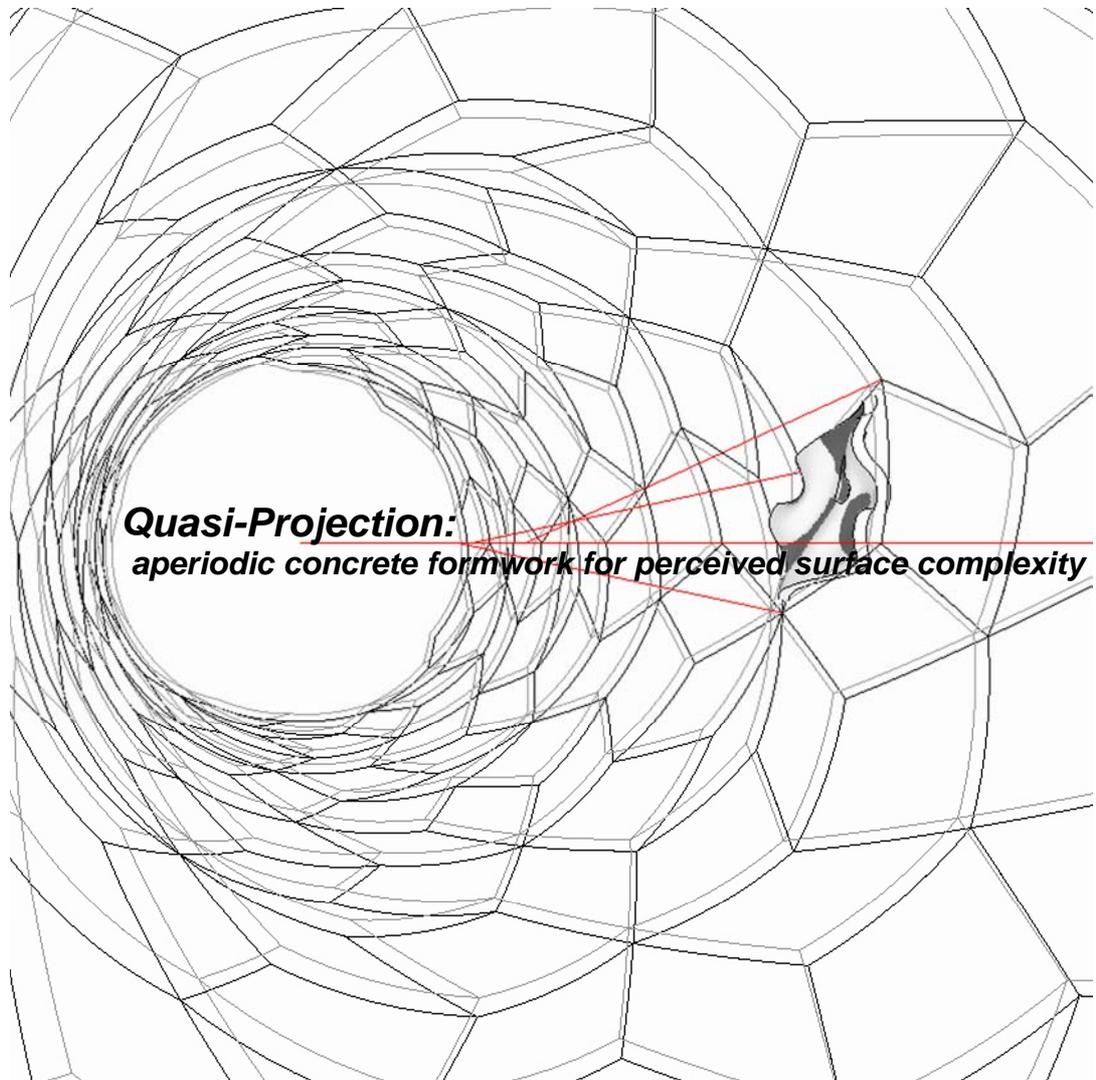


The Bartlett School of Graduate Studies- University College London



Olivier Ottevaere

This dissertation is submitted in partial fulfilment of the requirements for the degree of Master of Science in Adaptive Architecture & Computation from University of London

University of London, September 2008

I, Olivier Ottevaere, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Abstract

By disclosing long range order from few dissimilar tiles, aperiodic tilings can potentially diversify and spatially enrich the repetitive aspects of modular systems still pertinent in the production of architecture today. Such effective tilings have been discovered in quasicrystals and can be generated by the projection of higher dimensional grids in two or three dimensions. A Penrose tiling, for example can be derived from the projection of five dimensional grids onto a two dimensional plane.

