in this application are labeled as follows:

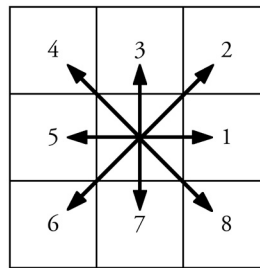


Figure 41: Chain code directions

The stationary move is labeled 9.

Corners are sensed using the detector by Chris Harris (1988):

$$M = \begin{bmatrix} \frac{\partial^2 I}{\partial x^2} & \left(\frac{\partial I}{\partial xy} \right) \\ \left(\frac{\partial I}{\partial yx} \right) & \frac{\partial^2 I}{\partial y^2} \end{bmatrix}$$

$$R = \det M - k(\text{trace } M)^2$$

$$k = 0.4$$

where I is the image.

Positive excursions of the filter above the noise floor are retained and a cellular calculation of the local maximum directions is simulated. This creates a image in which pixels are labeled with the chain code direction of the maximum value in the neighborhood

of the corresponding pixel in the detector image. If the background is set to zero, detector peaks may be read off as stationary moves in the label image. This removes the problem of having to decide how hot the corners are and thresholding the detector.

In regards to the noise floor, noise is present due to quantization error in the anti-aliasing of the region boundary. Part of this is dependent on the orientation of its edges. The Canny edge detector contains a Gaussian filter component that helps reduce this noise. For the purposes of this study, setting a noise threshold of 0.25 of the value of the peak detector signal was adequate to prevent false alarms.

The corner locations and the extracted boundary were found to be in agreement to within one or two pixels. A cellular algorithm using empirically derived heuristics on patterns in radius 1 and 2 neighborhoods of the corner pixels relative to the extracted boundary was simulated to relocate them to the boundary.

With this completed, the turtle was used to take one more traversal of the extracted boundaries to order the corners and assemble oriented boundary polygons. These were saved as a geometric complex consisting of vertices, edges, faces, and their relationships.

A cellular simulation of Kesheng Wu's (2005) algorithm for connected component labeling was used to label the connected components of the active region. This result was used to classify each part as either form, ground, or void. Connected regions labeling was replaced in the demonstration application by the Matlab function, `bwboundaries`. At this point we have finished all of the preliminary work leading up to the design of structure.

Automatic grids

The objective is to install a conformal quadrilateral mesh in the active region. A series of experiments were conducted to determine how one might fit an automatic grid to an arbitrary canvas:

- Automated mesh fitting by point projection evolution Browne (2005)
- An evolution based on contours of the Euclidean distance function.
- A self-organizing grid based on Kohonen (2001) networks
- A self-organizing grid based on social agency, Medina (2005)
- Semi-automated grid placement based on templates

Results of these experiments are summarized in the following sections.

Automated mesh fitting by point projection evolution

After locating the boundary of the active region, the edge terminating the upper right wing of the active region was selected and marked for subdivision into four parts. The remaining edges were subdivided using this measure. Following the procedure outlined by Browne, normals were calculated with respect to these edges and the trajectories along which points would evolve were drawn in order to gain some intuition as to how this would proceed. The trajectories are marked as yellow lines in the following figure:

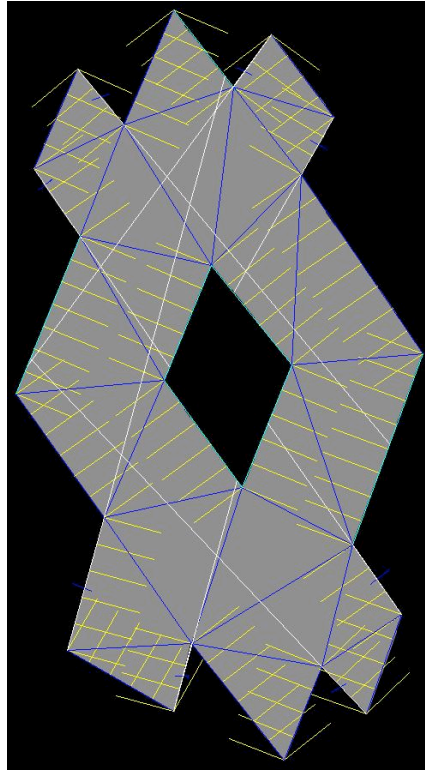


Figure 42: automatic mesh fitting trajectories

Also marked in this figure are the following features:

- Blue--the Delaunay triangulation from the corners
- White--Projections of edges from the wings

From these features we draw several conclusions:

- Because the geometry is skewed, the heuristic for the evolution of grid points applied by Browne must be adjusted, otherwise, evolutions from interior and exterior edges of the lozenge will be misaligned with the main symmetries.

- Projections of wing edges are not aligned with edges of the lozenge.
This means that evolution of grid points from the terminal edges of the wings into the main body of the lozenge will be misaligned.
- The Delaunay triangulation reveals the shape of the outer lozenge to be irregular with respect to the orientations and dimensions of the wings.
- The wings themselves are not identical

Mesh fitting based on contours of the Euclidean distance function

The Euclidean distance function with contours based on the subdivision of the preceding section yields the following result:

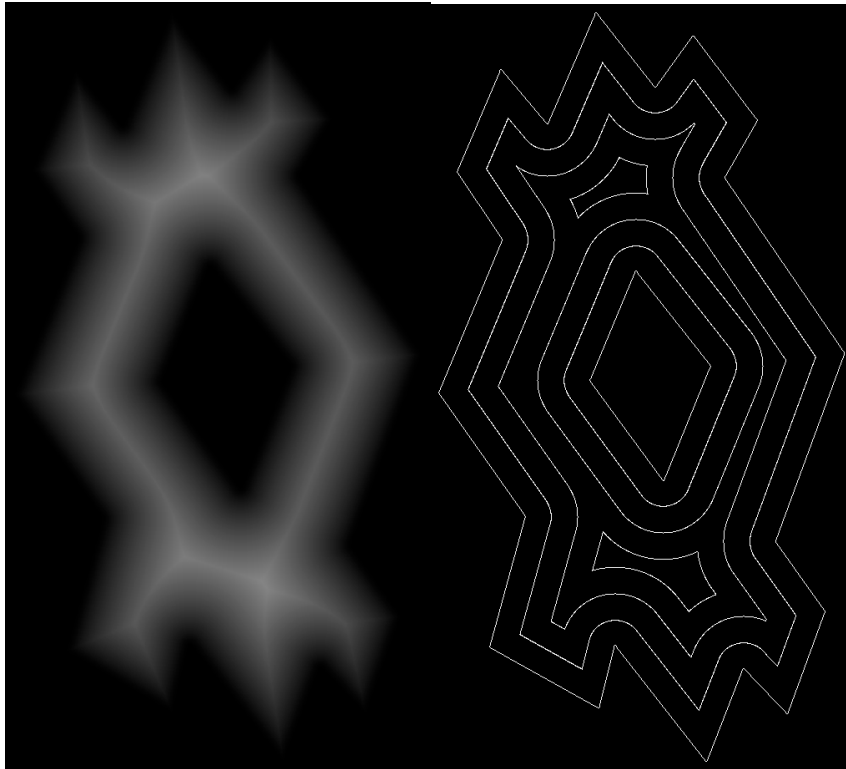


Figure 43: Euclidean distance function (left) contours (right)

As with the previous result, this shows us that three of the wings are of approximately the same width with one at the lower left being greater. This also shows that the channels in the active region are not of uniform width. Although alignment with the main symmetries use is good, the contours must still be interpreted to accommodate variations in width as well as the rounding that appears around the center void.

A self-organizing grid based on Kohonen (2001) networks

We begin with the simplest experiment of distributing weights of the Kohonen network over the surface area of the active region. The Kohonen is known to prefer approximate conformal mapping (Liou, 2000). We train the network by selecting points at random from the interior of the active region. We choose the dimension of the network to approximate the subdivision from the first section; and we obtain the following result:

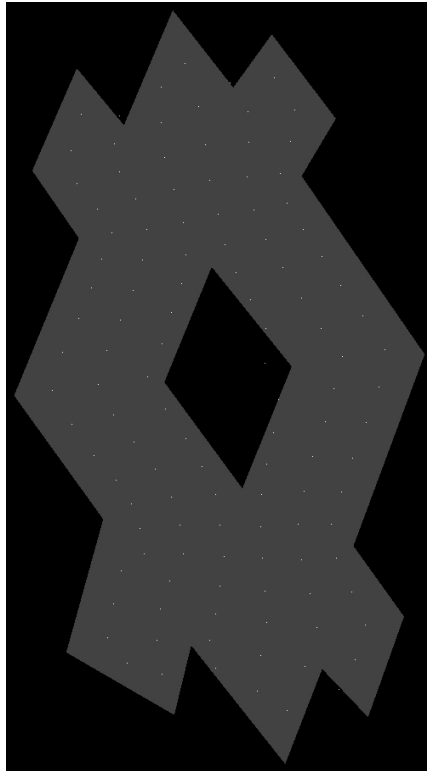


Figure 44: Kohonen grid weight distribution

We observe this to provide the best distribution of potential subdivisions within the channels of the active region. The uneven distribution of points in the wings are an artifact of the topology of the neural network itself (Liou, 2000). The excursion into the void at the center was somewhat unexpected. To obtain a very nearly (quasi-) conformal mapping of this region requires the derivation of a new energy function for the network to guide the evolution. This includes a measure of deformation and another of non-conformality (Liou, 2000). The method involves embedding an exotic bijection in the learning rule that performs a topological warping of the neural network so that it conforms to the topology of the region being mapped. Results of the method on test cases in the literature appear to be excellent. A clearer explanation or rederivation of the method will be needed to reproduce the result.

A self-organizing grid based on social agency

This approach would involve a cellular simulation of space filling in which groupings of pixels form organizations that differentiate themselves based on various objectives (Medina, 2005). These could include such things as competition, cooperation, size, shape, and proximity. In the present case we would be interested in seeing if some simple combination of objectives are sufficient to cause the evolution of a grid. This approach was held after reading the thesis in the next section.

Semi-automated grid placement based on templates

The Ph.D. thesis of Matthias Muller-Hannemann (1997) reports that the automatic design of conformal quadrilateral meshes with a shape objective is NP hard, based a mapping

to the 3-SAT problem. Therefore, the preferred approach is to use approximate methods based on templates. This can be done in cooperation with the artist as a drag and drop process, more automatically by applying a boundary flow heuristic that incrementally fits templates whose topologies most closely match subdivisions of a preliminary partitioning of a region, or algorithmically through the use of an integer programming method.

3.6 Posing and spontaneity

Among the thoughts at the beginning this research was that the simulation of natural phenomena might offer original patterns suitable for use as a basis for posing. As this thinking progressed the thought occurred to try and look for patterns in artificial noise. Perlin noise is a now famous procedural method for producing pseudo random noise patterns. These have been composed in various ways to simulate the look of a variety of natural phenomena.

Paulus Gerdes (1990) describes a common construction principal for Tchokwe and Tamil pictograms as follows:

Each of them is the smooth version of the polygonal path described by a light ray emitted from [a] point ... the ray is reflected in the sides of the circumscribed rectangle of a (basic) point reference frame. It encounters ... double-sided mirrors which are placed ... horizontally ... and vertically.

Examination of figures drawn in the paper show that these satisfy Thurston's theorem for the existence of interlace in knots and links. In other words, this common construction principal applies to Celtic design. Gerdes (1999) later applies the method to Celtic interlace for the purpose of generating Lunda designs.

The next section describes experiments with Perlin noise. That will be followed with a presentation of mirror curves as a lattice gas automata and then a presentation of

Christian Mercat's labeling algorithm as a lattice gas automata. Note that while the word automata is plural, its use is singular in parts of the literature that deal with lattice gas machines.

3.6.1 Perlin noise

The original technique (Perlin, 1985) involves the use of filtered noise vectors; and it was unclear as to how one might place the user in control the results. A second paper (Perlin, 2002) was published to address artifacts related to the continuity of a lofting function used in the original algorithm. This paper contained a comment that the filtered noise vectors could be replaced by a random distribution of vectors selected from a canonical set. What this suggested to the student was that the improved version of the Perlin noise field can be genetically specified as a sequence of labels, each of which denotes a vector from the canonical set.

Implementing Perlin's revised algorithm based on an explanation of first principles by Matt Zucker (2001), the following result was obtained from a 256 x 256 image with a scale factor setting the internal grid size to 32-pixels:

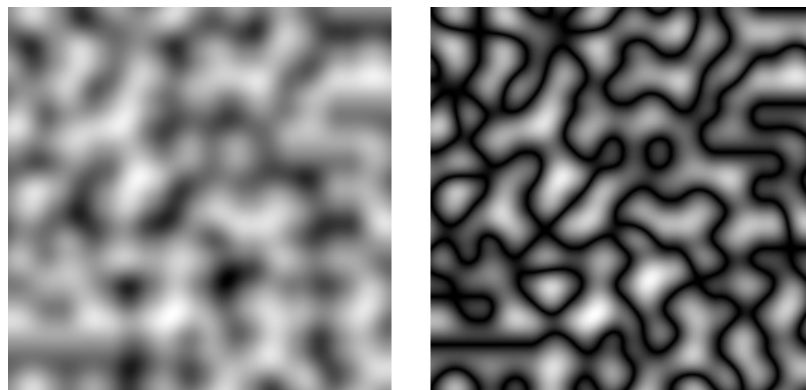


Figure 45: A Perlin noise image (left) and its absolute value (right)

The absolute value of the noise field was calculated as layering of this result is used to generate the texture of clouds. The filaments in the absolute value of the noise image were an immediate attention grabber. The question becomes whether or not one can provide enough direction to evolve the appearance of interlace.

Before we go too much further we need to explain the noise field generator. This explanation will closely follow that of Zucker, although my interpretation of at least one of the calculations is a little different. The set of canonical vectors is as follows: $[1\ 1\ 0]$, $[-1\ 1\ 0]$, $[1\ -1\ 0]$, $[-1\ -1\ 0]$, $[1\ 0\ 1]$, $[-1\ 0\ 1]$, $[1\ 0\ -1]$, $[-1\ 0\ -1]$, $[0\ 1\ 1]$, $[0\ -1\ 1]$, $[0\ 1\ -1]$, and $[0\ -1\ -1]$. If it is advantageous to have 16 vectors in the set, the following pad is recommended: $[1\ 1\ 0]$, $[-1\ 1\ 0]$, $[0\ -1\ 1]$, and $[0\ -1\ -1]$. These vectors are normalized and treated used as the normals of planes passing through the corners of the cells of a grid interior to the image. A single cell in this grid might appear as follows (angles have not been projected with any care):

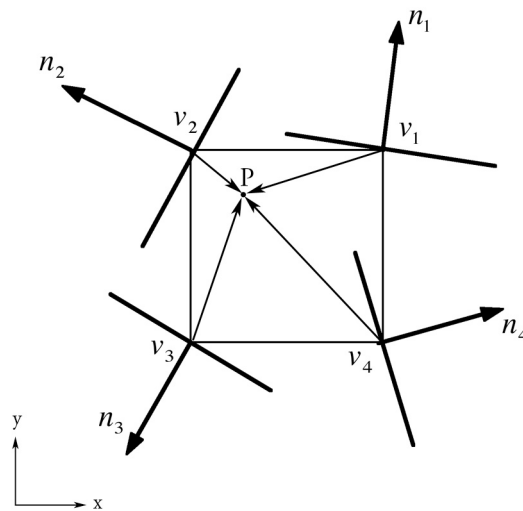


Figure 46: Perlin noise field grid cell

For each point P in the image, we calculate the vectors to it from the corners of the grid. We then take the dot product between these vectors and the plane normals calculating

the distance of the point from each of the planes. These distances are just scalars and we use them as measures of a random height field at each corner of the cell. We might visualize this as follows (heights not carefully projected):

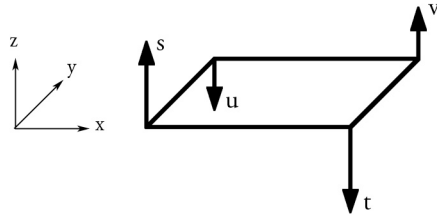


Figure 47: Perlin height field after dot products

We want to calculate a weighted average of these at P . We use a cubic smoothing function that adjusts the influence of each height according to its distance from P :

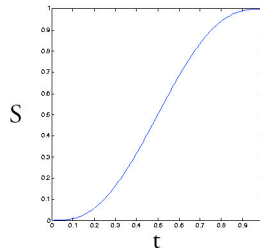


Figure 48: Perlin blending curve

$S(\mathbf{t}) = \mathbf{t}^3(6\mathbf{t}^2 - 15\mathbf{t} + 10)$ on the intervals $t_x, t_y \in [0,1]$ and $\mathbf{t} = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}$, where (x_0, y_0) are

barycentric coordinates of the point P in the cell. We calculate the averages as

$$\begin{aligned} a &= s + S_x(t - s) \\ b &= u + S_x(v - u) \\ z &= a + S_y(b - a) \end{aligned}$$

with z as the value of the Perlin field.

So, now we get to see what happens when we tune the normals. Using just a single orientation, $[1 \ 1 \ 0]$, we get the following result:

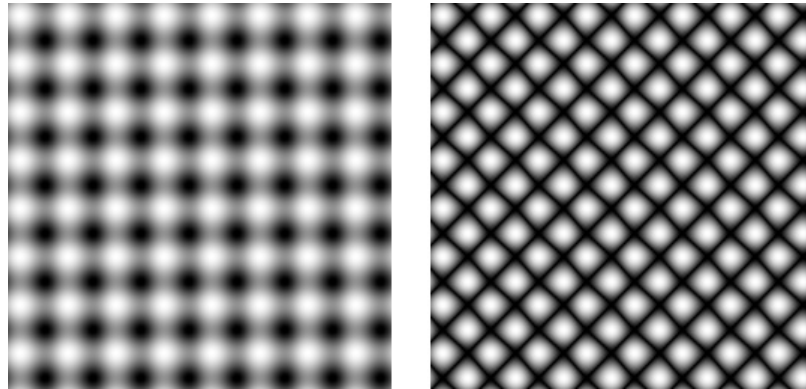


Figure 49: A Perlin noise image (left) and its absolute value (right)

We immediately recognize the plait pattern.

The next question that arises is whether or not we can find the elementary movements that were identified by Adcock. The hypothesis would be that we compose interlace using the elementary movements has the symbols of a grammar. We then need to find generators of the elementary movements. A conventional approach would be to use correlation with the scaled shape of an elementary movement as a fitness function for a genetic algorithm. Genotypes would be sequences of normals. Phenotypes would be the generated noise fields. We search for correlation sites within with the phenotype of each individual and record the orientations of the corresponding normals. After normal sets for each of the elementary movements have been found, one would attempt to construct interlace by spatial arrangement and sequencing of these groups.

An extraordinary example of patterns seen in the noise from early experiments with the genetic algorithm was the cross in circle in the following image:

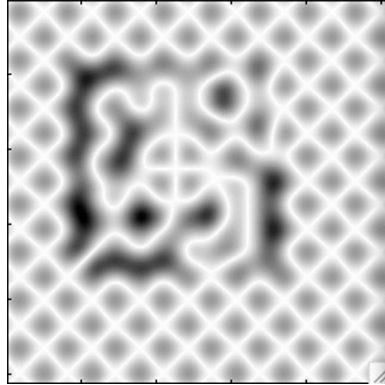


Figure 50: Cross in circle

The m-code version of the Perlin noise function runs relatively slow even after optimization. The function was translated to C for the genetic search experiment. The noise generator containing the C coded noise function ran about 40 times faster. The Matlab tools make translation relatively straightforward; however, a bug that caused Matlab to crash without successfully trapping was discovered during development. This was found to be peculiar to the Macintosh port of the product.

The use of a correlation filter is also referred to as template matching. Convolution with a correlation filter is limited in the sense that it is rotation dependent. Search based on this type of matching becomes impractical when the orientation and scale of a pattern may vary arbitrarily. To alleviate this problem we wish to change the approach to using wavelet based matching modified by a ring-projection transform to remove rotation dependence (Tsai, 2002).

The wavelet transform of a 2D image $f(x,y)$ is defined as its correlation with a family of wavelet functions $\{\varphi_{s,t}(x,y)\}$:

$$W_f(s,t;x,y) = f(x,y) * \varphi_{s,t}(x,y),$$

where wavelets are generated from a mother wavelet function:

$$\varphi_{s,t}(x,y) = \frac{1}{s} \varphi\left(\frac{x-t_x}{s}, \frac{y-t_y}{s}\right).$$

In most typical applications the transform is represented by a multi-resolution pyramid of transforms. The wavelets transform at each resolution is decomposed into four subimages: a smooth approximation subimage, a horizontal detail subimage $|f_H^{(J)}(x,y)|$, a vertical detail subimage $|f_V^{(J)}(x,y)|$, and a diagonal detail subimage $|f_D^{(J)}(x,y)|$, where J is the resolution level.

A wavelet basis with shorter support, such as the Haar wavelet, was reported to be the best choice for this type of application. For both the template and the search image, a composite detail subimage, $f_d^{(J)}(x,y)$, is computed from the horizontal, vertical, and diagonal edge information of the wavelet decomposition at appropriate resolutions:

$$f_d^{(J)}(x,y) = |f_H^{(J)}(x,y)| + |f_V^{(J)}(x,y)| + |f_D^{(J)}(x,y)|$$

The ring-projection representation, $p(r)$, for the template is computed from its composite detail subimage as the sequence of the transform means at radius r from its center

$$p(r) = \frac{1}{n_r} \sum_{k=1}^{n_r} f_d^{(J)}(r \cos \theta_k + r \sin \theta_k)$$

where $r = \{r : 1 \leq r \leq \text{Radius}_{\text{template}}\}$ and n_r is the number of transform samples at radius r .

This projection is also computed for each of the highest energy pixels in the composite detail subimage for the search image, again, to the radius of the template.

The proposed discriminator for use with this filter is the normalized correlation.

Given the radial-projection of the template $P_T = \{p(1), \dots, p(n)\}$ and those of candidate sites

in a sampled image $P_S = \{\hat{p}(1), \dots, \hat{p}(n)\}$, the normalized correlation function was defined as follows:

$$\rho_S = \frac{\sum_{r=1}^n [p(r) - \mu_T][\hat{p}(r) - \mu_S]}{\left\{ \sum_{r=1}^n [p(r) - \mu_T]^2 \cdot \sum_{r=1}^n [\hat{p}(r) - \mu_S]^2 \right\}^{\frac{1}{2}}}$$

where

$$\mu_T = \frac{1}{n} \sum_{r=1}^n p(r) \quad \text{and} \quad \mu_S = \frac{1}{n} \sum_{r=1}^n \hat{p}(r)$$

The normalized correlation did not work well as a discriminator for this application. Squared absolute error did not work very well, either. I discovered single pattern examples that failed to find their source images as best match. The source images score well, but not the best. The suspicion is that the radial-projection transform does not work well in the type of clutter produced by a Perlin noise field as signal to noise ratio affects the probability of detection. It may also be that the number of points in the projection is too small. In other words the scale of the experiment and the detail available is not adequate for this approach.

Revisiting the problem with convolution filters applied in multiple orientation did no better. It was observed that noise correlated with the templates as readily as the features of interest. Simple magnitude comparison using normalized correlation did not appear to be adequate.

Researching the problem more carefully located the Ph.D. thesis of Arica Nafiz (2003) on shape recognition. Nafiz's study observes that template matching is, in fact, susceptible to noise.

In the same test images where the normalized correlation function worked poorly, the mean-squared error estimate worked fairly well. The algorithm was able to correctly locate each of the test objects. Returning to Perlin field images where the signal to noise ratio is significantly worse, however, the mean-squared error estimate did not work well. In the places where the recognition process decided, it was observed that the local curvature of the error surface was very low. This was in contrast to the very high local curvature of the error surface observed in the estimator in the test cases. The hypothesis would be to modify the discriminator to include a gain term that varies as a function of the local curvature of the error surface for an apparent match. It is not known at this writing if there is an optimization for this type of modification.

The alternative for searching in the absolute image would be to first extract the edges to be searched. This would reduce the noise. Naively applying a turtle to attempt extraction, it was discovered that turtles routinely became lost using the magnitude tracking heuristic on 3×3 neighborhoods. So there is a more complex heuristic involved with the recognition of these paths.

At this point several approaches were experimented with to improve the performance of the correlation filters.

- Signal thresholding was applied by processing only those signal components which exceeded a certain level.
- Contrast stretching was used to weight the distribution of signal components to emphasize those that appear above a certain level while suppressing those that appear below this level.