The thesis initially investigates if a program allowing the grids to be rotated parametrically can provide for numerous alternative tillings using the projection method for any dimensions.

Some found tilings are then analysed and their assembly rules tested against the adaptation of other types of geometries in order to determine if a high level of diversity can still sustain the test of repetition of few different modules and field a spatial configuration of probable forces.

It is further demonstrated that these initial tilings can in fact perform as efficient organizational scaffolds by letting more complex geometries free flowing past the tiles' edges and pass the test of mass production with the aid of a minimum amount of formwork.

Word count: 10279 + (3217 for the supplemented Adaptation and Fabrication chapters)

Keywords: Quasicrystals, aperiodic tiling, strip projection method, assembly rules, adjacencies, porosity, tangential continuity, formwork, modularity.

Acknowledgments

I would like to thank my supervisors:

Sean Hanna for his insightful guidance and his critical opinions on the topic throughout the research

Alasdair Turner for his invaluable programming assistance, particularly in propelling the plunge in n dimensions

I would also like to thank:

Elsa Caetano for her inexhaustible help with *generative components*

Stylianos Dritsas for his numerous suggestions and expertises on the computation part

Guan Lee for never stopping to challenge and inspire the physical counterpart so pertinent to the thesis

Contents

1. Introduction	1
<i>The whole dilemma</i>	
<i>Aperiodic structure</i>	
<i>Motif versus pattern</i>	
2. Background	3
<i>Motif recognition</i>	
6.1 Scientific approaches	4
<i>The Penrose tiling; a special case</i>	
<i>Quasitiler and the projection method</i>	
6.2 Architectural correlations	8
<i>Modularities</i>	
<i>Quasi furniture</i>	
<i>Orgone Reef</i>	
3. Research objectives	12
<i>The projection method from arbitrary rotations</i>	
<i>Adjacency analysis</i>	
<i>Adaptation</i>	
<i>Fabrication</i>	
4. Methodology	13
<i>Tool used</i>	
8.1 The projection strip method	13
<i>Transformation matrices</i>	
<i>The 4 and 5d hypercube experiment</i>	
<i>The strip or clipping boundary</i>	
<i>In jumping down one dimension</i>	
<i>In jumping down two dimensions</i>	
<i>In jumping down more than two dimensions</i>	
8.2 Adjacency analysis	27
<i>Case study #1 to 6</i>	
5. Adaptation	34
<i>NURBS Line type</i>	
<i>Tangencies</i>	
<i>Fleeing the tile's original shape</i>	
<i>Porosity</i>	
<i>Quasi roller</i>	
6. Fabrication	45
6.1 Fieldproof	45
6.2 Tileproof	46
<i>Quasi wall prototype</i>	
<i>Quasi roller prototype</i>	
7. Discussion	51
<i>Overall impressions</i>	
<i>The strip projection method by arbitrary rotations</i>	
<i>Adaptation to fabrication</i>	
<i>Further Work</i>	
8. Conclusion	55
<i>Repetitions of differences</i>	
References	56
Appendix	58

1. Introduction

The recent and ongoing advance of CAD/CAM* technologies has considerably shrunk the passage between the digital and its physical counterpart, portraying it to the designer almost seamless and nearly non-existent. Yet this passage has been and remains an important conceptual ground for the designer to reside throughout the conception of a project. In doing so, she or he allows the design development to be informed back and forth by the digital as well as the physical.

Today, such conversion that CAD/CAM technologies facilitate, tends to polarize the design process more towards the digital and to consider its physical being later in the process.

While recognizing the advantages and the potentials of this smooth conversion (e.g. more sophisticated forms and structure are made physically possible nowadays by means of this type of technology), design projects are often submitted to the law of *fragmentation*.

This, results in the breaking apart of an entity into single and unique units, manufactured and then assembled into its original whole but this time for *real*. The simple fact that data can almost effortlessly be processed by a computer to a machine renders *efficiency* less crucial and the customization of individual or unique pieces more predominant. However a building today cannot still be manufactured in one single piece and the unavoidable issue of fragmentation originates more from a conceptual level than from the level of *eased* means of fabrication.

The whole dilemma

A distinction could be drawn between the relationships of a *fragment* and its *whole* versus the one of a *part* to its *whole*. By definition, a fragment is a piece cut off of something else. Many sets of fragments can make up this something else but which specific one is less so relevant. On the contrary, a part or a component is a constituent of a composite entity suggesting that it has intrinsic properties related to the whole and vice versa. Therefore, could some sort of intentional bondage between the specific constituents make for a singular whole? This initial question prompted the following thesis to investigate the ambiguity between parts and whole, but *in reverse*. Could one initially conceive of building units and their organizational properties and arrive at a unique spatial and physical assemblage? That is without being too restricted about what that (assembled) entity must be to start with. Could the rules making up this entity be embedded within the parts and therefore conducive of their assembly?

*CAD stands for Computer-aided design

*CAM stands for Computer-aided manufacturing and is the use of computer-based software tools that assist engineers and machinists in manufacturing or prototyping product components (www.wikipedia.org), or even soon entire buildings.

Aperiodic structure

A key motivation for the thesis was to generate relatively intricate entities from the least possible parts, calling for some sense of economy of means and efficiency without compromising the quest for diversity.

Similar characteristics to the ones just stated had been identified in aperiodic structures, such as the ones found in Quasicrystals. They comprised of structures made of few different tiles (or units) which are combined in a non repetitive manner and which project long range order. Rotational symmetry makes their non repetitive combination possible as opposed to periodic structures which only have translational symmetry (can be copy and paste next to one another). Quasicrystalline structures became the core material from which these questions were researched.

Motif versus Pattern

In “The Self-Made Tapestry”, Philip Ball defines pattern as “*arrays of units that are similar but not necessarily identical, and which repeat but not necessarily regularly or with a well-defined symmetry*” (p.9).

Patterns are rendered by *external forces* and travel extensively in space, Motifs are static and self-contained. Which category would Quasicrystals fall into? It could be argued that rotational symmetries although less immediate than translational ones are nonetheless well-defined symmetries. Therefore, aperiodic tiling could then be classified at best as enhanced motifs which are finite and bounded in space. Either this is accurate or not, the distinction made here only highlights one of the dangers in working with aperiodic tilings and raises a more definite question that this research would strive to tackle: -If aperiodicity can present *maximum* variation with *minimum* variety of tiles, would it suffice to perform as a provisional scaffold to govern only organized efficiency and allow other types of geometries to be spatially let free of the strict edges and recognizable motif those tilings ultimately depict? And if so, could this leave room for other types of geometries to be conditioned further in the process by external forces (physical) and become more responsive to its milieu?

A general approach to this question was

First to fully disclose the searching space inside which aperiodic structures emerge so that convergence to only special cases might be overcome (Penrose tiling)

Then gradually test how far other types of geometry can distance themselves from a found aperiodic tiling without disposing of the rules accounting for its initial efficiency.

2. Background

Motif recognition

Even though aperiodic structure and space packing (3d aggregates) have been an explored material for quite some time now within the scientific realm, their growing interests have been seen in recent years applied to architecture. Today well known examples of them are the *water cube* (National Swimming Centre) for the Beijing Olympics by PTW Architects and Arup (*fig. 2a*) or the RMIT Storey Hall in Melbourne, Australia by ARM Architects (*fig. 2d, 2f*). A thorough or fair architectural critic of these specific projects is beyond the scope of this research but it could be suggested that, while not undermining their respective technical achievements (i.e. structure and facade of the water cube), the strong look-alike proximity to the original scientific material is somewhat conspicuous of their architectural shortcomings. The water cube for instance borrows the Weaire-Phelan packing directly from its two authors of the same names (*fig. 2b*) to only operate a series of transformations (rotation and slicing) on it (*fig. 2c*). This was mainly prompted by a will from the architects to render the original structure of aggregates visually less repetitive while not risking a drift too far away from the initial metaphor; from water to soap bubbles.