In the case of signal thresholding, sharp boundaries are introduced at the threshold. This creates high frequency components in the wavelet transform. This appears as noise in the discrimination surface and increases the false alarm rate of the detector. The spline used in contrast stretching on the other hand, produces smooth transitions between extremes. In this sense it acts as a bandpass filter with smooth cutoffs. This generates smoother discrimination surfaces realizing the benefit of reduced background noise without introducing additional artifacts.

In the case of contrast stretching, a curve is introduced to perform the mapping between the old and new signal distributions. In advanced applications, the shape of this curve is available to be edited by the user. An analytic curve in the form of a Hermite cubic spline contained in the unit square and having unit tangent vectors was immediately available. Mapping based on this curve was observed to produce satisfactory contrast stretching for the purposes of this experiment. Repeating the correlation experiments, the results did not significantly improve.

Additional research in this area reveals that the correlation problem is such that researchers are now studying recognition algorithms that explain what they mean by drawing pictures of their interpretation. When a recognition algorithm reports a good correlation, what it draws as its interpretation is visually compared with the subject pattern. The two must match for the algorithm to be accepted (citation). Developing such a method is beyond the scope of this thesis.

The planes used by the Perlin field to generate images appear to form a set of basis vectors. As an alternative to genetic search for the elementary patterns, the question that occurs is whether or not it is possible to derive a representation for a given image in terms of these basis vectors. We would attempt to derive a representation for the pattern directly by manually studying the effect of varying the direction of individual normals in a generator. In

the demonstration application, this is called the Perlin field explorer. What we discover almost immediately is that the absolute value of a Perlin noise image is not symmetric. Lines enter from the upper right and leave on lower left. No variation changes this outcome. So, some combination of field representations is required to overcome this difficulty. Again, we reach the limits of what we can attempt in this direction.

3.6.2 Mirror curves as lattice gas automata

A difficulty encountered with Matthew Kaplan's implementation of Computer generated Celtic design and Christian Mercat's presentation of his labeling algorithm, is the requirement to be able to visualize interlace in terms of it's crossings. An interesting observation made by Gerdes (1990) with respect to mirror curves is ,

once this common construction principal formulated, it became possible to find a whole class of single closed curve that satisfied the same principle.

Part of the fascination of this exploration method is that the outcome may not be immediately obvious. If we return to the figure in Chapter two from Lindisfarne folio 26, it was not apparent at the time of setup that this design is not made from a single cord but rather three. Nor was it apparent when exploring breaks that the following pattern is a single knot:

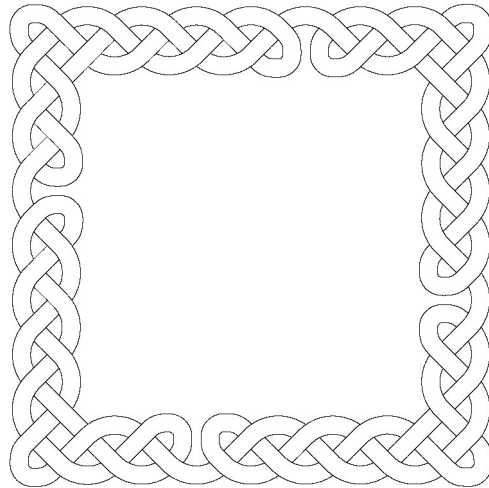


Figure 51: Single knot three cord braided border

The idea allows the study of patterns and questions about what a structure and its breaks can produce that would otherwise be difficult to examine. What is interesting is that we can represent the mirror curve as a lattice gas automata. We begin by examining a simple knot:

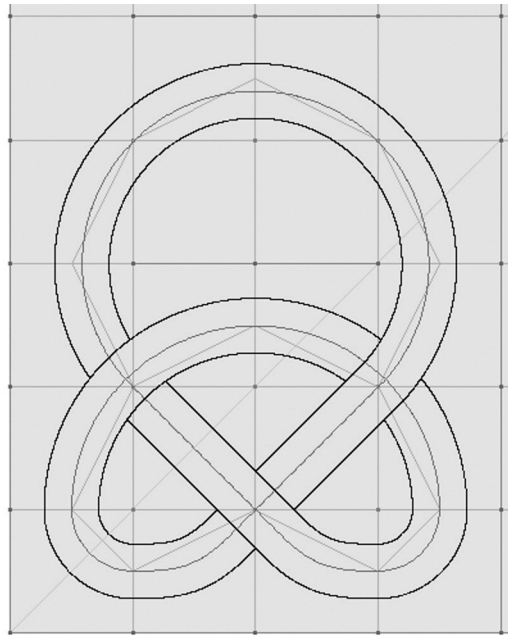


Figure 52: Simple knot

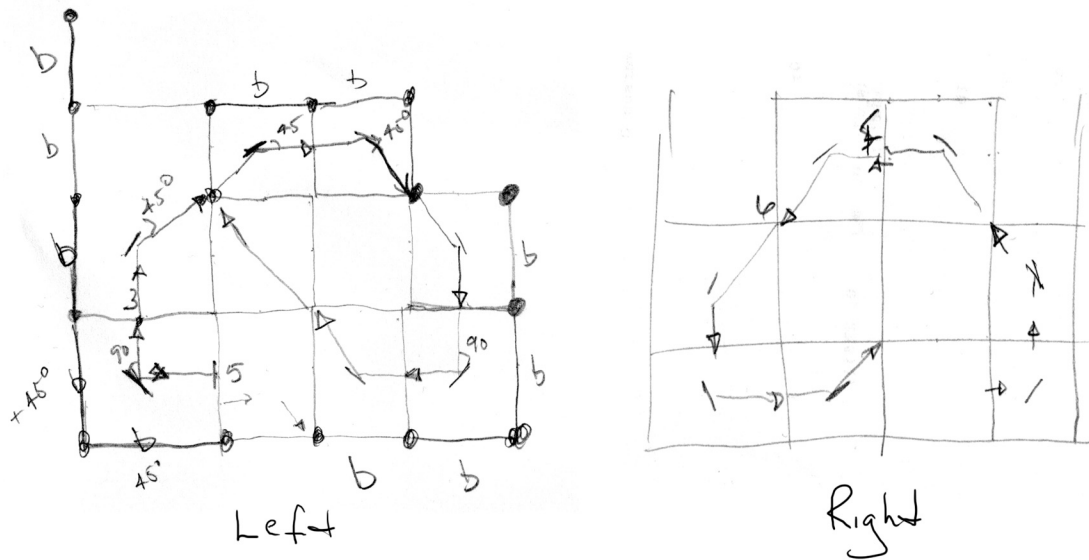


Figure 53: Derivation of mirror curve moves

The algorithm will operate on the cells of the subdivision of the major grid.

Suppose we want to model left reflection, in other words assume mirrors are on our left. If the chain code direction of a particle is five, and we are in the lower left corner, we notice there are two breaks. The direction we want to go is chain code three, making the 90-degree right turn to a vertical heading. Once we are there we have a single break to the left. This time the direction we want to go is chain code two, making a 45-degree turn to the right on to the diagonal. This time we enter a cell from the diagonal. We again want to turn 45 degrees to the right for a chain code heading of one. Chain code directions continue as eight, seven, five, and four. At this point we find ourselves entering a cell without breaks from a diagonal. This time we will want to move forward without changing direction. In other words, a particle in motion will continue in motion unless influenced by something (Sir Isaac Newton, First law of motion, *Philosophiae Naturalis Principia Mathematica*). The chain code move will again be four.

Physically we are modeling a particle that interprets a break as a 45-degree direction change. This interpretation is additive. If we see two breaks at a cell, depending on which edges they occur, we will expect a 90-degree turn. Notice also that possible moves are limited to five, two moves left and right of center. If we begin with this move set, we can cull moves based upon the location of breaks on the current cell. A culling function is used to code the configuration of breaks for the particle on the current cell. The particle interprets this as a break count that corresponds to five cases: 1-edge, 2-corner, 0-channel, 4-box, and 0-unrestricted. We have already examined the edge and corner cases. The channel case arises when movement is restricted by breaks left and right, and the particle is only able to continue in the open direction of the channel. The box case is when the particle reaches a dead end with breaks on three sides. With the five-move model the box case results in a halt. In a more general case of the model, the box causes a reflection that reverses the direction of the particle. For the five-move model we code the next move prediction for left reflection as

$$left_move_i = \text{mod}(left_move_{i-1} - 1 + 8 - break_count, 8) + 1$$

Following a similar analysis for the right move, we code the next move prediction for right reflection as

$$right_move_i = \text{mod}(right_move_{i-1} - 1 + break_count, 8) + 1$$

The union of these two results predicts the set of moves available for leaving the current cell.

The following initial move set is sent for cell configuration culling:

$$i \in \{5, 6, 7, 8, 9\}$$

$$move_set_i = \text{mod}(move_{i-1} - 1 + i, 8) + 1$$

The result from the culling intersected with the set of predicted moves forms the set of moves actually possible. From this final set a move is chosen at random. The expected cardinality of the final set is 1.

For initialization the last move can be calculated by reversing the direction that the user gives for an initial direction and examining the moves available from the corresponding cell(s).

If a last move is known, its reverse may be calculated as

$$\textit{opposite_move} = \text{mod}(\textit{move} - 1 + 4, 8) + 1$$

With this derivation we have a complex systems method that simulates mirror curve path generation. Applying Thurston's theorem we can assign the crossings of a valid Celtic interlace. Christian Mercat's labeling algorithm can be viewed as an implementation of Thurston's theorem.

3.6.3 Mercat's labeling algorithm as lattice gas automata

The grid cells associated with the preceding example are highlighted in violet in the next figure:

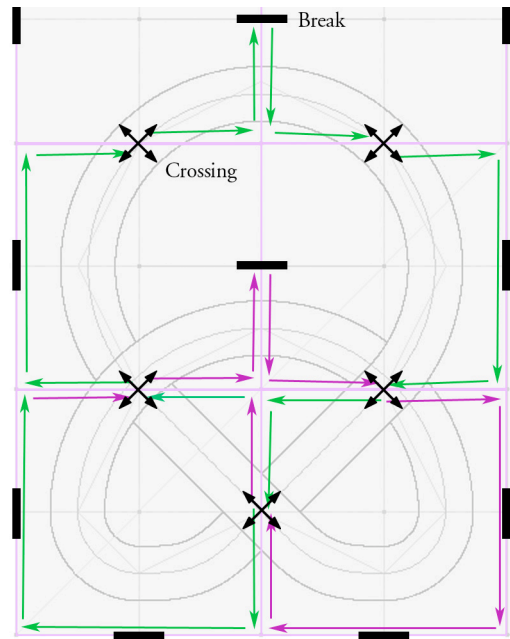


Figure 54: Example design grid

We note that crossings appear on the edges of the cells of the major grid that correspond to diagonals in the reflection model. These edges are the portions of Mercat's planar graph that correspond to this part of the design.

At first it is seductive to visualize a particle moving in subdivisions of the major grid while tracing its edges. The direction of the cell would be assigned as a vector corresponding to the direction it takes along the edge. The crossing connection association calculation would be performed each time a crossing marker appears between the particle and the next cell along the edge. It is the crossing marker that would determine the cell to which the particle would move next and control its crossings of edges of the major grid. However, careful inspection of the curve path indicates that the subdivision cells traced by this movement are the same as those traced by the mirror curve LGA. With this in mind, maintenance of the crossing labeling system should be allocated to the mirror curve LGA. This is convenient in that crossing states can be decided as the mirror curve LGA is used as a

planning tool by the designer. The avoids having to wait until curves are designed to analyze self-intersections and crossing shared with other curves. It also means that tangles that whose crossings are not resolvable can be detected at the earliest possible time.

A particle of the mirror curve LGA may be assigned a vector corresponding to the direction of the chain coded move leaving the current cell. If the departure leaves the cell along a diagonal, the corresponding crossing marker is retrieved and the dot product of the particle direction with the marker connection vectors are computed. The most negative result is the crossing connector associated with this move. A curve identifier is registered with that connector and a crossing state can be decided, at will in the case that this a new crossing marker, or as the state opposite that of a curve that has previously been registered with the crossing marker. When the corresponding guide curve is generated, the arc length associated with the curve at the crossing annotates its connections for later use by the renderer in resolving occlusions. Should the designer decide to delete the curve at some point, the annotations are backed out by retracing its chain coded moves as recorded by the mirror curve LGA when the curve was generated.

3.7 Summary

In this section we have developed an impulse model of curvature dynamics that allows us to accomplish genetic design of unit speed space curves. We have examined problems associated with designing the structure grids on which Celtic interlace is built. A template-based approach was found to avoid many of the difficulties associated with a number of other methods for designing grids. We reported the results of experiments on searching for spontaneous curves in Perlin noise and the difficulties associated with that. We developed an implementation of mirror curves based on lattice gas automata and

examined the integration of Christian Mercat's crossing label system into that behavior. In Chapter four we will present an overview of the demonstration system.

Chapter 4 Overview of the System

4.1 Introduction

The core of the demonstration application is an evolutionary design system that has been integrated with the workflow of a designer involved in Celtic design. Based on experience from the preceding chapters this person needs to do the following things:

- Work either on a complete design or in the frames of something larger
- Define structure
- Define elements of the design
- Arrange and pose these elements aided by the structure
- Solve the interlace, determine what will and will not work, and make adjustments accordingly
- Render the result

An evolutionary design system is organized around artificial life concepts from the area of complex adaptive systems. These systems normally consist of populations of

individuals whose interactions and or life cycles create dynamics that give rise to patterns. Components typically found in these systems include the following

- Individuals composed of a genotype that codes its design and a phenotype that is the result of the translation of the genotype into a physical, visual, or structural representation
- Genetic algorithms which regulate a life cycles process for such individuals
- Fitness functions or means for evaluating and guiding a work in progress

Departing from this precise model for this application, we have admitted all categories of complex systems methods. We have algorithms, such as lattice gas automata, that will also assist with the evolution of shape.

The core relationship between complex systems methods and design is that a composition is viewed as a population of individuals. A top-level view of the system is shown in the following figure:

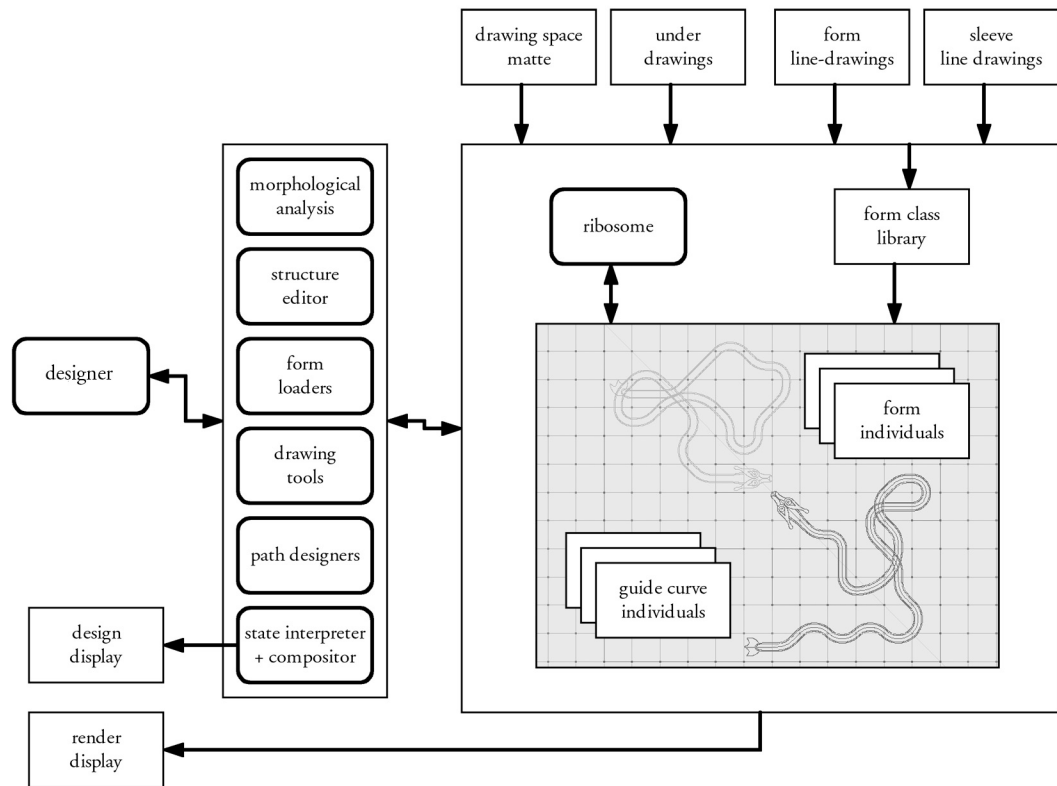


Figure 55: Top level system view

4.2 Organization

The system is organized around a set of user interface tools. The inputs to the system consist of a matte for the drawing space(s), optionally an under drawing, line drawings of forms to be used, and line drawings of sleeves that are used to shape interlace. Morphological analysis is used to analyze and provide an initial interpretation of the drawing space selected by the designer. Tools are provided to allow the designer to define and edit the structure of the Celtic design grid and the placement of breaks. The designer is also provided with tools for designing guide curves, applying them as

skeletal components for form posing, and recognizing irresolvable interlace coding conflicts.

At the heart of the system is an artificial life concept that provides flexibility and efficiency in exploring the complex shapes and relationships inherent to Celtic design. The discovery and evolution of curves, the representation of complex anatomy, their assembly and solution, their transfers to and from poses, and their rendering is facilitated by a linear genetic programming model.

4.3 Phenotypes

In the course of this research the composition itself has been variously viewed as a genetic algorithm and again as a phenotype. The forms and guide curves from which a design is assembled are more easily identified as individuals. Their phenotypes consist of both structural and visual components; although, each is tailored to its specific role. Forms consist of the strokes that draw the anatomy of the given creature. These are initially supported and posed by default guide curves that act as a skeleton. The phenotype of a guide curve is the solution of a unit speed space curve of assigned curvature that rests in the plane. In Chapter two we introduced medial line analysis for interlace. This is the visualization role played by guide curves in assisting a designer with exploring shape.

4.4 Genotypes

Genotypes code the specifications for the designs of individuals. Each genotype consists of a single chromosome that may contain one or more sequences. Each step in a

sequence is a statement from the instruction set of a linear genetic programming model. The genotypes of forms provide us with everything needed to solve and draw their respective anatomies around guide curves. This includes the sampling of strokes with respect to the symmetries of a default skeleton, labeling of strokes that form the exterior of the form, and the segmentation needed to shape the anatomy along the shape of guide curves presented by the designer for posing. The genotype of a guide curve provides a piecewise continuous specification for the curve in terms of an assignment of curvature as a function of arc length. Since these curves remain in the plane, torsion is set to zero.

4.5 Genetic algorithms and complex systems methods

For problems where the work is unstructured, such as the learning the parameters for the model of the curve being fitted between two arbitrary points or where the form of the curve is not analytically solvable, a genetic algorithm is appropriate. When designing curves with precise structure, a template-based method guided by lattice gas automata is appropriate. Both methods are provided for use at the discretion of the designer. The basic organization of a genetic algorithm is shown in the following figure:

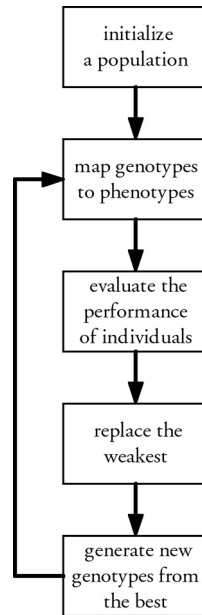


Figure 56: Genetic algorithm

This is a form of optimization procedure. In the case of the evolution of model parameters for curve shapes based on curvature dynamics, it was found that improvements could be made in the performances of individuals by progressively refining parameter values based on measures such as distance from the given goal and alignment with the goal's tangent; and this has been implemented.

4.6 User interaction

This system is not automatic in this sense that it generates design examples independently. This decision was based upon the literature review and its revelation of art and design as the personal intimate expression of a designer or artist. As a consequence we return to the advice set out by Peter Bentley (2001) the design of creative evolutionary design systems (CEDs):

- Find a domain in which creative enhancement by computer makes sense
- Find a good reason for using a creative system
- Evolve appropriate relationships between the roles of system and its user
- Find a niche the design process where the tool fits
- Make it generative and creative
- Make it understandable
- Have an easy and effective way of evaluating solutions and guiding their evolution
- *Find people who are actually prepared to use the system*
- Get lots of R&D money to pay for it
- *Start a company and make a billion*

The design of the present system has tried to be in conformance with at least the first seven recommendations; and corresponding tools have been provided to assist the designer or artist in performing the tasks of design

4.7 Evaluation

Our review of the literature found that automatic means for evaluating the quality of visual designs remains an open question. In the present case, visual comparison with hand drawn work is the primary means of determining the value of a technique. This is related to the method called pile sorting described in the research by

Isenberg (2006) in which participants are given a number of illustrations and asked to sort them into piles based on how well they believe the visual objective has been reached.

4.8 Summary

This chapter provides an overview of an evolutionary design system integrating complex systems methods with the design tasks of a human designer. We have introduced the major features of the design and the model of artificial life and user interaction at its core. The workflow, structures, and algorithms will be explained in Chapter 5.

Chapter 5 Interactions Representations and Processes

5.1 Introduction

In this chapter we will summarize details of the design of the demonstration system. This will include the user work flow introduced in Chapter 4, tools and processes supporting the work flow, and representations. We will begin with a discussion of structure. We will then look genotypes beginning with forms and Sleeves. We will then discuss guide curves and their design. We will then look at phenotypes. After that we will discuss form binding. Will next turn our attention to the genetic programming model, its representation of chromosomes, the definition of nucleotides, and how these are processed in the biological analog. We will conclude with a discussion of rendering.

5.2 Structure

The application is launched by typing *cgcc* at the Matlab command line. The initialization routine runs and then dispatches the hub panel of the user interface. The work flow associated with the design of structure is depicted in the following figure:

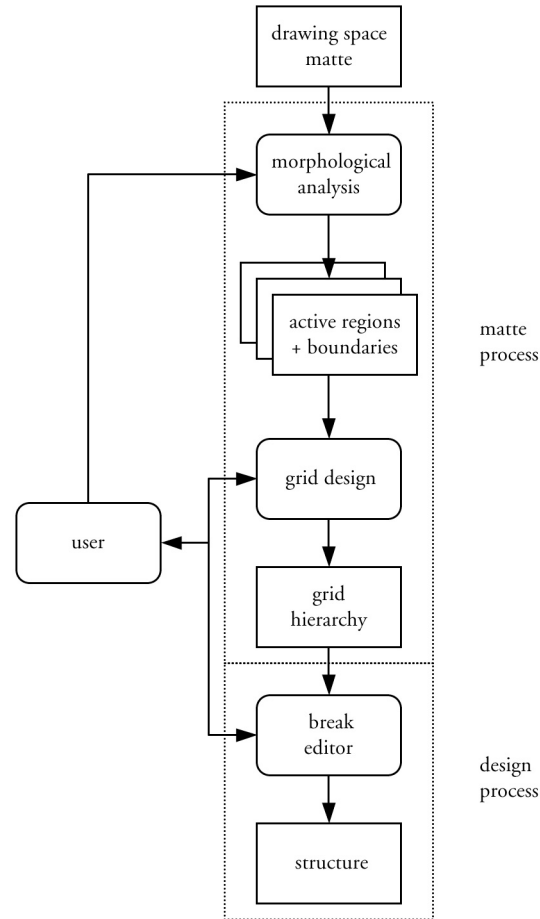


Figure 57: Structure load and editing work flow

Structure design is divided into two parts, the matte process and the design process. In the matte process the user loads an image that represents the space of the entire design. The regions in which work is to be performed are matted white. These are anti-aliased. The morphological analysis is automatic. This loads and displays a thumbnail of the image. It then locates each of the work regions, computes borders, a Delaunay triangulation for reference, and catalogs the regions. If it finds more than one region, a dialog is presented for the user to select which is to be active first, makes that region active, and displays it. It next asks which level of the hierarchical grid the user wishes to work on and dispatches the grid editor panel for that level. The active region

is now available for grid design. To install a Muller-Hannemann type 1 template (rectangular with proportional subdivision), the user selects, in order, the corners of the boundary and presses *make mesh*. A dialog is presented for the user to specify the number of grid cells to be installed in each direction. The grid system builds this grid as a mesh, renders the grid and updates the display. The subdivision used for the construction is displayed; and the user can select this as a target. If the user wishes to build another level in the grid the editor panel can be closed, the grid button on the hub panel can be pressed again, and the process repeated. A depiction of the process on the user interface is depicted in the following figure.