Fig. 2a Water Cube, in Beijing, China (from the Architects <http://www.ptw.com.au/>)

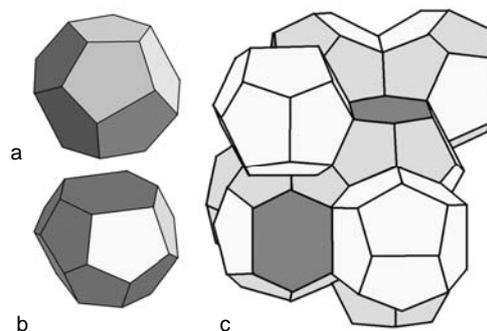


Fig. 2b Weaire-Phelan space packing aggregates
a: irregular pentagonal dodecahedron
b: tetrakaidecahedron
c: base cluster: 2 of a and 6 of b

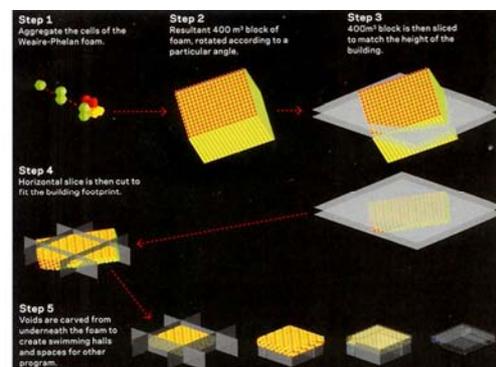


Fig. 2c procedure of transformations from the same aggregates for the water cube (from ARUP Australia)



Fig. 2d Parts of RMIT Storey Hall, in Melbourne, Australia (from the Architects <http://www.a-r-m.com.au/>)



Fig. 2f Parts of RMIT Storey Hall, in Melbourne, Australia (from the Architects <http://www.a-r-m.com.au/>)

“Our brief from RMIT for Storey Hall was the creation of an international standard exhibition and auditorium facility, and one which could make a **significant contribution to architectural theory and practice**, and to the present and future identity of RMIT.”
 (From the Architects at <http://www.a-r-m.com.au/>)

The reader may feel slightly uneasy with the presented evidences to this statement and a return later to why it may be architecturally problematic is inevitable.

2.1 Scientific approaches

But where these strong iconographic precedents partly originated from? Quasi periodic structures were discovered in 1982 within the field of crystallography from electrons diffraction disclosing patterns (*fig. 1a*) with icosahedral (20 sided) symmetry*. They are special crystals with no translational symmetry. That is, contrary to crystals they cannot repetitively align themselves as tiles or building blocks to fill up space without resorting to rotational symmetry (*see difference from fig.2a and fig.2b*). Crystals have close range order, whereas quasicrystalline structures disclose long range order even though they are comprised of only few different tiles.

* From “Quasicrystals to Kleenex”, by Alison Boyle
 (<http://plus.maths.org/issue16/features/penrose/>)

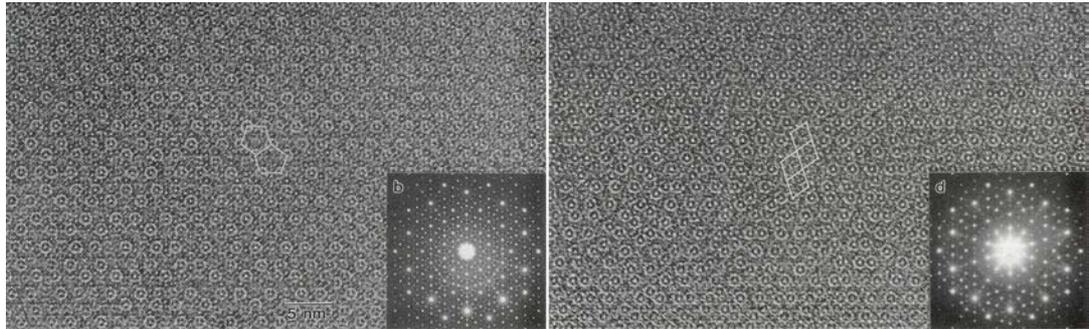


fig.1a High resolution images of electron diffraction pattern of the Al-Cu-Co allow annealed at 550 °C. (from Quasicrystals, The State of the Art, World Scientific, 1999, p.106)

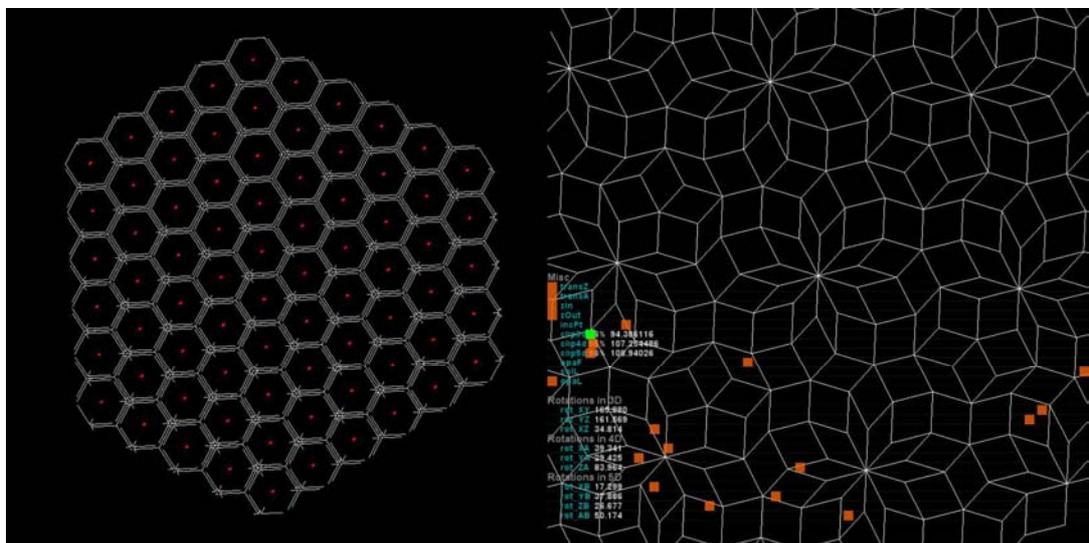


fig.2a crystalline structure with translational symmetry (derived from a sectioning of a three dimensional grid: 1 tile type)

fig.2b quasicrystalline structure with rotational symmetry (derived from 2d projection of a five dimensional grid: 10 tiles type)

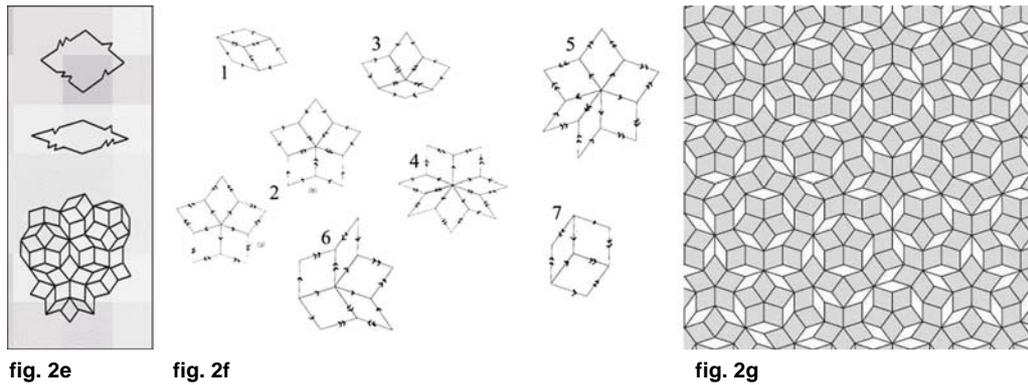
The Penrose tiling; a special case

An exhaustive documentation already exists on Penrose sets of tiles. But in short, they were devised (and patented) by the British mathematician Roger Penrose in the 1970's prior to the discovery of quasicrystalline structures (1984). What makes them particularly interesting is their ability to cover a plane or a space (2d or 3d) in an aperiodic manner only using two types of tiles: skinny and fat rhombi for the rhombus Penrose tiling (*fig. 2g*). There exist as well other types of Penrose tiling such as the pentagonal Penrose tiling using pentagons (instead of rhombi). The growth of a Penrose tiling is regulated by a finite number of (local) matching rules between tiles (*fig.2e, 2f*).

fig. 2e Two pairs of Penrose tiles and their grouping (from <http://plus.maths.org/issue18/features/penrose/>)

fig. 2f The 7 vertex stars allowed in the Penrose rhomb tiling (from <http://intendo.net/penrose/>)

fig. 2g An aperiodic Penrose tiling using 2 rhombi (from <http://plus.maths.org/issue45/features/kaplan/>)



Penrose tilings are briefly introduced here not only because of the later discovery of similar structure in Quasicrystals but more for the reason that they presented a break for a relevant question asked by Marjorie Senechal*: Can Penrose tilings be alternatively generated, using a projection method?