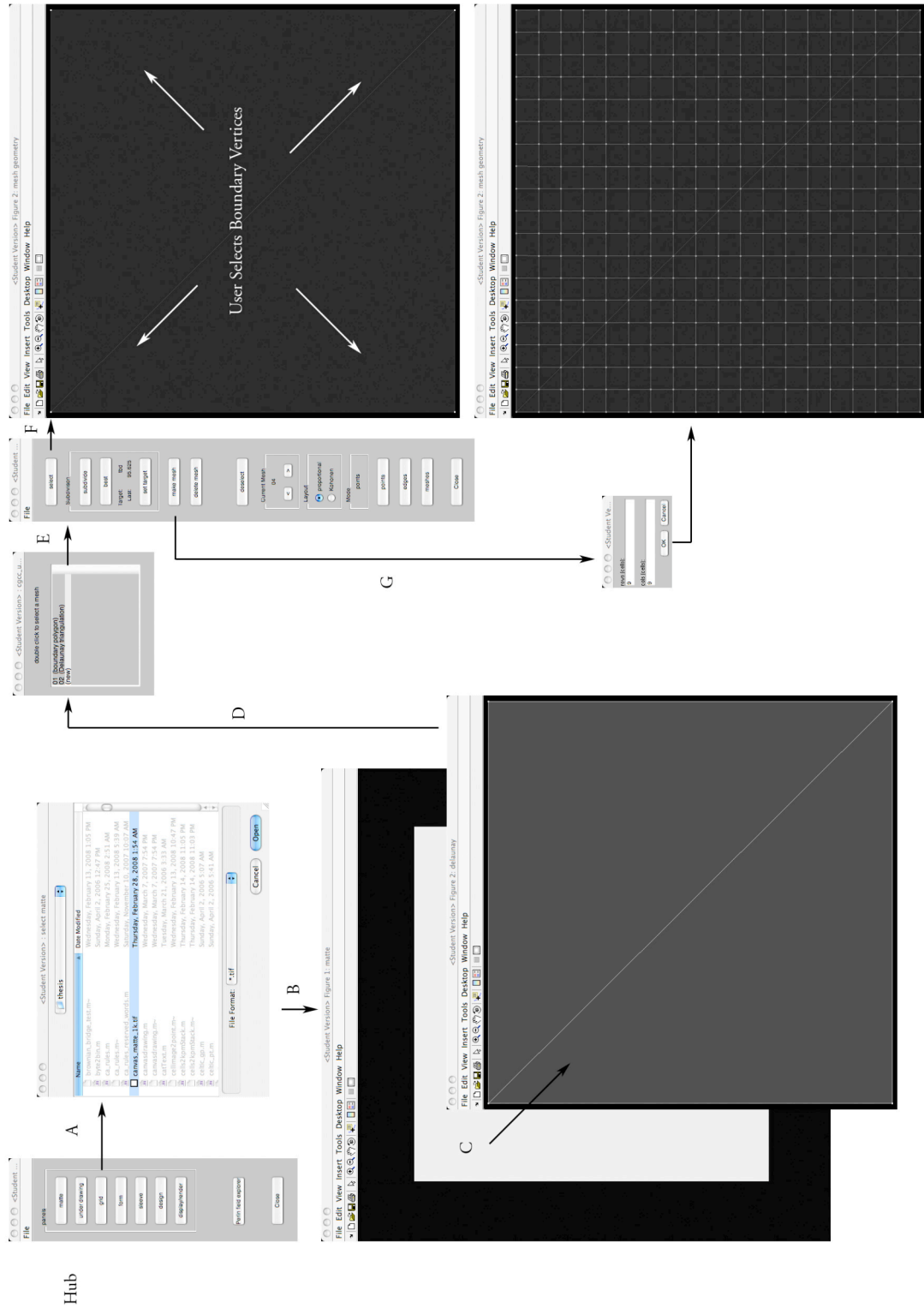


Figure 58: Use interaction for matte processing and grid design
Steps in this diagram are as follows:

- A. Select grid from the hub—this automates the work flow. Matte just opens a matte.
- B. Thumbnail of the entire design image
- C. Automated morphology through borders and triangulation
- D. Selection of a grid level to design
- E. The grid design editor—creates or deletes grids
- F. Select tool—modes are points, edges, grids
- G. Make mesh and it's result

Note that shuttle buttons are available to allow the user to navigate through the entire grid hierarchy. When this level the grid is completed according to the needs of the designer, it is placed in the grid table.

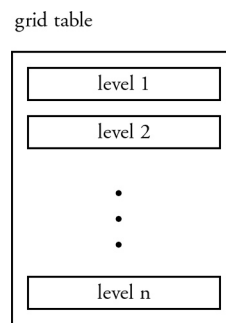


Figure 59: Hierarchical grid table

The grid table is organized as a simple global symbol table where each level of the grid has been assigned a numerical identifier. Each level of the grid is designed as a simple structure and contains the following fields:

A grid (one level) in the hierarchical grid table:

id: 3	Grid identifier
pred: 2 succ: 0	Parent/child identifiers
rendered: 1 im_mesh: <image filename> im_mesh_subdiv: <image filename> im_mesh_vtx: <image filename>	Display caching: Grid image Grid subdivisions Grid vertices
subdiv_last_dist: 42.5 subdiv_target_dist: 42.5	Last subdivision size Target subdivision size
breaks: [1x1 symbolTable]	Breaks table--edge identifiers with breaks
faces: [1x1 symbolTable] eList: [1x1 symbolTable] fEdges: [1x1 symbolTable] vEdges: [1x1 symbolTable] dEdges: [1x1 symbolTable]	Geometric complex--face polygons edge list edges associated with each face edges associated with each vertex dual edge table--directed edge pairing
type: 'proportional'	grid subdivision type
dim: [18 18]	grid dimensions in subdivisions
m_cage: [9x9 double] v_cage: [19x19 double] f_cage: [18x18 double]	Mercat label table matrix of vertex identifiers matrix of face identifiers

Table 1: One level in the grid

5.3 Forms and sleeves

The work flows for forms and sleeves are the same. Forms are line drawings that are intended to be segmented so that anatomy can be distributed consistently with how the form is to appear. Sleeves provide anatomy for interlace. They are intended to be stretched between two points along a guide curve and segmentation is not used. The idea is that the designer is free to draw these shapes as they need to be without necessarily relying on a drawing package. Each form is specified in terms of four files: anatomy, symmetry, segmentation, and pivots. The anatomy file defines what the form looks like. The symmetry file defines its skeleton. The segmentation file, if it contains anything, defines where the form can be stretched. The pivots file is intended to specify where parts of the anatomy may be independently turned. At this writing guide curves

provide all of the structure and the pivots file is there for future study. One further restriction is that the skeletal guide curves must initially be defined as straight lines. This restriction can be lifted with more work. All of the files have a common root name with their content denoted by suffix. The two work flows are depicted in the following figure:

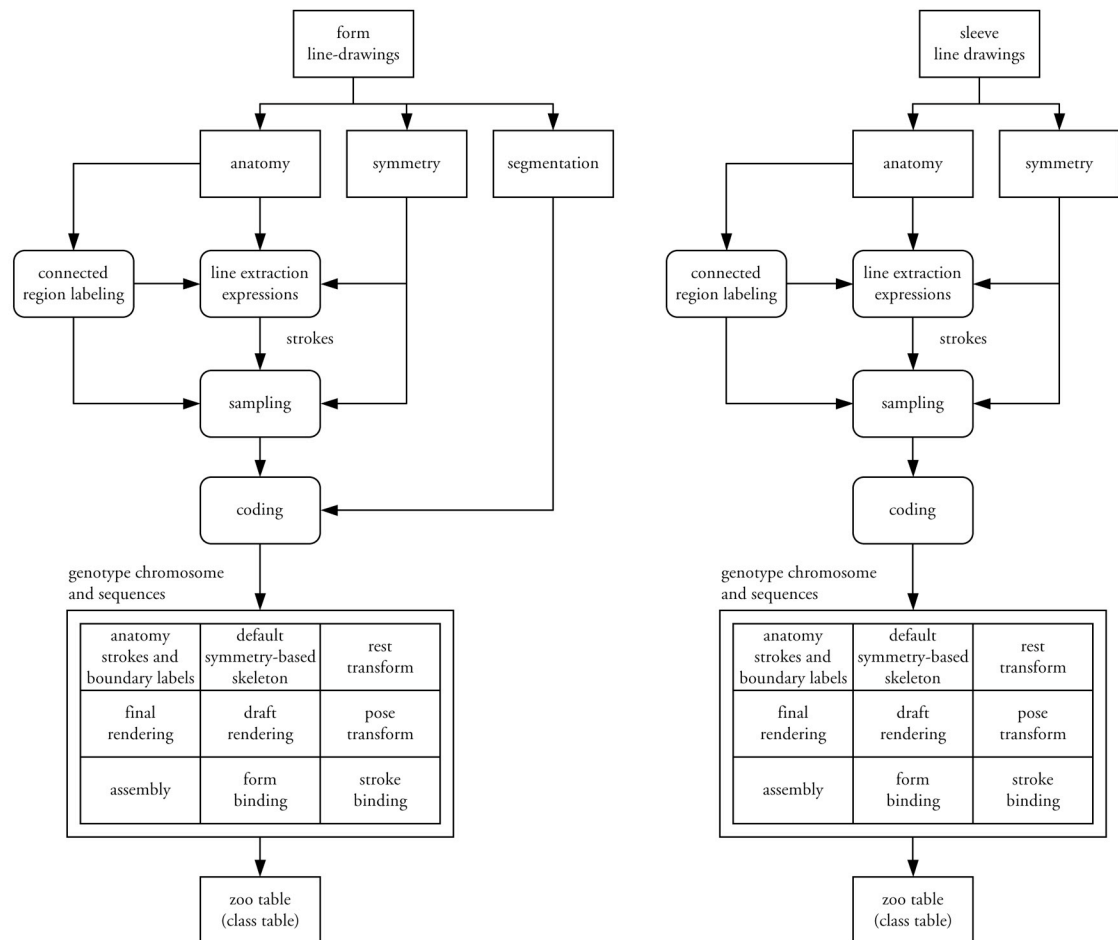


Figure 60: Form load path (left) Sleeve load path (right)

The user begins the process from the hub by pressing either one of the form or the sleeve buttons. The form editor reads the anatomy and symmetry files. It then asks the user which is the main symmetry and what it is called. It then reads the segmentation

and pivot files and opens a window in which to display extracted features. At this point the user can do one of two things for extracting features. The user can specifying the morphology for each of the features by way of a dialog or the morphology for the features can be listed in a file and batch processed.

The anatomy file is processed for connected regions. Each of these regions is numbered and displayed with the file as follows:

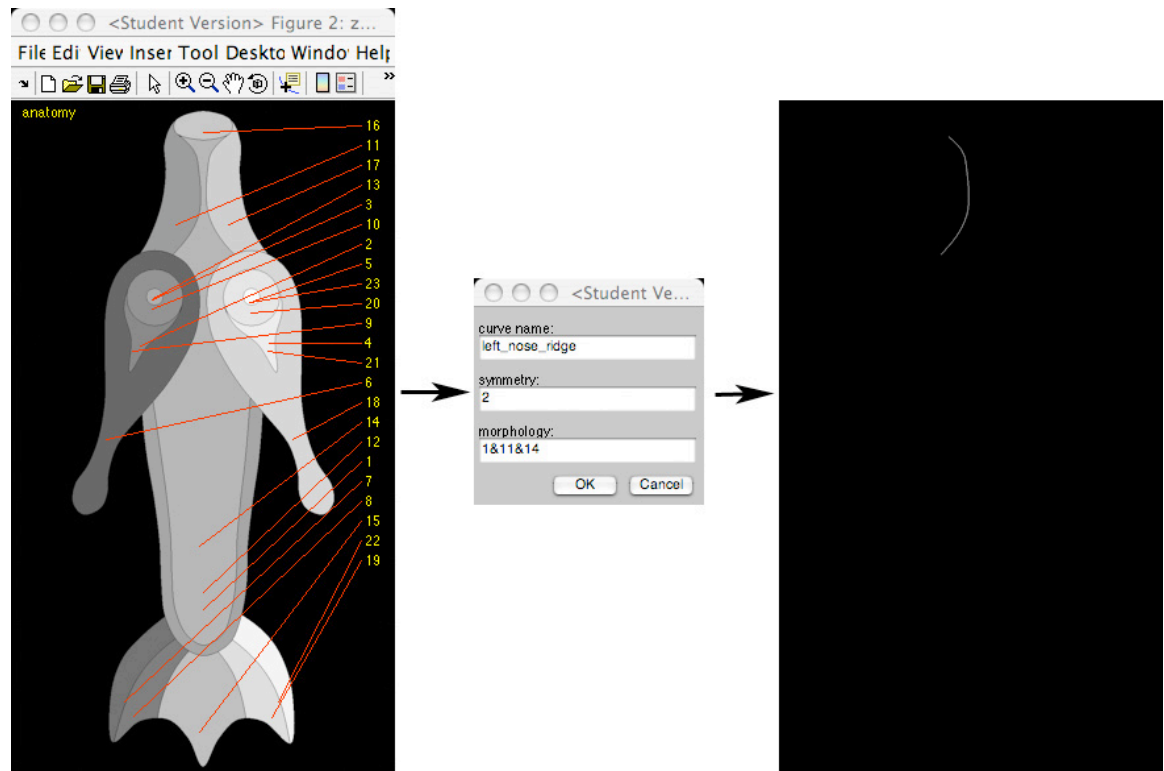


Figure 61: Stroke extraction from anatomy file

The expression given is: 1&11&14. This means that the stroke is defined as belonging to the regions labeled 1, 11, and 14. The editor parses and translates this expression and then evaluates it to capture the stroke shown on the right and its image is saved for sampling.. The same information for the entire form (a Celtic snake) can be placed in a file for batch processing. The contents of the file for everything appears as follows (~ means not):

```

left_nose_ridge  2  1&11&14
right_nose_ridge 2  1&17&14
snout            2  1&16
left_lip         2  1&11&~16&~14&~6
right_lip        2  1&17&~16&~14&~18
outer_body_lower 2  1&12&~14&~18&~6
inner_body_lower 2  1&12&14
tail_le          2  1&7&~8&~12
tail_li          2  1&8&~7&~15&~12
tail_m           2  1&15&~8&~19&~12
tail_ri          2  1&19&~15&~22&~12
tail_re          2  1&22&~19&~12
left_lappet      1  1&6
left_eye         1  2&10
left_pupil       1  3
left_eyelid      1  2&9&~10
right_lappet     3  1&18
right_eye        3  4&20
right_pupil      3  5
right_eyelid     3  4&21&~20

```

The extracted features are as shown in the following figure and displayed in the features window. The most recent is highlighted.

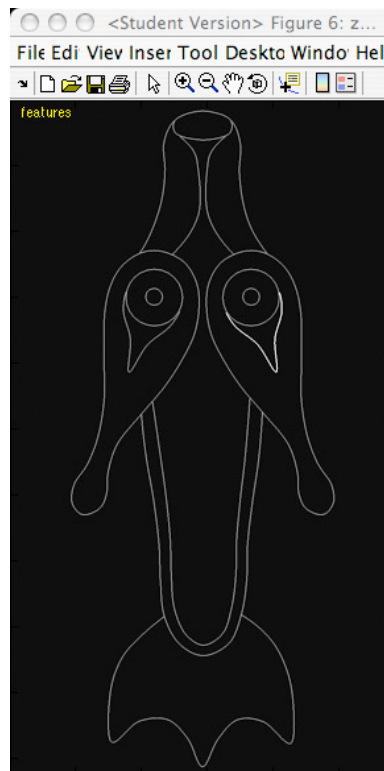


Figure 62: Extracted features of the Celtic snake

Since the labeling of the connected regions is somewhat coarse, the user may also request that these be broken out. This function produces thumbnails of all of the regions and their assigned numbers. The user can also examine the boundaries of the imagery contained in each of the files.

When all of the features have been extracted, processing proceeds to sampling and coding. The symmetry for each line is rotated so that its length is aligned with the y-axis. The angle will be $\varphi = \cos^{-1}((v_2 - v_1) \cdot u_y)$, where v_1 is the lowest point on the symmetry and v_2 the highest. This allows arc length along the symmetry to be measured without distortion. Each feature is rotated by this angle on the same center.

Unfortunately this is not a perfect process. Breaks in continuity (holes) have been observed to appear. This is due in part to mapping distortions that have been studied.

When rotating feature points the following transform by Danielsson and Hammerin (1992) is used:

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{pmatrix} 1 & -\tan \frac{\varphi}{2} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ \sin \varphi & 1 \end{pmatrix} \begin{pmatrix} 1 & -\tan \frac{\varphi}{2} \\ 0 & 1 \end{pmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

Unfortunately even this is not quite enough and the procedure recommended by Sung-Il Chien (2001), of rendering a line by means of scan conversion between points whose distances is separated by two or more pixels, is applied to assure that no holes appear. The distance from each stroke to its symmetry is then sampled as a function of arc length of that symmetry. Resolution is user defined. This defines each stroke by its distance from its symmetry. When all of the features have been sampled, the minimum and maximum of samples along the respective symmetries is measured and the samples for each feature are relocated such that the assembled form along each symmetry begins

at $s_0 = 0$ and ends at $s_1 = \max(s_{samples}) - \min(s_{samples})$. Where s_0 and s_1 become the beginning and ending arc lengths of the respective symmetries. This same procedure is followed with boundary labels for each of the strokes. Pixels that form the exterior boundary of the anatomy will guide occlusion processing during rendering.

The coding process maps the assembly and binding of a form into a genotype. Each genotype consists of a single chromosome that contains one or more sequences. Each sequence consists of a list of nucleotides. Each nucleotide is an instruction in prefix format from the instruction set of a linear genetic programming model. A sequence may be viewed a simple function that can be interpreted by a translating ribosome. Sequences may be viewed formally as follows:

```
sequence := nucleotide_list
nucleotide_list := [ nucleotide ] end_of_sequence_mark
nucleotide := { operator [ argument ] }
argument := quoted_string | symbol | number | expression | array
expression := { operator [ argument ] }
operator := quoted_string
```

[] denotes zero or more of the enclosed symbols

No precedence is applied in processing expressions.

Arguments are separated by spaces

Sequences are coded as linear Matlab cell arrays. As a consequence no parsing is required prior to translation. The ribosome simply fetches cells sequentially from a given array. Each of these cells represents a nucleotide. Each nucleotide is a cell array. The ribosome maps the first string in each nucleotide to the name of an interpreter and

passes the argument list. The `end_of_sequence_mark` is simply the end of the cell array from which the ribosome was fetching nucleotides. Operators of the LGA are listed in the next table.

Arithmetic Operators		Functions	
Operator	Description	Description	Syntax example
'+'	Addition		{ 'magnitude' [1 1] }
'-'	Subtraction	Matlab function	{ 'eval' <matlab expression> }
'*'	Multiplication	Matlab function	{ 'ones' [1 1] }
./	Multiplication--element wise	Matlab function	{ 'zeros' [1 1] }
/'	Division	Matlab function	{ 'padarray', 'myArray', 'myPad' }
'transpose'			{ 'round' '15.5' }
Relational Operators		Form statements	
Operator	Description	Description	Syntax example
'<'	Less than		{ 'bind', 'mySymmetry', 'myNewGUIDName', 'mySequenceName' }
'<='	Less than or equal to		
'>'	Greater than		
'>='	Greater than or equal to		
'=='	Equal to		
'~=='	Not equal to		
Logical Operators		Objects	
Operator	Description	Description	Syntax example
'&'	And		{ 'new', 'intrinsicCurve', 'guideName' }
' '	Or	creates instance	{ 'set', 'myCurve', 'name', 'value' }
'~'	Not or complement	sets an attribute value	{ 'get', 'myCurve', 'name' }
'xor'	Exclusive or	deletes the object	{ 'delete', 'myCurve' }
		loads a sensor data file	{ 'sensor', 'image_curves_sample_sets' }
Geometric transform statements		Genetic statements	
Operator	Description	Description	Syntax example
'translation'	translate the object		{ 'translate', 'sequence' }
'rotation'	rotate object in plane		{ 'clone', 'myCurve', 'newCurve' }
'reflection'	reflect across line		{ 'mark' }
'glide-reflection'	glide reflect across line		{ 'push', 'sequence' { 'solve', 's3b' } }
			{ 'sequence', 'translateThisOne' }
Space Curves		Evolutionary statements	
Operator	Description	Description	Syntax example
'curvature'	set curvature		{ 'bridge_0', 'myCurve', 'myGoal', 'learningRate' }
'impulse'	schedule an impulse	Genetic bridge--curvature	{ 'bridge_1', 'myCurve', 'myGoal', 'learningRate' }
'ramp'	schedule a ramp	Genetic bridge--curvature rate	{ 'bridge_2', 'myCurve', 'myGoal', 'learningRate' }
'extend'	extend the arclength	simple interpolating loop	{ 'loop_0', 'myCurve', 'myGoal', 'learningRate' }
'solve'	solve curve or form	constrained interpolating loop	{ 'loop_1', 'myCurve', 'myGoal', 'learningRate' }
'arclength'	set the arc length	Adcock's symmetrical loop	{ 'loop_2', 'myCurve', 'myGoal', 'learningRate' }
'arclength_at_k'	get arc length at cell k		
'position_at_k'	get position at cell k		
	For ramps: parm, parm_dot		
Assignment		Image handling	
Operator	Description	Description	Syntax example
'='		Red channel	{ 'R', 'im_cont' }
		Green channel	{ 'G', 'im_cont' }
		Blue channel	{ 'B', 'im_cont' }
		returns image of a curve	{ 'image_of', 'curve' }
		mask image with mask	{ 'mask_image', 'img', 'cont', 'im', 'D1' }
		Matlab function	{ 'find_values', 'im', 'D1', 'k' }
		intersect two images	{ 'intersection', 'im', 'S1b', 'im', 'S1b' }
Conditional statement		Rendering	
Operator	Description	Description	Syntax example
'if' <expr>	<nucleotide>	Matlab Painter	{ 'antialias', 'off' }
		Production renderer	{ 'render', 'myCurve', 'im', 'lga_curve', 'disp', 'image_dim', { 'Overlay', 'on' } }
		Draft renderer	{ 'final_render', 'myCurve', 'im', 'lga_curve', 'size', { 'Overlay', 'on' } }
		turns it on or off	{ 'antialias', 'on' }
		sets visibility state	{ 'visible', 'myCurve' }
Branch		Space Curves	
Operator	Description	Description	Syntax example
'jump'	relative to current position		{ 'jump', '-6' }

Table 2: Linear genetic programming model operators

Geometry for design of the skeleton for the Celtic snake is typical. Using the primary symmetry as the reference, the offsets to the starting points for each of the lateral symmetries is calculated and from this their originating positions. A back of the envelope illustration of the calculation is shown as follows:

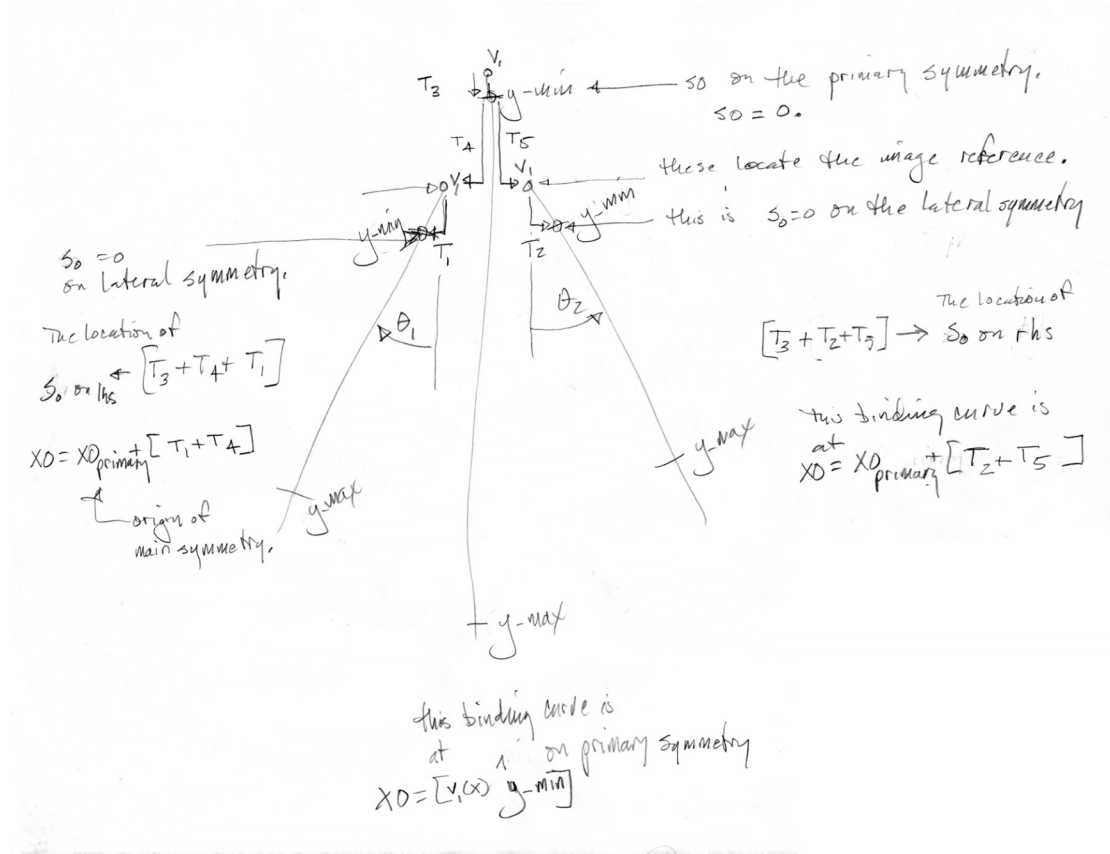


Figure 63: Snake skeletal geometry

Unit speed space curves are designed for each of these symmetries and the designs coded as part of the assembly sequence of the genotype with the rest position being the origin in image space the last step in the sequence is to translated to the center of the active region where the user can inspect it. Fragments from each of three sequences are shown for examples in the following text:

```
% Part of the skeletal design for the main symmetry
% '0' origin of the world coordinate system (image space)
```

```

% theta1 rotation of a guide curve to that of the symmetry
% theta_lateral rotation of a lateral guide to the main symmetry
% s0_ref_wcs translation of curve's design origin to the wcs origin
% s1 length of main symmetry with respect to the anatomy

skel_seq = push(skel_seq, { 'new' 'point' 'O' } );
skel_seq = push(skel_seq, { 'new' 'intrinsicCurve' sc_name } );
skel_seq = push(skel_seq, { 'rotation' sc_name 'O' theta1 } );
);
skel_seq = push(skel_seq, { 'rotation' sc_name 'O' theta_lateral } );
);
skel_seq = push(skel_seq, { 'translation' sc_name { '*' s0_ref_wcs 'scale' } );
);
skel_seq = push(skel_seq, { 'extend' sc_name { '*' s1 'scale' } );
);

% transform the design to its display position in the canvas
% this is part of the 'form_from_rest' sequence

locn_seq = push(locn_seq, { 'rotation' sc_name 'O' 'disp_orientation' } );
locn_seq = push(locn_seq, { 'translation' sc_name 'disp_translation' } );
locn_seq = push(locn_seq, { 'solve' sc_name } );

% this is part of the 'form_to_rest' sequence

rest_seq = push(rest_seq, { 'translation' sc_name { '*' -1 'disp_translation' } );
);
rest_seq = push(rest_seq, { 'rotation' sc_name 'O' { '*' -1 'disp_orientation' } );
);

% form_from_rest is used initially to relocate the design to active
% region center. When the form is rebound to a user designed guide
% curve, this transform is used to relate it to the origin of the new
% guide curve.

% form_to_rest is always used to return a form to the origin of the
% world coordinate system

```

The newly coded chromosome is saved along with the raw design data as a class (my nomenclature for prototypes of a form). This is organized as a Matlab structure:

form: zoomorph & sleeve structure:

attribute and example data	description
name: 'snake_lod3'	
specs: [1x1 symbolTable]	cross ref for spec image names
curves: [1x1 symbolTable]	extracted curves by name
dna: [1x1 chromosome]	coded chromosome
dim: [1432 817]	dimension of the form image
scale: 1	

Table 3: Form or zoomorph design data structure

The data for an extracted curve is also organized as a Matlab structure:

extracted curve data:

attribute and example data	description
name: 'inner_body_lower'	
label: 2	symmetry identifier
text: 'l&12&14'	morphological expression
expr: ' (zoo_anatomy.L == 1) ... & imdilate(zoo_anatomy.L == 12, g dil) & imdilate(zoo_anatomy.L == 14, g dil)'	and its expansion
fn: 'snake_lod3_inner_body_lower'	extracted curve's internal name
dna: [1x1 symbolTable]	curve DNA
lss: [104x2 double]	left hand side samples
rss: [105x2 double]	right hand side samples
lss_ext: [104x1 double]	left hand boundary label samples
rss_ext: [105x1 double]	right hand boundary label samples

Table 4: Extracted design curve data

Form specification images are stored as a Matlab structure in the global file table. The organization of this table for the anatomy image of the Celtic snake is as follows:

form anatomy image data:

attribute and example data	description
fn: 'snake_lod3_anatomy'	file name
ext: 'tif'	extension
pn: '/Applications/MATLAB_SV71/user/thesis'	path name
im: [1432x817 double]	the image array
dim: [1432 817]	the image dimension
B: {23x1 cell}	extracted boundaries
B: {23x1 cell}	extracted boundaries
L: [1432x817 double]	connected regions image
N: 5	number of boundaries
A: [23x23 logical]	adjacency matrix of nested objects

Table 5: Form anatomy image data

Please note that the contents of the fields B, L, N, and A, are as specified by Matlab for the `bwboundaries` function.

The organization of the structure for the symmetry image is a bit more complicated. This image has been annotated with the specifications for all of the symmetries. This includes all of the skeletal design data. The organization of this structure is shown as follows:

```
form symmetry image data:
attribute and example data
```

attribute and example data	description
fn: 'snake_lod3_symmetry'	
ext: 'tif'	
pn: '/Applications/MATLAB_SV71/user/thesis'	
im: [1432x817 double]	the image array
dim: [1432 817]	the image dimensions
specs: [1x1 symbolTable]	complete symmetry specifications
specs_by_name: [1x1 symbolTable]	specification name reference table
B: {3x1 cell}	
L: [1432x817 double]	
N: 3	
A: [3x3 logical]	
eList: [1x1 symbolTable]	edge form of the symmetry curve

Table 6: Form symmetry image data

A specification in the symmetry image specs table is organized as a Matlab structure. The fields for the symmetry on the left side of the Celtic snake appear as below:

form symmetry specification and design data:

attribute and example data	description
name: 'snake_lod3_symmetry_01'	
label: 1	symmetry label identifier
segment_table: [1x1 symbolTable]	table of segments
pivot_table: [1x1 symbolTable]	table of pivots
parent: 2	parent identifier
decendants: []	children identifier
rhs_mask: []	temporary sample mask storage
lhs_mask: []	temporary sample mask storage
plane: 'lateral'	anatomy name
v1: 362	starting position as a vertex
v2: 363	ending position as a vertex
v1c: 364	starting position in cartesian space
v2c: 365	ending position in cartesian space
angle: 18.0849789549418	angle with respect to Uy
v2p: 366	starting position in sample space
v1p: 362	ending position and sample space
aligned_fn: 'snake_lod3_im_symmetry_01_aligned'	image aligned with Uy
min_y: [363 432 325 405]	minimum stroke extents
max_y: [484 590 909 440]	maximum stroke extents
y_min: 325	minimum stroke extent
y_max: 909	maximum stroke extent
s0_ref_ds: [2x1 double]	S ₀ in design space
T_ds: [2x1 double]	translation from main symmetry start
Twcs: [2x1 double]	WCS origin in design space
s0: 0	S ₀ of design--minimum arc length
s1: 584	S ₁ of design--maximum arc length
s0_ref_wcs: [2x1 double]	relocation to WCS origin
theta1: 90	rotation from new curve default
theta_lateral: 18.0849789549418	angle with respect to main symmetry
scale: 1	scale
s_s0: 0	scaled S ₀
s_s1: 584	scaled S ₁
s_s0_ref_wcs: [2x1 double]	scaled relocation to WCS origin

Table 7: Symmetry specification (left) and skeletal design data

As the symmetry image was used to hold the specifications for symmetries, so the segmentation image is used to hold the specifications for segmentation lines. The structure that holds the segmentation image has the same organization as that for the symmetry image. The specification for a segmentation line, as it is stored in the specs table of the segmentation image, is again contained in a Matlab structure. An example of this is as follows:

form segmentation specification:

attribute and example data	description
name: 'snake_lod3_segmentation_03'	name
label: 1	region label
parent: 1	symmetry identifier
v1: 377	left position as a vertex
v2: 378	right position as a vertex
t_symmetry: 0.600760970660353	parameter of intersection with symmetry

Table 8: Segmentation specification data

Having waded through all of this detail and before we return to the user interaction, it may be worthwhile to remind ourselves of why we've gone to this much effort. The widths that we've just sampled define the strokes of the anatomy for our Celtic snake and will be used by the parallel line algorithm theorem from our work on differential geometry in Chapter two. Once we have solved a guide curve, we can solve the positions of the strokes for a snake's anatomy as lines parallel to the guide curve. We can then render the snake as the guide curve has posed it. A summary of the user interaction involved with loading a form or sleeve is shown in the next figure with the following steps

- A. Selection of the forms loader
- B. Selection of the form or sleeve
- C. Specification of the main symmetry
- D. Manual curve extraction
- E. Batched curve extraction
- F. Separated regions display
- G. Boundaries display

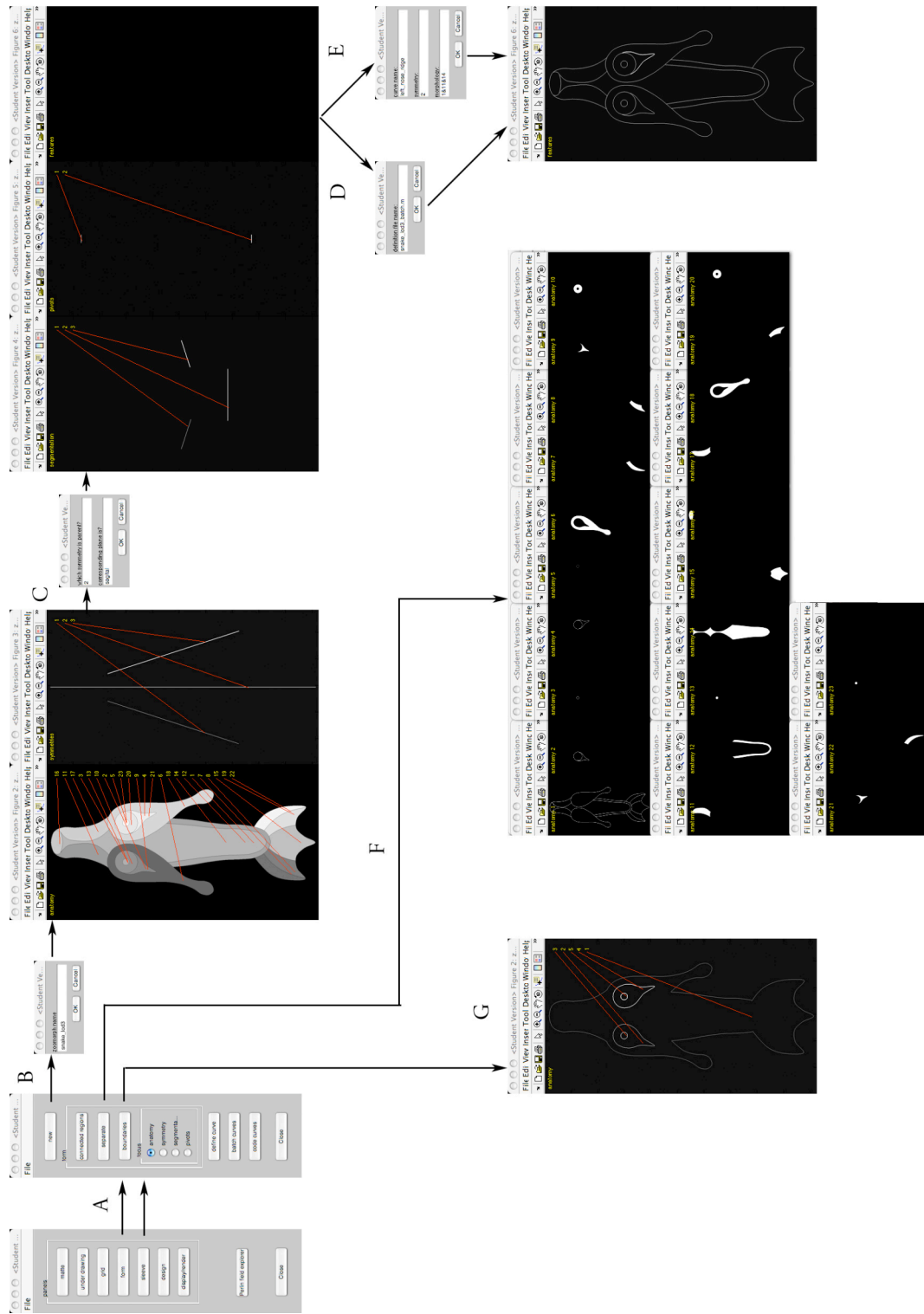
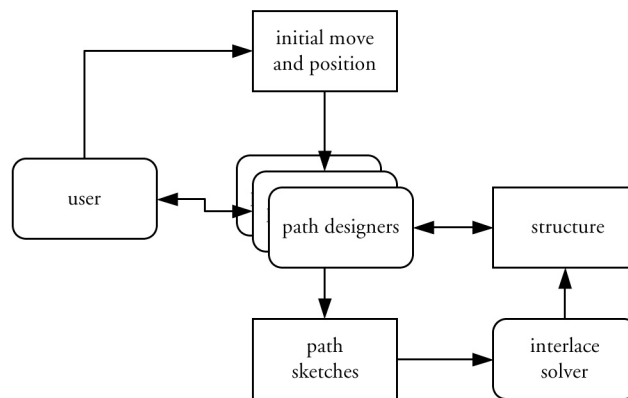


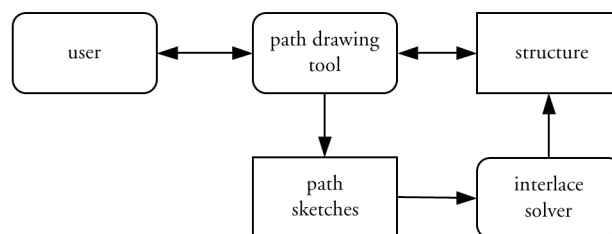
Figure 64: User interaction for form and sleeve loading

5.4 Guide curves

Guide curves provide the means by which a designer poses forms and knotwork. They are designed on a grid. The grids used in demonstration to this point have been regular. The grid, if there was one, in the lozenge of folio 34r of the Book of Kells is irregular in shape and, hence, in sizes and shapes of its subdivisions. During the course of design and testing, many interesting guide curves were observed with lattice gas path generators. Reconstruction of these curves turned out to be complicated by the semantics of the breaks especially in the presence of other guide curves. With these things in mind, there is a need for three paths in the work flow. The first two paths are on regular and irregular grids assisted by a guide curve path designer and the third is on an irregular grid freeform:



Regular and irregular grids with lattice gas path designer



Regular and irregular grids freeform

Figure 65: Work flow paths for guide curve sketching (Mercat interlace solver)

These work flows produce study paths in the form of polyline renderings of curves developed by the respective methods. The work flow for the regular grid has been implemented; and an example of a study curve from that process with a 5-move lattice gas designer is as follows:

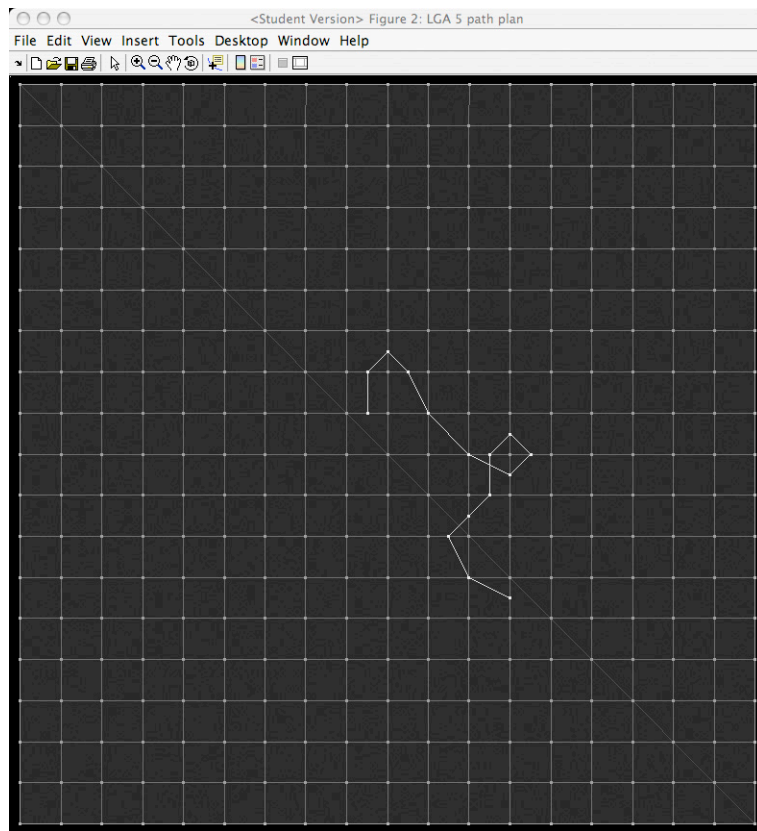


Figure 66: Study curve example

Note that without Mercat's interlace solver, resolution of interlace ordering is delayed until study paths have been explored and a guide curve constructed. Simulation of a cellular image based method for interlace order solving was the first implemented.

When something interesting appears, the designer tells the system to generate the curve. The processes involved are shown as follows:

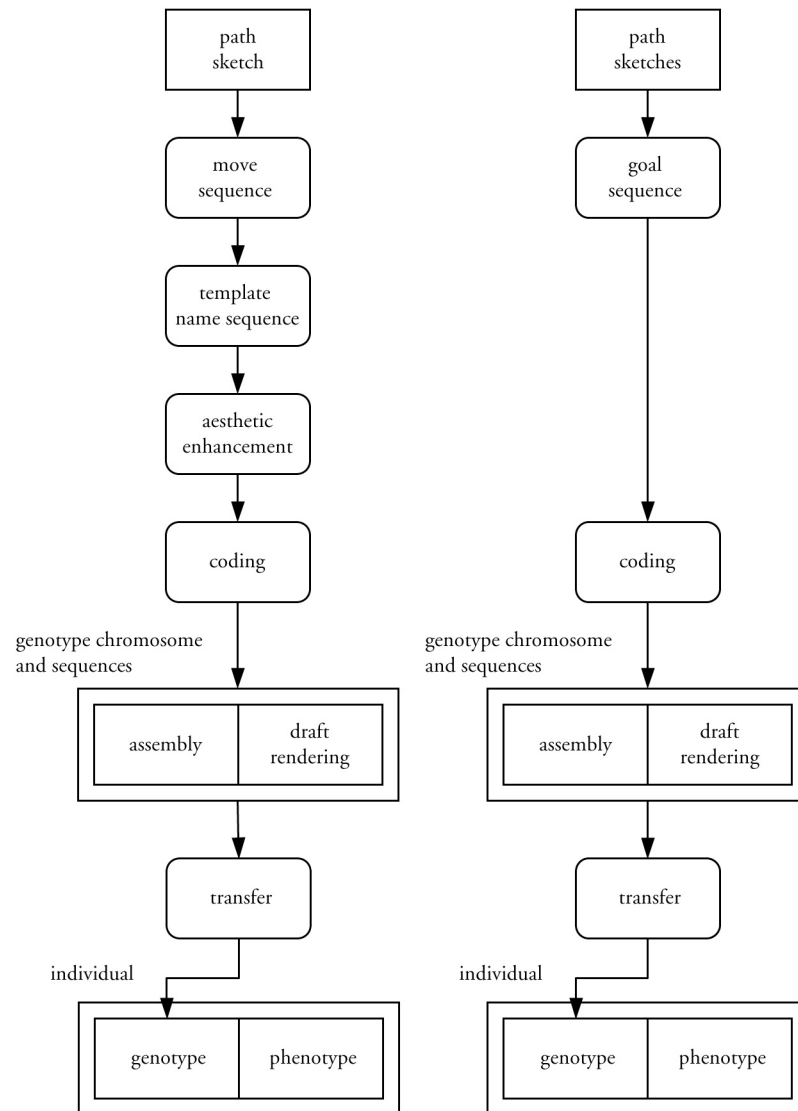


Figure 67: Guide curve generation for regular grids (left) irregular and freeform (right)

On a regular grid with a lattice gas path designer, the chain coded move sequence is obtained from the design for the study curve. This data is organized as a Matlab structure and for the preceding study curve appeared as follows:

study curve design data:

attribute and example data	description
name: 'guide_curve_0004'	
designer: 'cgcc_design_lattice_gas_5_move'	where the curve came from
generator: 'template'	coder to be used
linetype: 'open'	
vtx: [1x1 kpmStack]	vertex list for display render
ip: [2x1 double]	initial position
ip_cell: [9 9]	
include_ip: 'no'	initial position not include in curve
goal_list: [2x15 double]	goal positions of each move
move_list: [3 3 1 7 8 8 1 3 5 7 7 5 7 8 1]	generated move list

Table 9: Study curve design data

We will now describe each of the curve generation paths beginning with our example curve. Since the grid was regular, we can use templates for each segment of the curve and the move sequence is sufficient to define these. The first step is to translate the move sequence into a list of template names and add this to the design. The initial template set was defined as follows:

curve segment design templates:

name	description
'e2er'	edge to edge right
'e2dr'	edge to diagonal right
'e2es'	edge to edge straight
'e2dl'	edge to diagonal left
'e2el'	edge to edge left
'd2dr'	diagonal to diagonal right
'd2er'	diagonal to edge right
'd2ds'	diagonal to diagonal straight
'd2el'	diagonal to edge left
'd2dl'	diagonal to diagonal left

Table 10: Initial curve segment template names

The template needed for a particular curve segment is determined by looking at move pairs, as the previous move tells us whether we came from an edge or diagonal. If we were to generate the curve based on these templates we would get the following result based on a very early example. Notice the boxy shape of the curve:

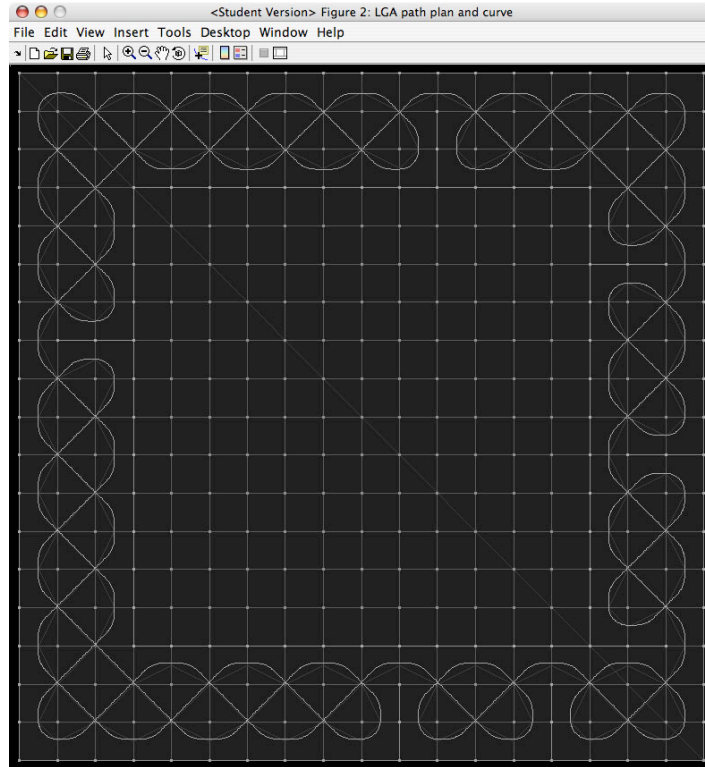


Figure 68: Guide curve without aesthetic enhancement

Most of the curves drawn in insular manuscripts are rounded unless the designer intended otherwise. One of the things we can do is to convert template pairs that move from diagonal to diagonal through an edge into a circular arc wherever we can infer that this is consistent with the curvature. We can do the same thing for template pairs that move from edge to edge through a diagonal. Note that the curvature of this second arc will be different from that of the first. One of the problems we will face with this second change is in the u-bends. If we alter the symmetry of a u-bend, we will change the arc length of the curve and the ends of closed curves will be broken. This can be avoided by delaying a pair search process by two steps after encountering a u-bend.

This replacement strategy is implemented as an aesthetic enhancement pass. This is similar to what an L-system does in that we look for pairs of template names

(strings) that correspond to the two types of transitions we've identified. Given that the u-bend constraint has been satisfied, we replace these pairs with the names of templates that generate the corresponding circular arcs. Unlike an L-system a simple sequential state machine scan performs the pass. For a border similar to the one above, the effect of this pass is as follows:

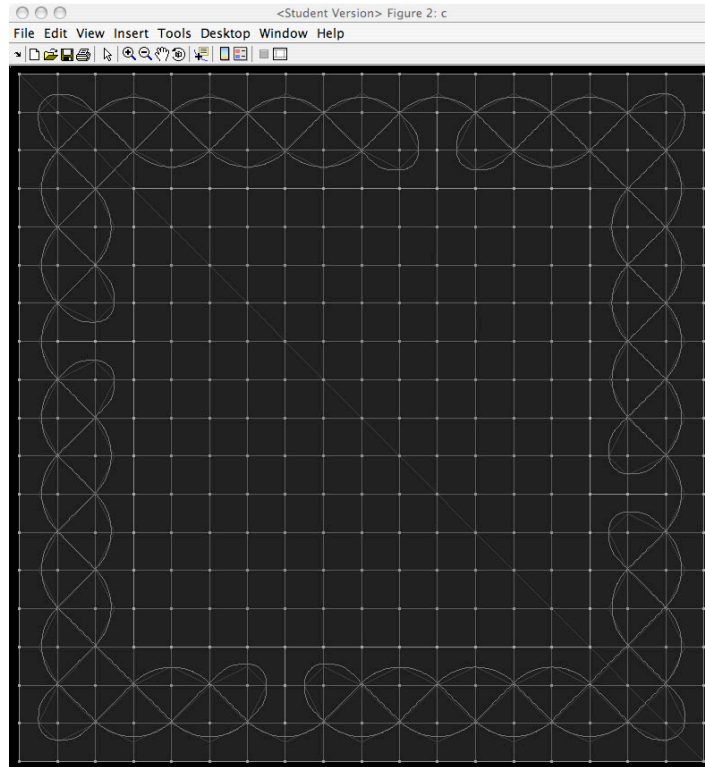


Figure 69: Guide curve with aesthetic enhancement

The boxy appearance of the u-bends themselves is moderately improved by changing the design of the edge to diagonal (and reverse) templates from a model based on circles and lines to one based on opposing clothoid splines. This involves designing opposing curvature ramps by a method proposed by Marko Pamali (2004). These are installed to make the transition; and even though a short line segment is still needed for the fit, the shape of the line is improved.

The design for our example study curve is sent through this process. Each of the templates is translated to nucleotides to build an assembly sequence for the guide curve. A render sequence is also built. The assembly sequence for our study curve turned out as follows:

```
K>> cgcc_inspect_sequence(seq)
{ 'new' 'point' '0' }
{ 'new' 'intrinsicCurve' 'guide_curve_0004' }
{ 'rotation' 'guide_curve_0004' '0' -90.000000 }
{ 'translation' 'guide_curve_0004' [2x1 double] }
{ 'curvature' 'guide_curve_0004' 0.000000 }
{ 'extend' 'guide_curve_0004' 42.500000 }
{ 'curvature' 'guide_curve_0004' 0.047059 }
{ 'extend' 'guide_curve_0004' 33.379422 }
{ 'curvature' 'guide_curve_0004' 0.047059 }
{ 'extend' 'guide_curve_0004' 33.379422 }
{ 'curvature' 'guide_curve_0004' 0.000000 }
{ 'ramp' 'guide_curve_0004' 20.465860 'parm_rate' -0.001875 }
{ 'ramp' 'guide_curve_0004' 20.465860 'parm_rate' 0.001875 }
{ 'curvature' 'guide_curve_0004' 0.000000 }
{ 'extend' 'guide_curve_0004' 8.802038 }
{ 'curvature' 'guide_curve_0004' 0.000000 }
{ 'extend' 'guide_curve_0004' 60.104076 }
{ 'curvature' 'guide_curve_0004' 0.000000 }
{ 'extend' 'guide_curve_0004' 8.802038 }
{ 'ramp' 'guide_curve_0004' 20.465860 'parm_rate' -0.001875 }
{ 'ramp' 'guide_curve_0004' 20.465860 'parm_rate' 0.001875 }
{ 'curvature' 'guide_curve_0004' -0.047059 }
{ 'extend' 'guide_curve_0004' 33.379422 }
{ 'curvature' 'guide_curve_0004' -0.047059 }
{ 'extend' 'guide_curve_0004' 33.379422 }
{ 'curvature' 'guide_curve_0004' -0.047059 }
{ 'extend' 'guide_curve_0004' 33.379422 }
{ 'curvature' 'guide_curve_0004' 0.000000 }
{ 'extend' 'guide_curve_0004' 42.500000 }
{ 'curvature' 'guide_curve_0004' 0.047059 }
{ 'extend' 'guide_curve_0004' 33.379422 }
{ 'curvature' 'guide_curve_0004' -0.047059 }
{ 'extend' 'guide_curve_0004' 33.379422 }
{ 'curvature' 'guide_curve_0004' -0.015686 }
{ 'extend' 'guide_curve_0004' 50.069133 }
{ 'curvature' 'guide_curve_0004' -0.015686 }
{ 'extend' 'guide_curve_0004' 50.069133 }
{ 'solve' 'guide_curve_0004' }
```

Table 11: Study curve assembly sequence

An individual is then instantiated and the chromosome is transferred as its genotype. The next step is to solve this individual. The work necessary to do this is depicted in the following diagram. Note that we include the path for solving the

interlace without Mercat’s algorithm; and we will provide a description of how that works:

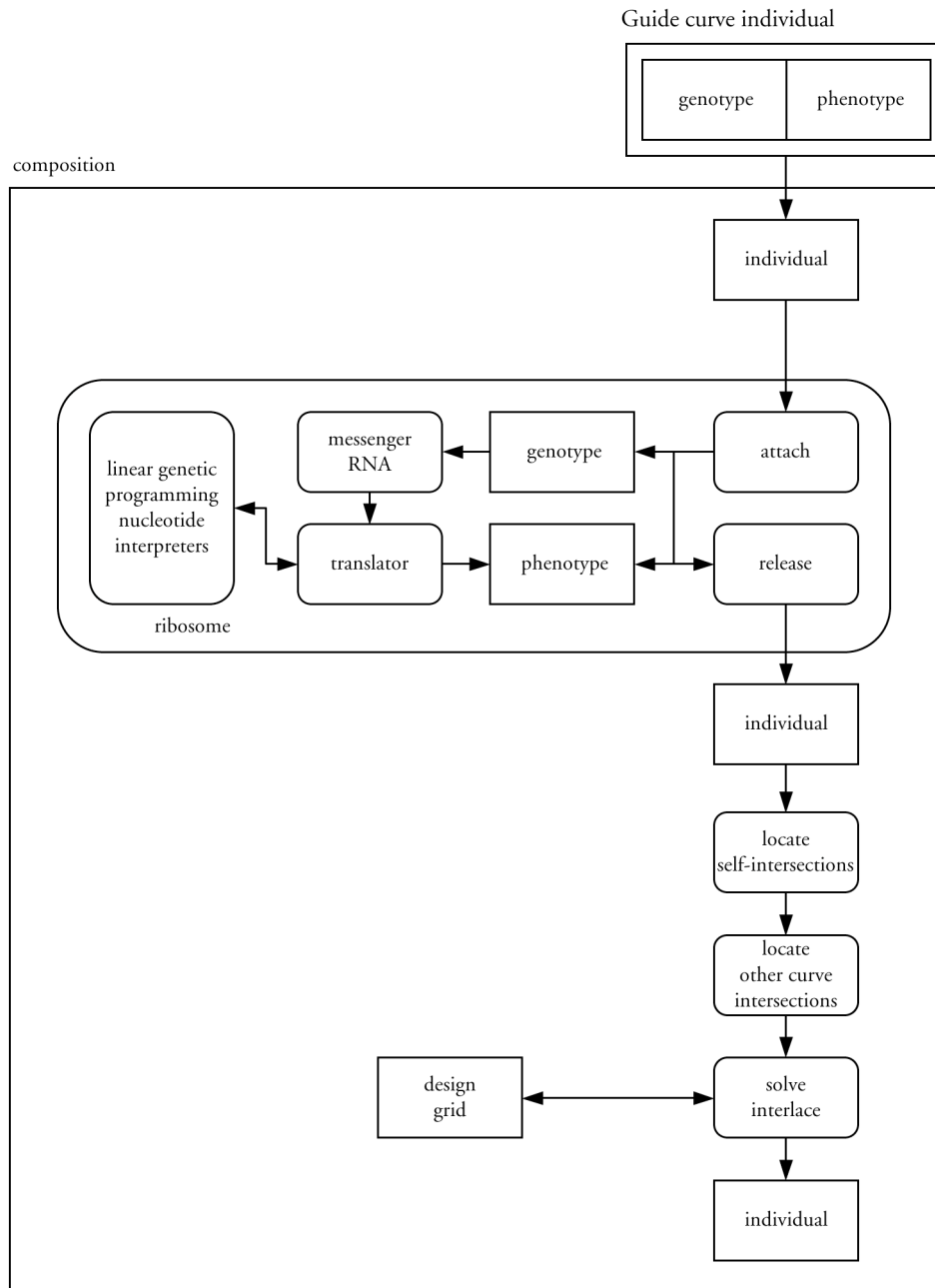


Figure 70 : Guide curve phenotype construction (without Mercat)

The individual is attached to a ribosome that is asked to translate the genotype. A segmented space curve is assembled by a sequence of curvature assignment events generated by the nucleotides. These events compose an analytical description of the curvature function on an interval table. And, yes, the biggest challenge is being sure that all of the boundary conditions are correctly resolved. Space curves are instances of a Matlab object. Attributive behaviors of this one are as follows:

segmented space curve with assigned curvature:

attribute and example data	description
name: guide_curve_0004	
segments:	segment table
object is intervalTable	
eos: true	iterated access end of list state
index: 17	iterated access index
tos: 18	number of segments in the table
allowDubs: false	no duplicates allowed
t0: cell	interval start list
t1: cell	interval end list
values: cell	interval data list
s0: 0.000000	arc length of curve start
s1: 578.365811	arc length of curve end
X0: [3x1 double]	initial position
T0: [3x1 double]	initial tangent
N0: [3x1 double]	initial normal
B0: [3x1 double]	initial binormal
s: 0.000000	arc length of requested solution
Xs: [3x1 double]	position at requested arc length
Ts: [1x3 double]	tangent at requested arc length
Ns: [3x1 double]	normal at requested arc length
Bs: [3x1 double]	binormal at requested arc length
Ks: 0.000000	
Ks_dot: 0.000000	
rez: 0.200000	resolution
S: [116x1 double]	arc length samples of solved curve
X: [116x12 double]	curve solution values at S(k)
solved: true	solution state

Table 12: Space curve with assigned curvature

An example of a segment from the interval table is as follows:

example of a curve segment on its interval:

attribute and example data	description
t0 = 1.092588e+02	start of interval (arc length)
t1 = 1.297247e+02	end of interval (arc length)
p0 = 0	parameter at interval start
p0_dot = -8647481222736871 ----- 4611686018427387904	rate at interval start
p0_dot_dot = 0	acceleration at interval start
p1 = -3.837602e-02	parameter at interval end
p1_dot = -1.875124e-03	rate at interval end
p1_dot_dot = 0	acceleration at interval end
ps = 8647481222736871 33242680327154436644147317727663 - ----- s + ----- 4611686018427387904 162259276829213363391578010288128	analytic form of parameter f(s)
ps_dot = -8647481222736871 ----- 4611686018427387904	analytic form of rate f'(s)
ps_dot_dot = 0	analytic form of acceleration f''(s)
ps_dot_dot_f = 0	indefinite form of acceleration
event = motionEvent	the scheduling motion event

Table 13: Segment from space curve interval table

This structure remembers the event that scheduled it. Motion events are instances of Matlab objects. The contents of the one here appear as follows:

example of a motion event:

attribute and example data	description
eventTime: 1.092588e+02	event start (arc length)
eventParm: 'parm_rate'	event parameter
eventEnd: 1.297247e+02	event end (arc length)
totalChange: -1.875124e-03	total change in event parameter
keyInterval: 0	not used

Table 14 : Motion event data

The nucleotide the does the solve directs the intrinsic curve to its solve behavior. The intrinsic curve obtains a solution by way of the Matlab differential equation solver, ode45, using the equations for the Frenet frame that we developed in Chapter two. The steps size used by ode45 must be set small enough so that none of the curve segments are overlooked. The solution is returned in two arrays. The array S, shown earlier, contains the values of arc length at which the solution was performed. The array X, also shown earlier, contains a solution for each of the arc length values in S. The solution is rendered for display; and all of this data becomes part of the phenotype for the guide curve individual.

The next step in the absence of Mercat's algorithm is to find all of the intersections in the composition. This was implemented as an image based process because of the difficulty of otherwise working with sampled curves. A turtle (eight-state machine) was designed to walk the solution of the curve while watching the image of the curve. The turtle records all of the places where the curve crosses itself. These positions are marked in a self intersection map. The second pass examines intersections with other curves. This is done by forming the combinations of all of the curves in the composition two at a time. We then take the image intersection of each pair and record the shared locations. We aggregate all of these locations into an intersection map for the composition. We again walk the solutions for all of the curves with a second turtle. This time with crossings objects assigned at each of the intersection locations. Each time the turtle encounters an intersection it registers the curve it is working on and assigns a crossings state taking into consideration of valid choice for the curve as well as any other curves that have already been registered at the intersection. Should a conflict in the interlace solution be found, the turtle backs out the solution of the curve that it's working on, changes its first choice of crossing states, and tries again. If it fails on the

The interlace solution and the distance weight images can be inspected through the display system and appear in the next diagram. With the interlace solution, black is below and white is above and state changes are made halfway between intersections and end points. With weights, intensity is 1 at intersections and 0 at midpoints or the beginning:

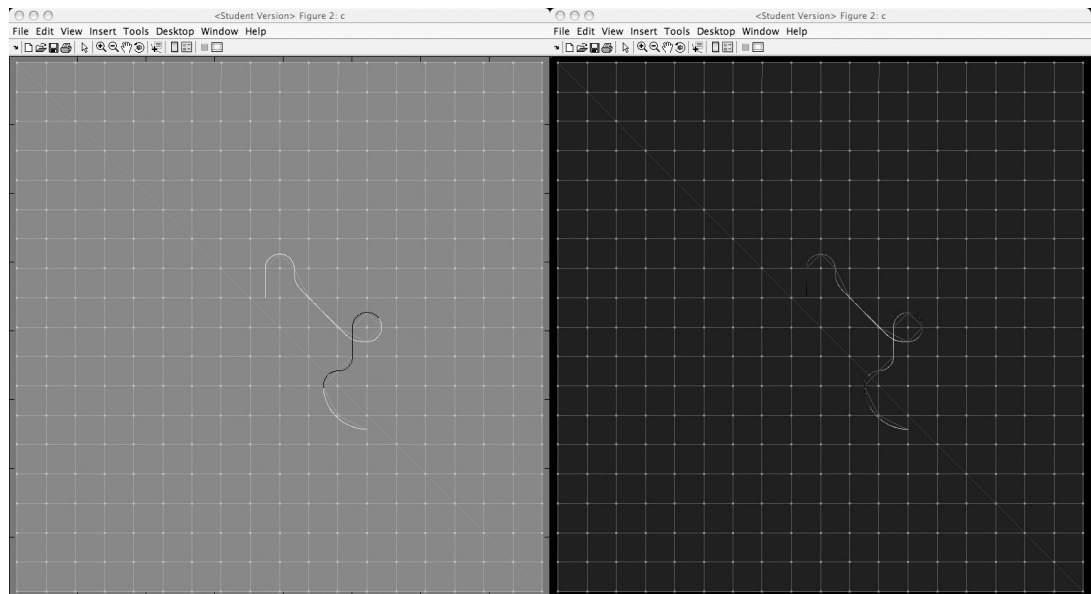


Figure 71: Interlace solution (left) distance weights (right)

In the case of irregular grids and freeform curve specifications, the moves fail to tell us where the curves are intended to go. To solve these curves we need to take the goal positions from the study curve design. In the construction of the genotype for such a curve, we use the bridge operators and use the goal positions as evolution objectives. The bridge operators evolve genetic descriptions scored by how well they accomplish the interpolation. When the interpolation is close enough (user defined error is small enough) the generating sequences for these curves is saved as the description for how

they are designed. Following solution guide curves generated in this way are handled the same as those by the template method.

5.5 Form solving and posing

So, we finally come to the part where we want to pose a Celtic snake. We have the snake loaded. We have designed and generated a guide curve. If we look on the design panel we notice that the snake isn't selected:

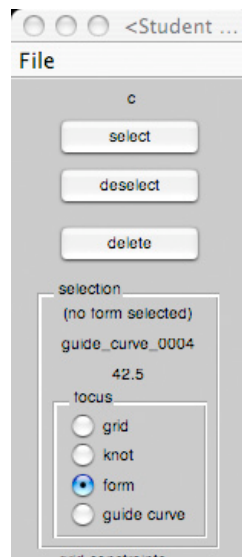


Figure 72: Form selection in design panel

With the focus set to form and press select we can pull up our snake through these automatic dialogs:

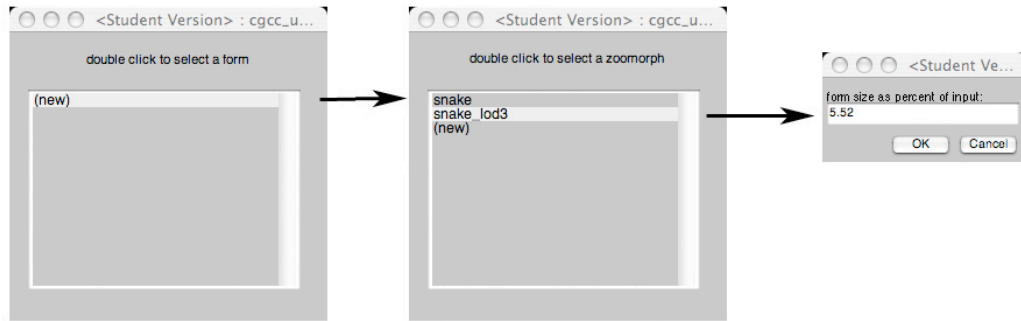


Figure 73: For selection dialogs

With our snake picked from the table of prototypes and its scale set, the active form is instantiated as an individual and it receives a copy of the chromosome from its prototype. The user interface sets the center of the design display as target transforms (display dimension, pose position, and pose rotation from rest), as well, as the scale, as values of variables in the local scope of the individual. Since these are referenced by sequences of the chromosome, they will direct the size and placement of the phenotype of the assembled form. The form is then attached to ribosome and the ribosome is asked to translate the following sequences: `assemble`, `form_from_rest`, `bind`, and `render`. Assembly instantiates the default skeleton consisting of three guide curves in the local scope of the individual. An anatomy object is instantiated to manage the solution and binding for each of the strokes used to draw the form. Anatomy objects are instances of Matlab classes. An example from the snake as follows:

example of an anatomy object:

attribute and example data	description
name: snake_lod3_tail_re_width_right	
class: snake_lod3	
symmetry: 2.000000	defined on main symmetry
feature: tail_re	
side: right	
binding: guide_curve_0004	
scale: 0.055200	
symmetry_s0: 0.000000	legacy--not used
symmetry_s1: 0.000000	legacy--not used
s0: 1070.000000	start of defined interval
s1: 1343.000000	end of defined interval
s: [55x1 double]	arc length samples array
width: [55x1 double]	width as function of arc length, s
boundary: [55x1 double]	boundary label as function of s
segs: 931.000000	segment breaks locations array
s_s0: 59.064000	scaled start of defined interval
s_s1: 575.661011	scaled end of defined interval
s_s: [55x1 double]	scaled arc length samples array
s_width: [55x1 double]	scale width array
s_boundary: [55x1 double]	boundary labels array
s_segs: 51.391200	scaled segment breaks locations array
X: [55x3 double]	position solution
B: [55x1 double]	boundary label
Wt: [55x1 double]	weight solution--distance from cvs
AB: [55x1 double]	interlace solution--above/below
S: [55x1 double]	arc length samples
visible: true	visibility state

Table 16: Anatomy object for snake stroke

Continuing with the sequences, the `form_from_rest` sequence applies the transform that positions the skeleton of the zoomorph in the display. The `bind` sequence assigns the name of the respective default guide curve for each of the strokes in the anatomy. This causes each stroke to determine its position along its guide curve. If the stroke is split by segmentation, and the length of the guide curve is longer than the default, the stroke will just its length to fit the region of the guide curve over which it is intended to be drawn. Where this happens the segment will be stretched by replicating the width value at the segment boundary to fill the length of the stretch at the user defined sample frequency. Although this creates a constant width throughout the

stretch, with more work the anatomy could be varied over the stretch, as well. The last step in the binding process is to solve the curve. This is done by interpolating the solution of the guide curve for each width sample in a stroke. Interpolation is done using the Matlab function, `pchip`, which performs the interpolation by means of a piecewise cubic Hermite polynomial. For each width sample in a stroke we now have the solution to the normal of the guide curve. We apply the parallel curve theorem from differential geometry to position each point of each stroke at its scaled sampled distance from the guide curve along its normal. When this is finished, the render sequence performs a draft render of the form for display. All of this data becomes part of the phenotype of the snake. In the case of our example, the display is redrawn as follows:

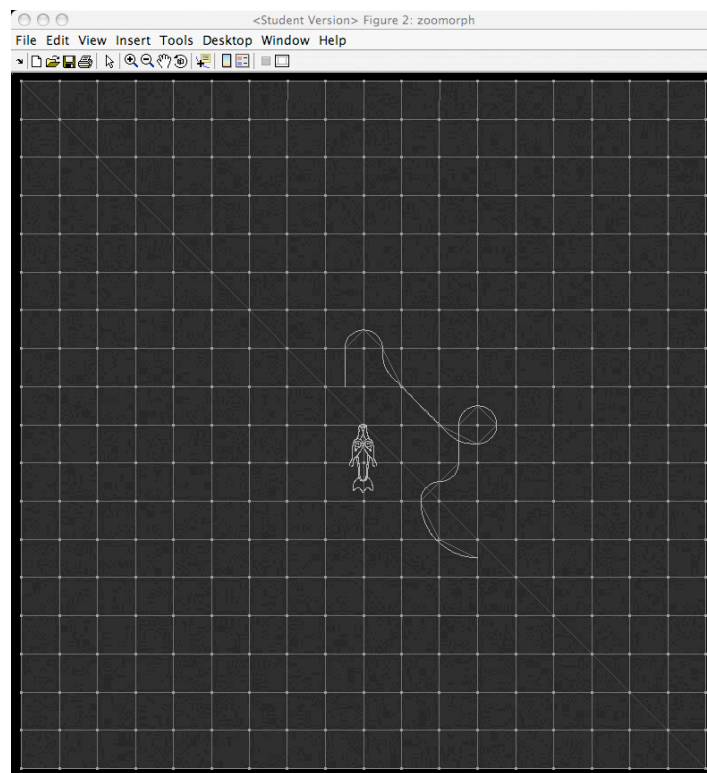


Figure 74: Phenotype of form in default display

The form and the guide curve now appear as the current selections. All we need to do is bind them in order to repose the snake. The user directs this by pressing the

bind button on the design panel. The following prompt appears to allow the user to select the symmetry of the binding:

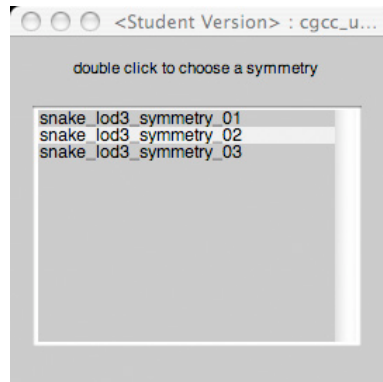


Figure 75: Form symmetry binding dialog

The procedure for this binding is almost the same as the one we have just described. The first thing done does is to attach the form to a ribosome, translate its `form_to_rest` sequence, and release the form. We calculate a translation that places the origin of the guide curve at the origin of the chosen symmetry in its rest position. We also calculate a rotation of the guide curve so that its initial tangent is the same as that of the chosen symmetry in the rest position. These transforms are assigned to variables in the global scope of the composition. The `form_to_rest` sequence for the chosen symmetry is copied and rebound as a `guide_to_rest` sequence in the form. The form is again attached to a ribosome, the `guide_to_rest` sequence translated, and the form released. The result is the new guide curve being in position to replace the chosen symmetry in the rest position. The user interface then changes the name of the binding for strokes on the symmetry. This is an assignment in a sequence named, `rebind`. Since the main symmetry of the form was chosen for rebinding, the inverse of the `guide_to_rest` transform is assigned to variables in the local scope of the snake. The

snake is again attached to a ribosome and the following sequences are translated: rebind, form_from_rest, bind, and render. The snake is released and the display is updated:

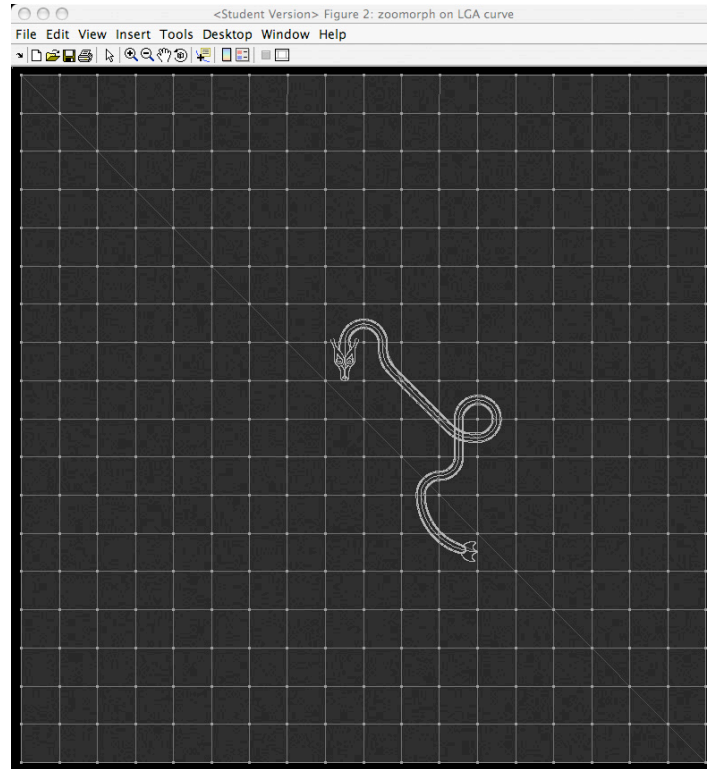


Figure 76: Celtic snake in new pose

As seen with some earlier examples, the design panel allows us to select edges in the design grid and place and delete break marks to influence designs in that way as well.

5.6 Rendering

Rendering of the draft design display is handled by a compositor. At first this function was very simple. As the system evolved and requirements were resolved, this function came into its own as a subsystem. Draft design displays are made up of many layers. Which ones are included and at what intensity depends on the state of the design

at the time the compositor is called. For example, if the design contains more than one form the compositor will gather and composite the draft renderings so that all of them are displayed. Then it became clear that sometimes the need is different and the series of options were developed. Control of display and render options are now grouped together in what some commercial software packages call the render globals. This is called up as a separate panel from the hub:

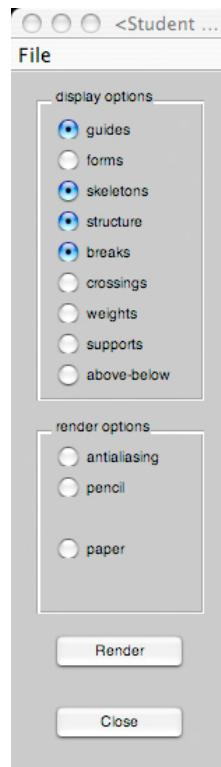


Figure 77: Render panel

Each button in the upper grouping controls the value of a global variable that influences the behavior the compositor. We have seen examples of the above and below relationships and the weights. The options on the bottom part this panel control values of global variables for final rendering.

Final rendering is a two pass process involving a label buffer and a z-buffer. On the first pass those portions of strokes in guide curve intervals in the *above* state are rendered. Their boundary labels are recorded in a label buffer. The second pass involves a turtle (four-state machine) that walks all of the strokes and renders those portions in guide curve intervals that are in the *below* state. When lines rendered in the first pass are encountered, the boundary buffer is consulted and external boundary labels are counted. When the count is even, strokes in the below state are visible. When the count is odd, they are occluded. Final renders are displayed in a separate window from the draft display. A final render without anti-aliasing for our example appears as follows:

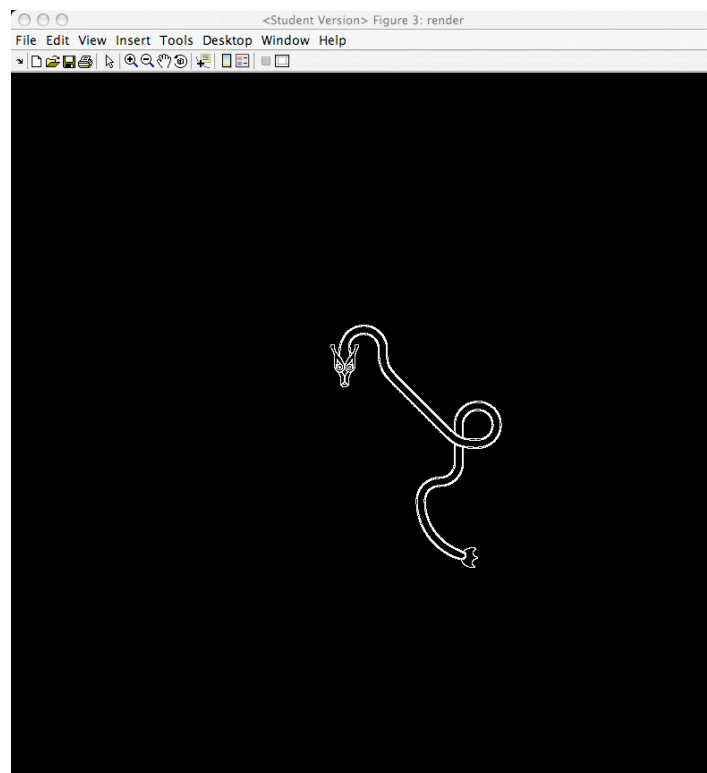


Figure 78: Final rendering

5.7 Summary

In this chapter we have reviewed design details of the demonstration the system. We have studied the design of experimental grids. We have examined forms and sleeves. We have followed the procedures involved in constructing genotypes of forms and guide curves. We have followed user interactions through the process of designing guide curves and generating their phenotypes. We have looked at the generation of the phenotype for a form and the details involved with the experiments on posing a form using a guide curve. Finally, we have presented a labeled z-buffer approach that allows us to render the results.

Chapter 6 Results and Conclusions

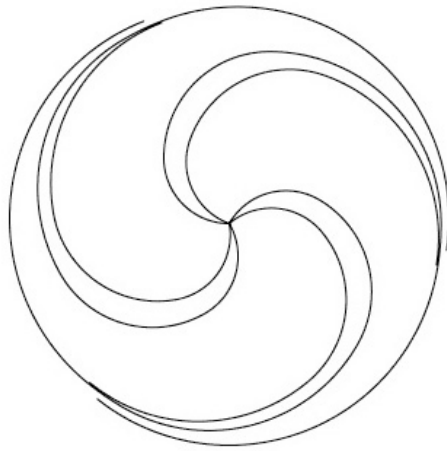


Figure 79: Celtic Spiral

6.1 Introduction

This chapter describes a series of experiments and outcomes to test the ideas developed during this investigation. They are tested against examples of knotwork, plant, and animal designs from the book of Kells, the Lindisfarne gospels, and drawings from the studies of George Bain. The primary references for comparison with hand-drawn examples

will be found in the M.Phil thesis and Ph.D. thesis of Gwenda Adcock, the illustrations of George Bain, and the early works of Aiden Meehan.

6.2 Celtic spiral capability demonstration

The figure the opening this chapter is a Celtic triple spiral. It was determined that one of the first things the methods developed in this thesis should be able to accomplish were Celtic spirals. This example was motivated by the following spiral that appears on folio 34r of the Book of Kells:



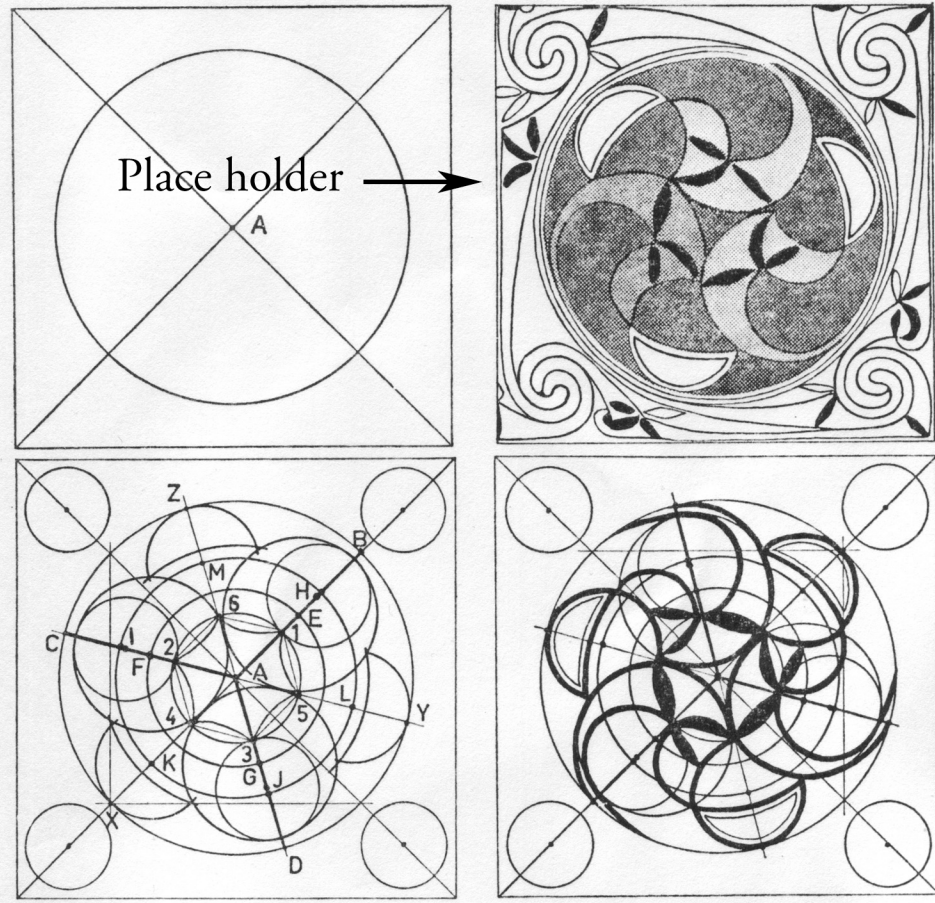
Figure 80: Triple spiral MS 58 f 34r (Board of Trinity College Dublin)

The example represents a complete coding of a generating method in terms of the linear genetic programming model.

6.3 Lindisfarne Gospels spiral

A much more complex example comes from analysis and reconstruction by Bruce-Mitford from the Lindisfarne gospels. His analysis involves considerable construction work. The purpose of this experiment was to determine the extent to which effort can be reduced

by observations on where curves start and terminate. This tile extends in four quadrants to fill a panel in its folio. Only the design in the center was examined here.



(still in progress)

Figure 81: Lindisfarne spiral studied by Bruce-Mitford (Alexander, 1978, p. 11)

6.4 Knotwork

Although knotwork is not part of the lozenge in folio 34r, it is of fundamental part of most Celtic design. The first valid test of the two pass production render engine was made on a design from the Lindisfarne Gospels folio 11 drawn by Iain Bain (1992, p. 73):

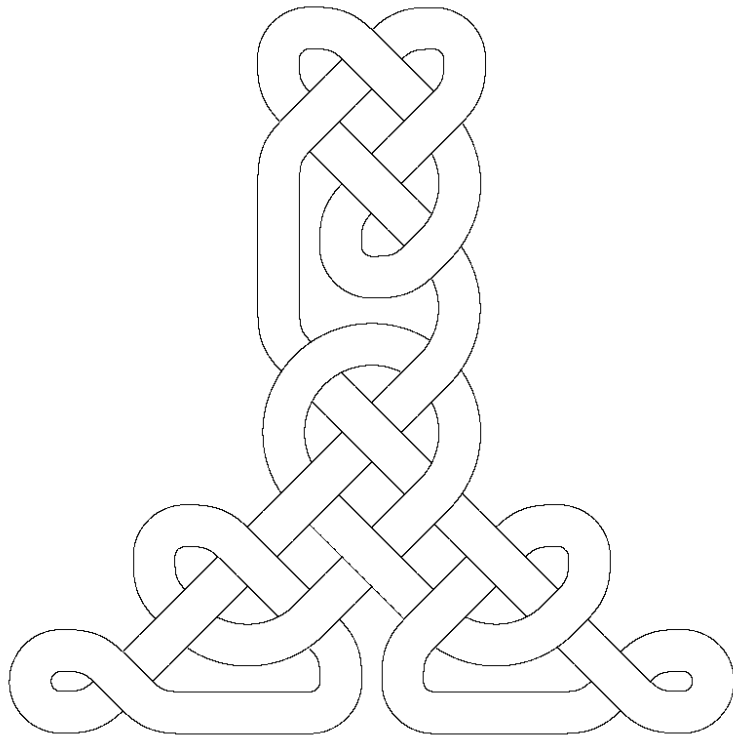


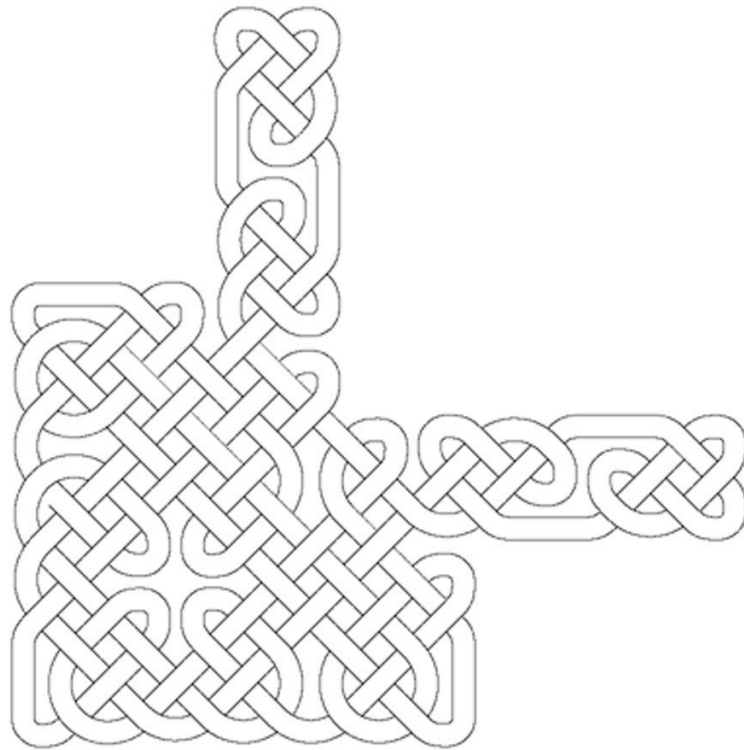
Figure 82: Lindisfarne knotwork folio 11 (Bain, 1992)

(waiting for artifact render)

Figure 83: Lindisfarne knotwork folio 11 (Bain, 1992)

6.5 Lindisfarne knotwork (Adcock)

A more complex example of knotwork is from Gwenda Adcock's Ph.D. thesis illustrations. This, border fragment, shown earlier, consists of 3 linking knots:



(waiting for artifact render)

Figure 84: Lindisfarne folio 26 border (Adcock, 2002)

6.6 Lozenge of MS 58 f 34r

One of the primary motivators of this investigation is the lozenge in cross of the Chi in folio 34r that was first observed in George Bain's studies of modern methods of construction (1951, p. 114):



By his count the lozenge contains four hairy male humans, four reptiles, four beasts and twelve birds. In this section as an experiment in capability one of each of these forms was modeled.

6.3.1 Reptile (snake)

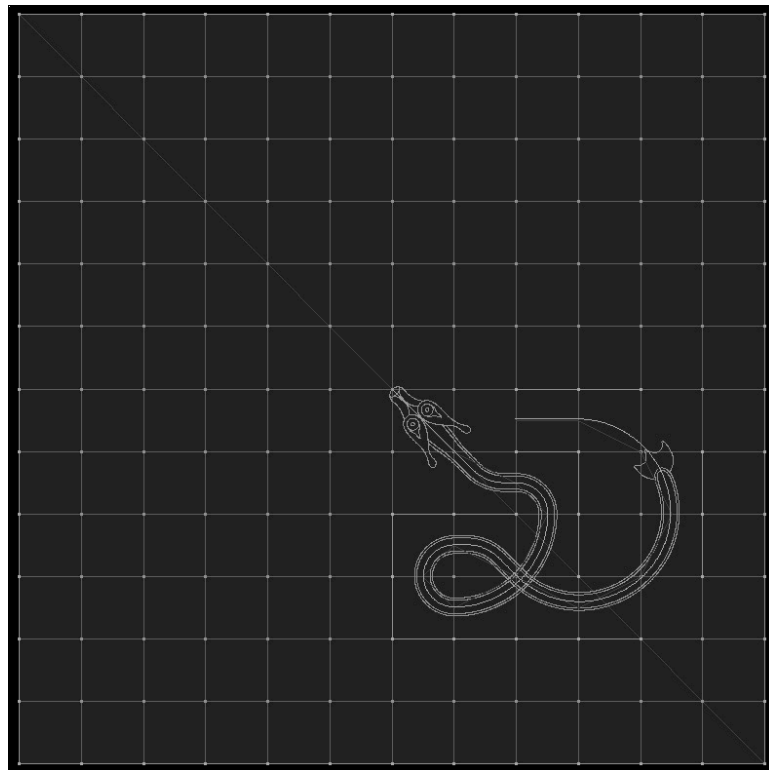
The philosophy of forms is that they are prepared as line drawings and produced by what ever means are convenient up to and including pencil drawings. In the case of the reptile the figure represents a composite of the various snakes one finds in the pages of the Book of Kells. It was assembled by tracing and reshaped as a line drawing in lightwave. The figure was then captured by screen captures. The anatomy, symmetry, segmentation, and pivot files were composed in photoshop and placed in the project folder.

(figure)

Figure 85: Design files for the reptile

These were used for the user interaction examples presented in Chapter 5.

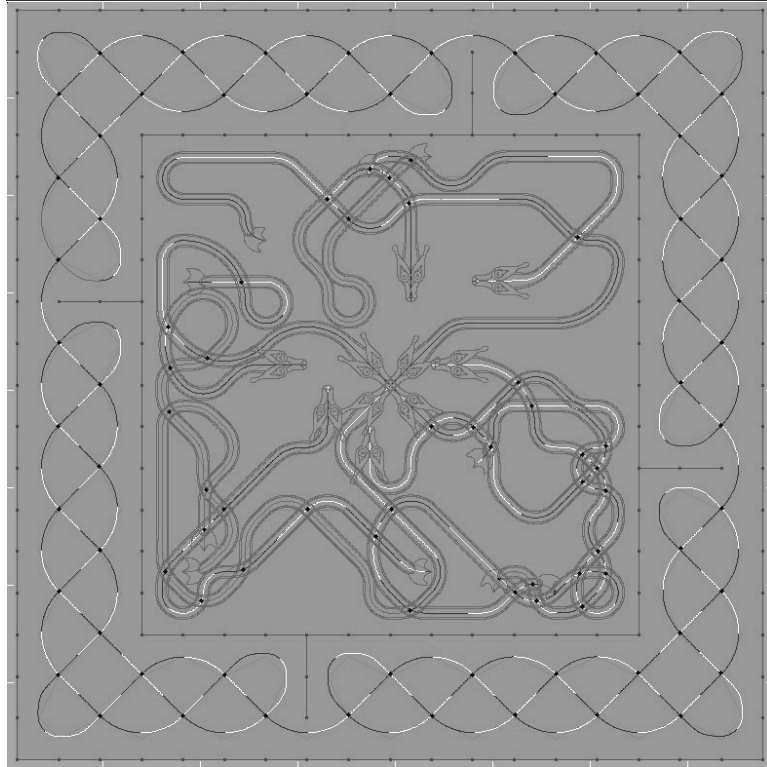
Rendering of a posed reptiles is shown in the following figure:



(waiting render of alternate pose and artifacts)

Figure 86: Rendering of a Posed reptile

A significant portion of the lozenge in folio 34r is devoted to a complex tangle of creatures. A series of reptiles were assembled as a test of the functions that solve tangle interlace. An example of this as follows:



(waiting for artifact render—and correction of an error)

Figure 87: Tangle of reptiles

6.3.2 Hairy male human

The hairy male is an assembly compiled from the drawing of George Bain:

(figure)

Figure 88: Design files for a hairy human

An example rendering of the posed form is shown below:

(figure)

Figure 89: Rendering of hairy male

6.3.3 Beast

The beast is also an assembly compiled from the drawing of George Bain:

(figure)

Figure 90: Design files for a beast

An example rendering of the posed form is shown below:

(figure)

Figure 91: Rendering of a beast

6.3.4 Bird

The bird is an assembly compiled from the Lindisfarne gospels. Well preserved examples of these are hard to find. Some of the best examples are contained in the pages of Lindisfarne gospels. While this bird may not be the same as that in the book of Kells, it provides us with an example:

(figure)

Figure 92: Design files for a bird

An example rendering of the posed form is shown below:

(figure)

Figure 93: Rendering of a bird

(Reserved for one more thing if ...)

6.7 Plant

Celtic design also includes plants. The following is an example from the book of Kells. It is derive from an illustration by George Bain (1951, p. 123):

(figure)

Figure 94: Design files for a plant

An example rendering of the posed form is shown below:

(figure)

Figure 95: Rendering of a plant

6.8 Summary

This section has presented a variety of knotwork and animal design examples that include hand drawn artifacts. Very little research is available regarding the aesthetic

characteristics of details both find and grand in such work. We leave the reader to contemplate on the effectiveness of the methods demonstrated here. The hope would be that this will stimulate continued interest in discovering ways to extend both the capability and the participation of complex systems methods as they applied to highly structured tasks.

Of great interest is the computational neuroscience modeling of major systems used in the human cognitive system and their relationship to skills only vaguely understood at this point. Research in interactive recognizers capable of drawing what they intend their recognition experience to be, promises deep revelations about the ability of machines to learn and participate in more cognitively challenging roles than they enjoy at present.

6.9 Triple spiral example code

```
% golden ratio spiral experiment
% spiral_design_triple_A

% clear the workspace

close(findobj('Type','figure'));
drawnow;

clear classes

% initialize

jitter;

rad2deg = 180/pi;
deg2rad = pi/180;

% Figure title and file name

figure_title = 'spiral experiment (triple)';
figure_file = figure_title;

str_ndx = strfind(figure_title, ' ');

for i=1:length(str_ndx)
    figure_file(str_ndx) = '_';
end

% define a canvas

im_canvas = ones(512,512);
[im_rows, im_cols] = size(im_canvas);

% global construction parameters
```

```

bias = -60;
offset = 20;
step = 120;
initial_radius = 40;
sections = 4;
golden_ratio = (1 + sqrt(5)) / 2;
canvas_center = [ im_cols/2 im_rows/2 0 ]';

% initial figure window

im2_max = max(im_canvas(:));
im2 = uint8(imFromRange(im_canvas, 0, im2_max, 0, 255));
fig_h = figure('Units','pixels', 'Name', figure_title, 'Position', [0 40 im_cols
im_rows ]);
img_h = image(im2rgb(im2,im2,im2));
set(gca,'Position', [0,0,1,1]);

% DNA and RNAs

DNA = chromosome;
rRNA = ribosome;
mRNA = messenger;

% instantiate a phenotype

sequence = { ...
    { 'phenotype' 'phenotype_comp' 'myComp' } ...
    { 'rez' 2 } ...
    { 'antialias' 'off' } ...
    { 'jitter' j15 } ...
};

DNA = addSequence(DNA, 'phenotype', cells2kpmStack(sequence));

% center of the WCS and center of the canvas

sequence = { ...
    { 'new' 'point' 'O' } ...
    { 'translation' 'O' [0 0 0]' } ...
    { 'new' 'point' 'C' } ...
    { 'translation' 'C' canvas_center } ...
};

DNA = addSequence(DNA, 'wcs', cells2kpmStack(sequence));

% intrinsic curve constructors

sequence = { ...
    { 'new' 'intrinsicCurve' 'S1a' } ...
    { 'rotation' 'S1a' 'O' (0*step + 0*offset + bias) } ...
    { 'translation' 'S1a' canvas_center } ...
    { '=' 'k' sections } ...
    { '=' 'r' (initial_radius / golden_ratio) } ...
    { '=' 'r' { '*' golden_ratio 'r' } } ...
    { '=' 'kappa' { '/' 1 'r' } } ...
    { 'curvature' 'S1a' 'kappa' } ...
    { 'extend' 'S1a' { '*' pi/2 'r' } } ...
    { '=' 'k' { '-' 'k' 1 } } ...
    { 'if' { '>' 'k' 0 } { 'jump' -6 } } ...
    { 'solve' 'S1a' } ...
    { 'visible' 'S1a' } ...
};

DNA = addSequence(DNA, 'S1a', cells2kpmStack(sequence));

sequence = { ...
    { 'new' 'intrinsicCurve' 'S1b' } ...

```

```

    { 'rotation'      'S1b' '0' (0*step + 1*offset + bias) } ...
    { 'translation'  'S1b' canvas_center } ...
    { '=' 'k'        sections } ...
    { '=' 'r'        (initial_radius / golden_ratio) } ...
    { '=' 'r'        { '*' golden_ratio 'r' } } ...
    { '=' 'kappa'    { '/' 1 'r' } } ...
    { 'curvature'   'S1b' 'kappa' } ...
    { 'extend'      'S1b' { '*' pi/2 'r' } } ...
    { '=' 'k' { '-' 'k' 1 } } ...
    { 'if' { '>' 'k' 0 } { 'jump' -6 } } ...
    { 'solve'      'S1b' } ...
    { 'visible'    'S1b' } ...
};

DNA = addSequence(DNA, 'S1b', cells2kpmStack(sequence));

sequence = { ...
  { 'new' 'intrinsicCurve' 'S2a' } ...
  { 'rotation'      'S2a' '0' (1*step + 0*offset + bias) } ...
  { 'translation'  'S2a' canvas_center } ...
  { '=' 'k'        sections } ...
  { '=' 'r'        (initial_radius / golden_ratio) } ...
  { '=' 'r'        { '*' golden_ratio 'r' } } ...
  { '=' 'kappa'    { '/' 1 'r' } } ...
  { 'curvature'   'S2a' 'kappa' } ...
  { 'extend'      'S2a' { '*' pi/2 'r' } } ...
  { '=' 'k' { '-' 'k' 1 } } ...
  { 'if' { '>' 'k' 0 } { 'jump' -6 } } ...
  { 'solve'      'S2a' } ...
  { 'visible'    'S2a' } ...
};

DNA = addSequence(DNA, 'S2a', cells2kpmStack(sequence));

sequence = { ...
  { 'new' 'intrinsicCurve' 'S2b' } ...
  { 'rotation'      'S2b' '0' (1*step + 1*offset + bias) } ...
  { 'translation'  'S2b' canvas_center } ...
  { '=' 'k'        sections } ...
  { '=' 'r'        (initial_radius / golden_ratio) } ...
  { '=' 'r'        { '*' golden_ratio 'r' } } ...
  { '=' 'kappa'    { '/' 1 'r' } } ...
  { 'curvature'   'S2b' 'kappa' } ...
  { 'extend'      'S2b' { '*' pi/2 'r' } } ...
  { '=' 'k' { '-' 'k' 1 } } ...
  { 'if' { '>' 'k' 0 } { 'jump' -6 } } ...
  { 'solve'      'S2b' } ...
  { 'visible'    'S2b' } ...
};

DNA = addSequence(DNA, 'S2b', cells2kpmStack(sequence));

sequence = { ...
  { 'new' 'intrinsicCurve' 'S3a' } ...
  { 'rotation'      'S3a' '0' (2*step + 0*offset + bias) } ...
  { 'translation'  'S3a' canvas_center } ...
  { '=' 'k'        sections } ...
  { '=' 'r'        (initial_radius / golden_ratio) } ...
  { '=' 'r'        { '*' golden_ratio 'r' } } ...
  { '=' 'kappa'    { '/' 1 'r' } } ...
  { 'curvature'   'S3a' 'kappa' } ...
  { 'extend'      'S3a' { '*' pi/2 'r' } } ...
  { '=' 'k' { '-' 'k' 1 } } ...
  { 'if' { '>' 'k' 0 } { 'jump' -6 } } ...
  { 'solve'      'S3a' } ...
  { 'visible'    'S3a' } ...
};

DNA = addSequence(DNA, 'S3a', cells2kpmStack(sequence));

```

```

sequence = { ...
    { 'new' 'intrinsicCurve' 'S3b' } ...
    { 'rotation' 'S3b' '0' (2*step + 1*offset + bias) } ...
    { 'translation' 'S3b' canvas_center } ...
    { '=' 'k' sections } ...
    { '=' 'r' (initial_radius / golden_ratio) } ...
    { '=' 'r' { '*' golden_ratio 'r' } } ...
    { '=' 'kappa' { '/' 1 'r' } } ...
    { 'curvature' 'S3b' 'kappa' } ...
    { 'extend' 'S3b' { '*' pi/2 'r' } } ...
    { '=' 'k' { '-' 'k' 1 } } ...
    { 'if' { '>' 'k' 0 } { 'jump' -6 } } ...
    { 'solve' 'S3b' } ...
    { 'visible' 'S3b' } ...
};

DNA = addSequence(DNA, 'S3b', cells2kpmStack(sequence));

% rendering

sequence = { ...
    { 'render' fig_h { 'Overlay', 'on', 'LineWidth', 0.5, 'Color', 'k' } } ...
};

DNA = addSequence(DNA, 'render', cells2kpmStack(sequence));

% analysis of the resulting figure

sequence = { ...
    { '=' 'active_rows' 256 } ...
    { '=' 'active_cols' 256 } ...
    { '=' 'tweak' 5 } ...
    { '=' 'im_S1a_X' { 'image_of' 'S1a' 'position' } } ...
    { '=' 'im_S1a_Z' { 'image_of' 'S1a' 'state' } } ...
    { '=' 'im_S1b_X' { 'image_of' 'S1b' 'position' } } ...
    { '=' 'im_S1b_Z' { 'image_of' 'S1b' 'state' } } ...
    { '=' 'im_conj' { '&' 'im_S1a_X' 'im_S1b_X' } } ...
    { '=' 'im_conj' { '*' { 'R' 'im_conj' } { '~' { 'padarray' { 'square' [10
10] [ 502/2 502/2 ] } } } } ...
    { '=' 'im_D1' { 'magnitude' { '-' 'im_S1a_X' 'im_S1b_X' } } } ...
    { '=' 'im_I' { 'mask_min' 'im_conj' 'im_D1' } } ...
    { '=' 'im_T1_ka' { '*' 'im_I' 'im_S1a_Z' } } ...
    { '=' 'im_T1_kb' { '*' 'im_I' 'im_S1b_Z' } } ...
    { '=' 'I_S1a_k' { 'find_values' 'im_T1_ka' } } ...
    { '=' 'Sa_r' { 'get_arclength_k' 'S1a' 'I_S1a_k' } } ...
    { '=' 'I_S1b_k' { 'find_values' 'im_T1_kb' } } ...
    { '=' 'Sb_r' { 'arclength_at_k' 'S1b' 'I_S1b_k' } } ...
    { '=' 'rb' { 'magnitude' { '-' { 'position_at_k' 'S1b' 'I_S1b_k' } {
'position_at_k' 'C' 0 } } } } ...
    { '=' 'rb' { '+' 'rb' 'tweak' } } ...
    { '%' } ...
    { 'antialias' 'on' } ...
};

DNA = addSequence(DNA, 'analysis', cells2kpmStack(sequence));

% edit the figure

sequence = { ...
    { 'arclength' 'S1a' 'Sa_r' } ...
    { 'curvature' 'S1a' { '/' 1 'rb' } } ...
    { 'extend' 'S1a' { '*' (2*pi/3) 'rb' } } ...
    { 'solve' 'S1a' } ...
    { 'arclength' 'S1b' 'Sb_r' } ...
    { 'solve' 'S1b' } ...
    { 'arclength' 'S2a' 'Sa_r' } ...
    { 'curvature' 'S2a' { '/' 1 'rb' } } ...
    { 'extend' 'S2a' { '*' (2*pi/3) 'rb' } } ...
};

```

```

    { 'solve'      'S2a'      } ...
    { 'arclength' 'S2b' 'Sb_r' } ...
    { 'solve'      'S2b'      } ...
    { 'arclength' 'S3a' 'Sa_r' } ...
    { 'curvature'  'S3a' { '/' 1 'rb' } } ...
    { 'extend'     'S3a' { '*' (2*pi/3) 'rb' } } ...
    { 'solve'      'S3a'      } ...
    { 'arclength' 'S3b' 'Sb_r' } ...
    { 'solve'      'S3b'      } ...
};

DNA = addSequence(DNA, 'edit', cells2kpmStack(sequence));

% inject a copy of the DNA into the phenotype

mRNA = setLabel(mRNA, 'phenotype');
mRNA = findSequence(mRNA, DNA);
rRNA = translate(rRNA, mRNA);

myComp = get(rRNA, 'phenotype');
myComp = set(myComp, 'dna', DNA);
rRNA = set(rRNA, 'phenotype', myComp);

% biological analog functional relationships

schedule = {'wcs' 'S1a' 'S1b' 'S2a' 'S2b' 'S3a' 'S3b' 'analysis' 'edit' 'render'
};

for i=1:length(schedule)

    sequence_label = schedule{i};

    fprintf(' translating sequence %s\n',sequence_label);

    mRNA = setLabel(mRNA, sequence_label);
    mRNA = findSequence(mRNA, DNA);
    rRNA = translate(rRNA, mRNA);

end

hold off;

```

Place holder for chapter formatting (do not delete below here)

Chapter 7 Suggestions for Further Research

7.1 Introduction

This thesis has described an investigation into the application of complex systems methods in Celtic design. Specific problems observed when comparing insular manuscripts from the 9th century with contemporary Computer generated work revealed limitations involved with the process that artists refer to as mark-making. A wide interdisciplinary review of the prodigious body of existing literature relating to the design and construction of illuminated manuscripts was conducted. Of the modern methods of Celtic design explored to this point, it was found that very little had been done with complex systems theory. Matches were found between activities in the design work flow and methods in complex systems theory. In this section we will summarize the actual capabilities of the system that resulted from this study and examine questions that are presented for further study.

7.2 Summary of system capabilities

The study in chapter two revealed that a computer generated Celtic design system will often need to support the following capabilities:

1. To work within panels of a larger design

As shown in Chapter 5 the system has the ability to load either the drawing space of a complete design or allow the designer to work with in the panels of a much larger design. This was facilitated by means of morphological analysis that assisted with organizing the space in preparation for design work. This included the location of boundaries and corners and the construction of a geometric complex to support the definition of structure.

2. The process involves the development of structure in the form of a grid

Also shown in chapter five was the ability to assist the designer in constructing a hierarchical grid based on templates.

3. The drawing space can be irregular

The morphological analysis technique has the ability to extend to complex drawing spaces that includes multiple forms and voids. Although not fully implemented during this investigation, the template method for assembling grids offers the ability to accommodating design spaces of arbitrary shape.

4. Design elements are comprised of forms both of knots of tangles

We have demonstrated the ability to accommodate forms as found in Celtic animal and plant designs. A Method based on the sampling of line drawings was developed to

describe the anatomy of forms. This method reconstructs these shapes as parallel curves to a guide curve. The method was shown to work for both complex forms as well as cords. The notion was extended for cords to the idea of a sleeve allowing the modular description of shape and the integration of flat regions with chevrons. Through the use of unit speed space curves with assigned curvature we demonstrated the ability to represent complex knots and the open curves of tangles. The lattice gas automata was introduced as a means of discovering the complexity of design possible within the structure of grids. We demonstrated a modular way of describing the specifications for these guide curves through the use of a linear genetic programming model.

5. Design elements form alternating interlace with one another

The requirements for solving alternating interlace between knots and tangles were examined. The constraints by which knots will always have alternating solutions in Celtic design were determined. It was also determined that tangles do not always have alternating solutions. Modest extensions to the theory were presented to provide guidance to designers as to when interlace involving tangles have alternating solutions to crossing assignments and when they don't. We also presented lattice gas automata in the solution of alternating crossings associated with the method devised by Christian Mercat.

6. To approach the quality observed in the insular manuscripts of the 9th century, the rendering of results should express the care in mark making associated with lines drawn by the human hand

This thesis is demonstrated the use of noise processes in the curvature of unit space curves to introduce symmetric imperfections. The method similarly applied to the anatomy of forms and cords demonstrates the ability to introduce the asymmetries also apparent in the line drawing. The use of physics based models of writing instruments begins the process of introducing the way these elements appear in physical media.

7.3 Further work

This thesis has pioneered the use of complex systems methods in Computer generated Celtic design. Because of its early nature, the main object of the research was to demonstrate the feasibility this approach to some of the questions involved.

Notably missing are the fitness functions normally associated with evolutionary processes. Measuring of fractal dimension has been constructively used in the authentication of line drawings by Pollock. The salience map associated with Laurent Itti's work implementating the primate vision system has been observed to require further development. Promising along its line of inquiry, though is the potential for computational neurobiological modeling of human cognitive performance to help us better understand skilled human performance and the structure beneath it.

Regarding the specifics of rendering on animal skin vellum, physical modeling for this is needed similar to the research already carried out for paper and canvas.

There is reason to believe that the lozenge in the heart of the Chi in folio 34r was hand drawn. The study presented here on using Perlin noise to facilitate spontaneous

expression suggests that further investigation of the dynamics in nature may provide insight into how patterns evolve and why they are recognized.

Further research is needed into the understanding freeform drawing. The thesis of Jowers on freeform shape grammars suggests the beginnings of this. The combination of neural cognitive modeling and understanding of the patterns and rhythms involved with creative reasoning reaches back into Yevin's notion that the Criticality of the art may be associated with a Criticality in the mind. Santella's work in the eye motion tracking reveals at least one way in which we can observe the complex system that is the human mind. Further research along that direction may again lead us to a better understanding of what we can measure and quantify about skilled performance and then what we might do to model it.

The questions are vague and unsettling. The prospect of one day understanding them is intensely exciting.

7.4 Summary

(This may not be needed.)

Place holder for chapter formatting

References

- Abbot, Steve (1998). *Steve Abbot's computer drawn Celtic knotwork*, Knots3D, retrieved on October 29, 2005, from <http://www.abbott.demon.co.uk/knots.html>.
- Abt, Jeffrey (1989). A Byzantine Scholar's Letter on the Preparation of Manuscript Vellum, *Journal of the American Institute for Conservation*, Vol. 28, No. 2.
- Adcock, Gwenda (1974). *A Study of the Types of Interlace on Northumbrian Sculpture*, M. Ph. thesis, Durham University, United Kingdom.
- Adcock, Gwenda (2002). *Interlaced Animal Design in Bernician Stone Sculpture Examined in the Light of Design Concepts in the Lindisfarne Gospels*, Ph.D dissertation, University of Durham, United Kingdom.
- Alexander, J. J. G. (ed.) (1978). *Insular Manuscripts, 6th to the 9th Century*, Harvey Miller, London, pp. 9-18, 72-75.
- Alexander, Howard (1975). The computer/plotter and the 17 ornamental design types, *Proceedings of SIGGRAPH '75*, pp 160-167.

- Alexander, Jonathan (1992). *Medieval Illuminators and Their Methods of Work*, Yale University Press.
- Allen, J. Romilly and Joseph Anderson (1903). *The Early Christian Monuments of Scotland*, the Society of Antiquaries of Scotland.
- Augusteijn, Marikje, and B. A. Folkert (2002). Neural network classification and novelty detection, *International Journal of Remote Sensing*, Vol. 23, No. 14, pp. 2891-2902.
- Bain, George (1951). *Celtic Art: The Methods of Construction*, William Maclellan & Co. Ltd., Glasgow, republication by Dover, New York, 1973.
- Bain, Iain (1992). *Celtic Knotwork*, Sterling Publishing Company.
- Baker, Amy (2005). *Common Medieval Pigments*, retrieved on September 3, 2007, from <http://www.ischool.utexas.edu/~cochine/pdfs/a-baker-04-pigments.pdf>.
- Barker, Kenneth (ed.) (1985). *The New International Version Study Bible*, Zondervan Publishing House.
- Barrow, Fr. Gabriel (1975). The Iconoclastic Crisis, *Word Magazine*, February, p. 11-12, retrieved on April 9, 2008, from http://www.orthodoxresearchinstitute.org/articles/church_history/barrow_iconoclastic_crisis.htm

- Baxter, William (2004). *Physically-Based Modeling Techniques for Interactive Digital Painting*, Ph.D dissertation, University of North Carolina at Chapel Hill.
- Beach, Richard, and Maureen Stone (1983). Graphical style towards high quality illustrations, *ACM SIGGRAPH Computer Graphics*, Vol. 17, Issue 3, pp. 127-135.
- Bentley, Peter J. (1996). *Generic Evolutionary Design of Solid Objects using a Genetic Algorithm*, Ph.D dissertation, The University of Huddersfield.
- Bentley, Peter J. and Una-May O'Reilly (2001). Ten Steps to Make a Perfect Creative Evolutionary Design System, *GECCO 2001 Workshop on Non-Routine Design with Evolutionary Systems*.
- Blade, Richard (1968). *Fundamental Physics*, University of Colorado.
- Bonabeau, Eric, Marco Dorigo, and Guy Theraulz (1999). *Swarm Intelligence: from Natural to Artificial Systems*, Santa Fe Institute, Oxford University Press.
- Brameier, Markus (2004). *On linear Genetic Programming*, Ph.D. thesis, University of Dortmund.
- Brisco, Adriana, and Lars Chittka (2001) The Evolution of Color Vision in Insects, *Annual Review of Entomology*, 46:471-510.

- Brown, Katherine, and Robin Clark (2004a). Analysis of key Anglo-Saxon manuscripts (8-11th centuries) in the British Library: pigment identification by Raman spectroscopy, *Journal of Raman Spectroscopy*, 35, pp. 181-189.
- Brown, Katherine, and Robin Clark (2004b). The Lindisfarne gospels and two other 8th century Anglo-Saxon/Insular manuscripts: pigment identification by Raman spectroscopy, *Journal of Raman Spectroscopy*, 35, pp. 4-12.
- Brown, Michelle (2003). *The Lindisfarne Gospels: Society, Spirituality & the Scribe*, The British Library, pp. 85-86.
- Brown, Michelle (2004). *Painted Labyrinth: The World of the Lindisfarne Gospels*, The British Library.
- Browne, Cameron (1998). Font decoration by automatic Mesh fitting, *Lecture Notes in Computer Science*, Vol. 1375, pp. 23-43.
- Browne, Cameron (2005). Cantor knots, *Computers and Graphics*, Vol. 29, Issue 6, pp. 988-1003.
- Browne, Cameron (2006). Wild knots, *Computers and Graphics*, Vol. 30, Issue 6, pp. 1027-1032.
- Buick, James (1997). *Lattice Boltzmann Methods in Interfacial Wave Modelling*, Ph.D. thesis, The University of Edinburgh.

- Burgio, Lucia, et.al. (1979). Pigment identification on medieval manuscripts, paintings and other artifacts by Raman microscopy: applications to the study of three German manuscripts, *Journal Molecular Structure*, 405, pp. 1-11.
- Canny, John (1986). A Computational Approach to Edge Detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-8, No. 6, pp. 679-698.
- Castlmen, Kenneth (1996). *Digital Image Processing*, Prentice Hall, pp. 460-461.
- Celniker, George, and Dave Gossard (1991). Deformable Curve and Surface Finite-Elements for Free-Form Shape Design, *ACM SIGGRAPH Conference Proceedings, Computer Graphics*, Vol. 25, No. 4. pp. 257-266.
- Chen, Jidong, Xiaofeng Meng, Yanyan Guo, Stephane Grumbach, and Hui Sun (2006). Modeling and Predicting Future Trajectories of Moving Objects in a Constrained Network, *Proceedings of the 7th International Conference on Mobile Data Management*.
- Chien, Sung-Il, and Yung-Mok Baek (1998). A fast black run rotation algorithm for binary images, *Pattern Recognition Letters*, Vol. 19, pp. 455-459.
- Christie, Archibald (1929). *Pattern Design: An Introduction to the Study of Formal Ornament*, Dover, 1969 reprint of the 1929 2nd edition of *Traditional Methods for Pattern Designing*.

- Chu, Nelson, and Chiew-Lan Tai (2005). MoXi: Real-Time Ink Dispersion in Absorbent Paper, *ACM Transactions on Computer Graphics*, Vol. 24, No. 3.
- Conway, J. H. (1970). An enumeration of knots and links, and some of their algebraic properties, in John Leech (ed.), *Computational Problems in Abstract Algebra*, pp. 329–358.
- Cramp, Rosemary, et. al. (2006). *Corpus of Anglo-Saxon Stone Sculpture*, Durham University and The British Academy, <http://www.dur.ac.uk/corpus>.
- Corridori, Chiara, and Michele Zanin (2004). High curvature two-clothoid road model estimation, *Proceedings, The 7th International IEEE Conference on Intelligent Transportation Systems*, pp. 630-635.
- Cromwell, Peter (1993). Celtic knotwork: Mathematical Art, *The Mathematical Intelligencer*, 15(1): 36-47.
- Cromwell, Peter (2004). A Topological Analysis of Interlaced Ornament, Preprint, retrieved on December 17, 2007, from <http://www.liv.ac.uk/~smp02/maths/interlace.ps.gz>.
- Danielsson, Per-Erik, and Magnus Hammerin (1992). Note High-Accuracy Rotation of Images, *Graphical Models and Image Processing*, Vol. 54, No. 4, pp. 340-344.

Davis, Neil (1981). Bird Navigation, *Alaska Science forum*, Article #512, retrieved on April 14, 2008, from <http://www.gi.alaska.edu/ScienceForum/ASF5/512.html>.

de Hamel, Christopher (1994). *A History of Illuminated Manuscripts*, Phaidon.

de Hamel, Christopher (2001). *The British Library Guide to Manuscript Illumination: History and Techniques*, University of Toronto Press.

DeCarlo, Doug, et.al. (2003). Suggestive contours for conveying shape, *Proceedings of SIGGRAPH 2003*.

Dickmanns, Ernst and Mysliwetz, Birger (1992). Recursive 3-D Road and Relative Ego-State Recognition, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 14, No. 2.

Donis, Donis (1973). *A Primer of Visual Literacy*, MIT Press.

Doyle, Richard (2005). *Computer generated Celtic design: patterns explored by cellular automata*, unpublished experimental results.

Duran, B. (1995). Mathematical Sophistication of the Insular Celts-Spirals, Symmetries, and Knots as a Window onto their World View, *Proceedings of the Harvard Celtic Colloquium*, Vol. 15, Issue 1, pp. 258-289.

Drago, Frederic (2004). Painting Canvas Synthesis, *The Visual Computer*, 20, pp. 314-328.

- Drewes, Frank, and Renate Klempien-Hinrichs (2000). Picking Knots from Trees: The Syntactic Structure of Celtic Knotwork, *Lecture Notes in Computer Science*, Vol. 1889, pp. 77-106.
- Edmonds, Columba (1908). St. Columba, in *The Catholic Encyclopedia*, Vol. 4, Robert Appleton Company, retrieved on January 30, 2008, from <http://www.newadvent.org/cathen/04136a.htm>.
- Edwards, H., et.al. (2001). Application of FT-Raman spectroscopy to the characterization of parchment and vellum, I; novel information for paleographic and historiated manuscript studies, *Spectrochimica Acta Part A*, 57, pp. 1223-1234.
- Edwards, H., and Fernando Perez (2004). Application of Fourier transform Raman spectroscopy to the characterization of parchment and vellum, I; Effect of biodeterioration on spectral interpretation, *Journal of Raman Spectroscopy*, 35, pp. 754-760.
- Elam, Kimberly (2001). *Geometry of Design*, Princeton Architectural Press.
- Elimeliah, Craig (2006) art Vs. design, *AIGA*, January 13. Retrieved on April 11, 2008, from <http://www.aiga.org/content.cfm/art-vs-design>.
- Farin, Gerald (2002). *Curves and Surfaces for CAGD: A Practical Guide*, 5th Ed., Morgan Kaufmann and Academic Press.

Farr, Carol (1975). *The Book of Kells: Its Function and Audience*, The British Library and University of Toronto Press.

Ferwerda, James, Holly Rushmeier, and Benjamin Watson (2002). Psychometrics 101: How to Design, Conduct, and Analyze Perceptual Experiments in Computer Graphics, *Course Notes of SIGGRAPH 2002*.

Forest, James (1997). *Praying with icons*, Orbis Books.

Freeman, William, Joshua Tenenbaum, and Egon Pasztor (2003). Learning Style Translation for the Lines of the Drawing, *ACM Transactions on Graphics*, Vol. 22, No. 1, pp. 33-46.

Fisher, Gwen, and Blake Mellor (2004). On the Topology of Celtic Knot Designs, *Proceedings of the 7th Annual BRIDGES Conference*.

Fishwick, Paul (2006). *Aesthetic Computing*, MIT Press.

Fitzalleyne, Frae (1997). The work of angels: The Design and Production of Hiberno-Saxon Manuscripts, *Tournaments Illuminated*, Issue #124, Society for Creative Anachronism, Australia, Summary by Mark Calderwood,
<http://www.sca.org.au/tmi/showarticle.php?article=20000130/211844&mode=flat>.

- Fleischer, Kurt (1995). *A Multiple-Mechanism Developmental Model for Defining Self-Organizing Geometric Structures*, Ph.D dissertation, California Institute of Technology.
- Foley, James, Andries van Dam, Steven Feiner, and John Hughes (1996). *Computer Graphics: Principles and Practice*, 2nd Ed. Addison-Wesley.
- Forest, Jim (1997). *Praying with Icons*, Orbis Books.
- Freeman, William (2003). Learning Style Translation for the Lines of a Drawing, *ACM Transactions on Computer Graphics*, Vol. 22, No. 1, pp. 33-46.
- Fung, King (2007). *Celtic Knot Generator*, BSc. (Hons) thesis, University of Bath.
- Ganguly, Niloy, et. al. (2003). *A Survey on Cellular Automata*, Technical Report, Center for High Performance Computing, Dresden University of Technology.
- Gardner, Andrew, et. al. (2003). Linear Light Source Reflectometry, *ACM Transactions on Computer Graphics*, Vol. 22, No. 3, pp. 749-758.
- Gerber, Edwin (2005). A Stochastic Model for the Spatial Structure of Annular Patterns of Variability and the North Atlantic Oscillation, *Journal of Climate*, Vol. 18, pp. 2102-2118.

Gerdes, Paulus (1989). Reconstructions and Extension of Lost Symmetries: Examples from the Tamil of South India, *Computers and Mathematics with Applications*, Vol. 17, No. 4-6, pp. 791-813.

Gerdes, Paulus (1990). On ethnomathematical research and symmetry, *Symmetry: Culture and Science*, Vol. 1, No. 2, 154-170.

Gerdes, Paulus (1997b). On the geometry of Celtic knots and their Lunda-designs, *Mathematics in School*, Vol. 28, No. 3, pp. 29-33.

Gips, James, and George Stiny (1975). An Investigation of Algorithmic Aesthetics, *Leonardo*, Vol. 8, No. 3.

Glassner, Andrew (1996). Andrew Glassner's notebook: Frieze groups, *IEEE Computer Graphics and Applications*, 16(3), pp 78-83.

Glassner, Andrew (1999a). Andrew Glassner's notebook: Celtic Knotwork, Part I, *IEEE Computer Graphics and Applications*, 19(5), pp 78-84.

Glassner, Andrew (1999b). Andrew Glassner's notebook: Celtic Knotwork, Part II, *IEEE Computer Graphics and Applications*, 19(6), pp 82-86.

Glassner, Andrew (2000). Andrew Glassner's notebook: Celtic Knotwork, Part III, *IEEE Computer Graphics and Applications*, 20(1), pp 70-75.

- Gonzalez, Rafael, and Richard Woods (1992). *Digital Image Processing*, Addison-Wesley Publishing Company, pp. 418-419.
- Gray, Alfred (1998). *Modern Differential Geometry of Curves and Surfaces*, 2nd Ed. CRC Press.
- Greenfield, Gary (2005). On the Origins of the Term 'Computational Aesthetics,' in L. Neumann, M. Sbert, B. Gooch, and W. Purgathofer (eds.), *Computational Aesthetics in Graphics, Visualization and Imaging*, The Eurographics Association.
- Grünbaum, Branko, and G. C. Shephard (1986). *Tilings and Patterns*, W. H. Freeman.
- Hannah, Gail (2002). *Elements of Design: Rowena Reed Kostfellow and the Structure of Visual Relationships*, Princeton Architectural Press.
- Hardy, J., and O. de Pazzis (1976). Molecular dynamics of a classical lattice gas: Transport properties and time correlation functions, *Physical Review A*, pp. 1949-1961.
- Harris, C., and M. J. Stephens (1988). A Combined Corner and Edge Detector, *Proceedings British Machine Vision Association Conference*. Proceedings of The Fourth Alvey Vision Conference, Manchester, pp 147-151
- Haseloff, Gunther (1987). Insular Animal Styles with Special Reference to Irish Art in the Early Medieval Period, in Michael Ryan (ed.), *Ireland and Insular and Art A.D. 500-1200*, Royal Irish Academy, Dublin.

- Hayes, Jean Anne (2005). *Anglian Leadership in Northumbria, 547 A.D. through 1075 A.D.*, Ph.D dissertation, Louisiana State University and Agricultural and Mechanical College.
- Henry, Françoise (1974). *The Book of Kells: Reproductions from the Manuscript in Trinity College Dublin*, Thames and Hudson.
- Heudin, Jean-Claude (ed.) (1999). *Virtual Worlds: Synthetic Universes, Digital Life, and Complexity*, Perseus Books.
- Hitchens, Megan (1996). Building on Belief: The Use of Sacred Geometry and Number Theory in the Book of Kells, f. 33r, Society for Creative and Anachronism, retrieved on June 19, 2007, from http://www.sca.org.au/scribe/articles/building_on_belief.htm
- Hornby, Gregory (2001). The advantages of Generative grammatical encodings for physical design, *Proceedings of the 2001 Congress on Evolutionary Computation*, Vol. 1, pp. 600-607.
- Hsu, S., et. al. (1993). Skeletal Strokes, *Proceedings 6th Annual ACM Symposium on User Interface Software and Technology (UIST)*.
- Huang, S, et.al. (2003). Physically-Based Model of Ink Diffusion in Chinese Ink Paintings, *Proceedings of WSCG*.

- Hull, Derek (2003). *Celtic and Anglo-Saxon Art: Geometric Aspects*, Liverpool University Press.
- Isenberg, Tobias, et.al. (2006). Non-photorealistic rendering in context: an observational study, *Proceedings of the 4th International symposium on Non-photorealistic animation and rendering*.
- Itti, Laurent, N. Dhavale, and F. Pighin (2004). New realistic avatar and head animation using a neurobiological model of visual attention, *Proceedings of SPIE 48th Annual International Symposium on Optical Science and Technology*.
- Jablan, Slavick (1995). Mirror generated curves, *Symmetry, Culture, and Science, quarterly publication for the Interdisciplinary Study of Symmetry*, Vol. 6, Issue 2, pp. 275-278.
- Jacobsen, J., (2002). The combinatorics of alternating tangles: from theory to computerized enumeration, in *Asymptotic Combinatorics with Application to Mathematical Physics*, pp. 97-112.
- Jones, David (2007). Improving the accuracy of transmembrane protein topology prediction using evolutionary information, *bioinformatics*, Vol 23, No. 5, pp. 538-544.
- Jowers, Iestyn (2006). *Computation with Curve Shapes: Towards Freeform Shape Generation in Design*, Ph.D dissertation, The Open University.

- Kaplan, Matthew, and Elaine Cohen (2003). Computer Generated Celtic Design, *Eurographics Symposium on Rendering*.
- Kaplan, Craig (2002). *Computer Graphics and Geometric Ornamental Design*, Ph.D dissertation, University of Washington.
- Kauffman, Louis (1994). Knot Automata, *Proceedings*, 24th International Symposium on Multiple-Valued Logic.
- Kepner, Jeremy (2001). Parallel Programming with MatlabMPI, *Proceedings of the High Performance Embedded Computing (HPEC 2001) workshop*, September 25-27, MIT Lincoln Laboratory, Lexington, MA.
- Kicinger, Rafal, Tomasz Arciszewski and Kenneth De Jong (2005). Parameterized versus Generative Representations in Structural Design: An Empirical Comparison, *Proceedings of the 2005 conference on Genetic and evolutionary computation*, pp. 2007-2014.
- Kohonen, Tuevo (2001). *Self-Organizing Maps*, 3rd Edition, Springer.
- Koo, John (1996). *Canny Edge Detector Algorithm and Matlab Codes*, retrieved on April 3, 2004 from <http://robotics.eecs.berkeley.edu/~mayi/imgproc/cacode.html>.
- Lawlor, Robert (1982). *Sacred Geometry: Philosophy and Practice*, Thames and Hudson.

- Leborg, Christian (2006). *Visual Grammar*, Princeton Architectural Press.
- Lee, Y. (1984). Automatic Finite-Element Mesh Generation from Geometric Models – a Point-Based Approach, *ACM Transactions on Graphics*, Vol. 3, No. 4, pp. 287-311.
- Lentin, Louis (2000). *The Book of Kells: The Work of Angels?* Crescendo Concepts Ltd.
- Liu, Wenyin, et.al. (2001). A System for Performance Evaluation of Arc Segmentation Algorithms, *Proceedings of the third CVPR Workshop on Empirical Evaluation Methods in Computer Vision*.
- (2008). *Matlab*, Online Documentation, The MathWorks, Inc., Natick, MA.
<http://www.mathworks.com/access/helpdesk/help/techdoc/matlab.html>.
- Malevanets, Anatoly, and Raymond Kapral (1996). Links, Knots, and Knotted Labyrinths in Bistable Systems, *Physical Review Letters*, Vol. 77, No. 4.
- Medina, Ricardo (2005). *Computational Explorations of Creativity and Innovation in Design*, Ph.D dissertation, University of Sydney.
- Meehan, Aidan (1991b). *Celtic Design: Knotwork: The Secret Methods of the Scribes*, Thames and Hudson, reprinted in 2002.
- Meehan, Aidan (2003). *Celtic Knots: Mastering the Traditional Patterns*, A Step by Step Guide, Thames & Hudson.

Meehan, Bernard (1994). *The Book of Kells: An Illustrated Introduction to the Manuscript in Trinity College Dublin*, Thames and Hudson.

Meehan, Bernard (1996). *The Book of Durrow: A Medieval masterpiece at Trinity College Dublin*, Roberts Reinhart.

Meek, D., and R. Thomas (1991). A guided clothoid spline, *Computer Aided Geometric Design*, Vol. 8, Issue 2, pp. 163-174.

Meek, D., and D. Walton (1992). Clothoid Spline Transition Spirals, *Mathematics of Computation*, Vol. 59, No. 199, pp. 117-133.

Mello-Thoms, Claudia, C. Nodine, and H. Kundel (2002). What attracts the eye to the location of missed and reported breast cancers? in *Proceedings of the Eye Tracking Research and Applications (ETRA) Symposium*, pp. 111-117.

Mengistu, Temesgen, and Wahid Ghaly (2004). Single and Multipoint Shape Optimization of Gas Turbine Blade Cascades, *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*.

Mercat, Christian (1997). Les Entrelacs des Enluminures Celts, *Pour La Science*, No. 15. A very complete English translation is available at the following web page:
<http://www.abbott.demon.co.uk/mercatmethod.html>.

Mercat, Christian (1998). *Holomorphie discrete et modele d'Ising*, Ph.D. thesis, University of Strasbourg I-Louis Pasteur University.

Mercat, Christian (1997). *Celtic Knotwork, the ultimate tutorial*,
<http://www.entrelacs.net/en.index.html>.

Merne, John (1931). *A Handbook of Celtic Ornament*, The Education Company of Ireland, Ltd., Dublin.

Miall, R., and John Tchalenko (2001). A Painter's Eye Movements: A Study of Eye and Hand Movement during Portrait Drawing, *Leonardo*, Vol. 34, No. 1, pp 35-40.

Mitchell, Melanie, James Crutchfield, and Rajarshi Das (1996). Evolving Cellular Automata with Genetic Algorithms: A Review of Recent Work, *Proceedings of the First International Conference on Evolutionary Computation and Its Applications*.

Moore, G. F. (1893). The Vulgate Chapters and Numbered Verses in the Hebrew Bible, *Journal of Biblical Literature*, Vol. 12. No. 1., pp. 73-78.

Muller-Hannemann, Matthias (1997). *Quadrilateral Mesh Generation in Computer-Aided Design*, Ph.D dissertation, Technische Universitat Berlin.

Muller-Hannemann, Matthias (1999). On the discrete core of quadrilateral mesh refinement, *International Journal for Numerical Methods in Engineering*, Vol. 46, pp.

Newcomb, Harvey (2003). *Life of Columba: The Apostle of Scotland*, Kessinger Publishing.2)

NOAA (2007). NWUS54 KLIX 191839 LSRLIX, *Preliminary Local Storm Report*, National Weather Service New Orleans LA, 1:39 pm CDT Thu Jul 2007,
<http://www.nws.noaa.gov/view/prodsByState.php?state=LA&prodtype=public>.

Arica, Nafiz (2003). *Shape: Representation, Description, Similarity and Recognition*, Ph.D dissertation, The Middle East Technically University.

Ó Corráin, Donnchadh, (1998). Viking Ireland--afterthoughts, in H. B. Clarke, Máire Ní Mhaonaigh, and Ragnall Ó Floinn (ed), *Ireland and Scandinavia in the early Viking age*, Dublin, pp. 421-52.

O'Sullivan, David (2001). Graph cellular automata: a generalised discrete urban and regional model, *Environment in Planning B: Planning and Design*, Vol. 28, pp.687-705.

Pamali, Abhinand (2004). *Using Clothoidal Spirals to Generate Smooth Tool Paths for High Speed Machining*, M.S. Thesis, North Carolina State University.

Panfilov, Marko Evanovich (2003). "Design and Construction of Celtic Knotwork," *Society for Creative Anachronism*, <http://www.zuggsoft.com/sca/celtic/celtic.htm>.

Parker, John H. (1851). *The Calendar of the Anglican Church: Illustrated*, Shrimpton.

Retrieved on January 31, 2008, from

<http://books.google.com/books?id=eTw3AAAAMAAJ>

- Parks, Hunter Guymin (2003). *A Process for Creating Celtic Knotwork*, M.S. thesis, Texas A&M University.
- Perlin, Ken (1985). An Image Synthesizer, *Computer Graphics*, Vol. 19, No. 3, pp. 287-296.
- Perlin, Ken (2002). Improving Noise, *Computer Graphics*, Vol. 35, No. 3, 681-682.
- Peters, Mark (1979). *Mathematics and Celtic Design*, B.A. (Hons) thesis, Liverpool School of Art and Design.
- Power, Rosemary (2006). A Place of Community: “Celtic” Iona and Institutional Religion, *Folklore*, Vol. 117, pp. 33-53.
- Prusinkiewicz, Przemyslaw, and James Hanan (1989). *Lindenmayer Systems, Fractals, and Plants*, Lecture Notes in Biomathematics, Springer-Verlag.
- Pulliam, Heather (2006). *Word and Image in the Book of Kells*, Four Courts Press, Dublin, Ireland.
- Reeves, William (tr., ed.) (1874). *The Life of Columba, Founder of Hy, written by Adamnan, Ninth Abbot of that Monastery*, Edmonston and Douglas Edinburgh, accessed through CELT: Corpus of Electronic Texts, Text ID No. T201040, University College,

Cork, Ireland.

Russell, Stuart (2002). *Artificial Intelligence: A Modern Approach*, 2nd Ed, Prentice Hall.

Santella, Anthony (2005). *The Art of Seeing: Visual Perception in Design and Evaluation of non-Photorealistic Rendering*, Ph.D dissertation, Rutgers.

Sarkar, Palash (2000). A Brief History of Cellular Automata, *ACM Computing Surveys*, Vol. 32, No. 1.

Scha, Remko and Rens Bod (1993). Computational Esthetics, English translation of *Computationele Esthetica*, originally published (in Dutch) in: *Informatie en Informatiebeleid* 11, 1, pp. 54-63.

Scharein, Robert (1998). *Interactive Topological Drawing*, Ph.D dissertation, The University of British Columbia.

Scoredos, Sande (2007). Spider-Man 3 - Triple Play, Special Session, *Proceedings of SIGGRAPH '07*, SOMA session video.

Shalizi, Cosma (2006). Chapter 1: Methods and Techniques of Complex Systems Science: An Overview, *arXiv:nlin/0307015v4* [nlin.AO].

Sheffer, Fr. Luke (2007). Consultation on art and iconography, unpublished.

Sheridan, Kirsten (2007). *August Rush*, Warner Bros. Pictures.

Shugrina, Maria, et.al. (2006). Empathic painting: interactive stylization through observed emotional state, *Proceedings of the 4th International symposium on Non-photorealistic animation and rendering*.

Sims, Karl (1994). Evolving 3D morphology and behavior by competition, *Artificial Life IV Proceedings*, Brooks and Maes (ed.), MIT Press, pp.28-39.

Sloss, Andy (1995). *How to Draw Celtic Knotwork: A practical handbook*, Blandford Press, 2002 republication by Cassell & Co., London.

Small, David (1991). Modeling Watercolor by Simulating Diffusion, Pigment, and Paper Fibers, *Proceedings of the SPIE*.

Sousa, Mario, and John Buchanan (1999) Computer-Generated Pencil Drawing, in *Proceedings Western Computer Graphics Symposium*.

Solso, Robert (2001). Brain Activities in a Skilled versus a Novice Artist: An fMRI Study, *Leonardo*, Vol. 34, No. 1, pp. 31-34.

Sousa, Mario, and John Buchanan (2000). "Observational Models of Graphite Pencil Materials," *Proceedings of Eurographics: Computer Graphics Forum*, Vol. 19, No. 1, pp. 27-49.

- Sousa, Mario, and Przemyslaw Prusinkiewicz (2003). A Few Good Lines: Suggestive Drawing of 3D Models, *Proceedings of Eurographics: Computer Graphics Forum*, Vol. 22, No. 3, pp. 381-390.
- Spencer, Dorothy (1989). Computer Art-An Oxymoron? Views from the Mainstream, Special Session, *Proceedings of SIGGRAPH '89*, pp. 211-221.
- Stebbing, Peter D. (2004). A Universal Grammar for Visual Composition? *Leonardo*, Vol. 37, No. 1, pp. 63-70.
- Stevick, Robert (1994). *The Earliest Irish and English Bookarts: Visual and Poetic Forms before A.D. 1000*, University of Pennsylvania Press.
- Stiny, George (1975). *Pictorial and Formal Aspects of Shape and Shape Grammars: On Computer Generation of Aesthetic Objects*, Birkhauser Verlag Basel.
- Stiny, George (2001). How to Calculate with Shapes, in Erik Antonsson (ed.), *Formal Engineering Design Synthesis*, Cambridge University Press.
- Stiny, George (2006). *Shape*, MIT Press.
- Suber, Peter (2004). *Knots on the Web*, Department of Philosophy, Earlham College, <http://www.earlham.edu/~peters/knotlink.htm#software>.

- Suber, Peter (2004). *Knot Tying Notation*, retrieved on December 19, 2007, from <http://www.earlham.edu/~peters/knotting/notate.htm>.
- Swenson, Inga (1978). The Symmetry Potentials of the Ornamental Pages of the Lindisfarne Gospels, *Gesta*, Vol. 17, No. 1, pp. 9-17.
- Taylor, R., et. al. (2007). Authenticating Pollock paintings using fractal geometry, *Pattern Recognition Letters*, Vol. 28, pp. 695-702.
- Taylor, Grant (2004). *The Machine that Made Science Art: The Troubled History of Computer Art 1963-1989*, Ph.D dissertation, University Western Australia.
- Thurston, H. A. (1954). Celtic Interlacing Patterns and Topology, *Science News*, No. 33, pp. 50-62.
- Toffoli, Tommaso, and Norman Margolus (1987). *Cellular Automata Machines: A New Environment for Modeling*, MIT Press.
- Toner, John (1998). Flocks, herds, and schools: a quantitative theory of flocking, *Physical Review E*, Vol. 58, No. 4, pp. 4828-4858.
- Tsai, Du-Ming (2002). Rotation-invariant pattern matching using wavelet decomposition, *Pattern Recognition Letters*, Vol. 23, pp. 191-201.
- Tyler, Tim (2007). *Invertible Wolfram automata*, retrieved March 13, 2007, from

http://cell-auto.com/inv_wolfram/index.html.

- Van Laerhoven, Tom, et.al. (2004). *A Paper Model for Real-time Watercolor Simulation*, Technical Report, TR-LUC-EDM-0403, Limburgs Universitair Centrum.
- Van Laerhoven, Tom, et.al. (2004). Real-time Watercolor Painting on a Distributed Paper Model, *Proceedings of the Computer Graphics International*.
- Van Laerhoven, Tom, and Frank Reeth (2005). Real-time Simulation of Watery Paint, *Computer Animation and Virtual Worlds*, 16, pp. 429-439.
- Voss, Richard (1988). Fractals in nature: From characterization to simulation, in Heinz-Otto Peitgen (ed.), *The Science of Fractal Images*, pp. 21–70.
- Walmisley, Michael (2004). *The Design and Implementation of the Celtic Knot Drawing Package with Multimedia Help Guides*, B.S. (Hons) thesis, University of Bath.
- Werner, Martin (1990). The Cross-Carpet Page in the Book of Durrow: the Cult of the True Cross, Adomnan, and Iona, *The Art Bulletin*, Vol. 72, No. 2, pp. 174-223.
- Wiley, Keith (2006). *Druid: Representation of Interwoven Surfaces in 2 ½ D Drawings*, Ph.D dissertation, The University of New Mexico.
- Wolfram, Stephen (2001). *A New Kind of Science*, Wolfram Media, Inc.

- Wong, Michael, Douglas Zongker, and David Salesin (1998). "Computer Generated Floral Ornament," *Proceedings of SIGGRAPH '98*.
- Wong, Wucius (1993). *Principles of Form and Design*, John Wiley & Sons.
- Wonka, Peter, Pascal Muller, Ben Watson, and Andy Fuller (2007). Urban Design and Procedural Modeling, *Courses SIGGRAPH 2007*.
- Wu, Kesheng (2005). Optimizing connected component labeling algorithms, *Proceedings of the SPIE*, Vol. 5747, pp. 1965-1976.
- Yevin, Igor (1997). Art as a self-organizing system, Proceedings of International Conference "Mathematics and Art," Moscow-Suzdal, pp. 104-110.
- Yevin, Igor (2002). Criticality of the Brain and Criticality of the Art, paper #209, *Proceedings of the Fourth International Conference on Complex Systems*, The New England Complex Systems Institute.
- Zhai, Jian, et. al. (2003). A Line Drawings Degradation Model for Performance Characterization, *Proceedings of the Seventh International Conference on Document Analysis and Recognition*, pp. 1020-1024.
- Zhang, Qing, et. al. (1999). Simple Cellular Automata Based Simulation of ink Behavior and Its Application to Suibokuga-like 3D Rendering of Trees, *The Journal of Visualization and Computer Animation*, 10, pp. 27-37.

Zongker, Douglas (2000). *Celtic knot thingy*, <http://www.cs.washington.edu/homes/dougz/hacks/knot/>.

Zucker, Matt (2001). *The Perlin noise math FAQ*, retrieved January 4, 2008, from <http://www.cs.cmu.edu/~mzucker/code/perlin-noise-math-faq.html#algorithm>.