Quasitiler and the projection method

By then, N.G. De Bruin* (1981) already showed that Penrose tilings could be seen as the projection of an object in 5-dimensional space (5d to 2d) onto a plane. However the question was put to task and generalized by Eugenio Durand to become the aspiration behind the program Quasitiler (*fig 2h*) that he later went on to develop. Unfortunately the program itself is no longer available for use but a useful description of the method still remains (<http://www.geom.uiuc.edu/apps/quasitiler/>).

It mainly consisted in drawing Penrose tilings from the projection of five dimensional grids onto a two dimensional plane. Quasitiler also offers drawings of other types of tilings from 4, 5 and more dimensional grids. Prior to projection, the vertices making those grids need to be selected or sliced from a specific region.

“Eugenio Durand, a Geometry Center programmer, has written the program QuasiTiler to find the described quasiperiodic tilings of the plane. He originally wrote it to help Marjorie Senechal with her work on quasicrystals. The program allows the user to specify the “slope” of the plane E (see fig. 2j), using a mouse to modify a picture of the five-dimensional unit cube. There are three degrees of freedom for the offset of the plane. The user uses three sliders to change the offset. One of the offset directions specifies whether the tiling is a Penrose tiling. The program then shows the tiling that the user has specified. In addition, the user may specify a lattice dimension other than five. It is an easy program to use, and the results are beautiful.”

(From <http://www.geom.uiuc.edu/docs/forum/quasitiler/quasitiler.html>)

Rotation angles appear to be contained in quasitiler inside an overall rotational matrix (summary of many rotational matrices into one). However this precedent became an inspiring benchmark for the research to follow.

*Marjorie Senechal has been a leading figure for years in the field of mathematical crystallography (<http://maven.smith.edu/~senechal/>).

*M. Senechal, *Quasicrystals and Geometry*, Cambridge University Press, 1995.

*N.G.deBruijn, *Algebraic theory of Penrose's nonperiodic tilings of the plane, I, II*, *Nederl. Akad. Wetensch. Indag. Math.* 43 (1981) 39-52, 53-66.

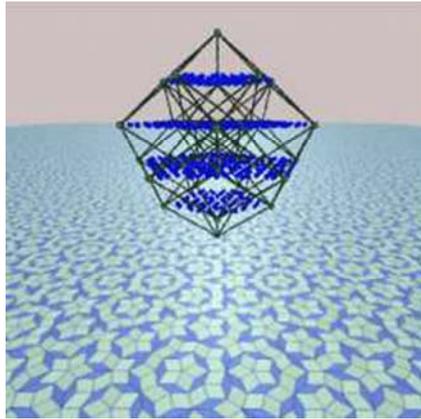


fig. 2h Penrose tiling from Quasitiler 3.0
(from <http://www.geom.uiuc.edu/apps/quasitiler/>)

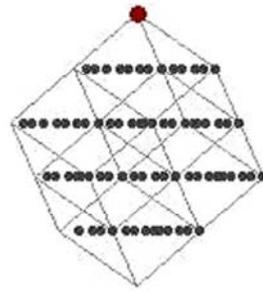


fig. 2i selection region for a Penrose tiling
(from <http://www.geom.uiuc.edu/apps/quasitiler/>)

The slicing region of n -dimensional lattices is given by an area just wide enough to fit one unit of an n -dimensional lattice (*fig. 2j, 2i*).

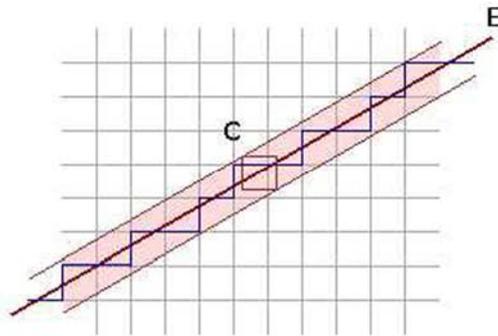


fig. 2j Example of slicing region in 2d for a projection of points onto a 1d line (from <http://www.geom.uiuc.edu/apps/quasitiler/>)

Alternative methods are also known today for generating Penrose tilings such as the *Updown generation** method (*fig 2k*) and the *dual grid* or *Pentagrid** method (*fig 2l*). These do not have a direct connection to the following thesis and therefore are only briefly illustrated and referenced. For the 'Updown generation' method, half rhombi are combined under some specific assembly rules to recursively make an aperiodic tiling (*fig. 2k*).

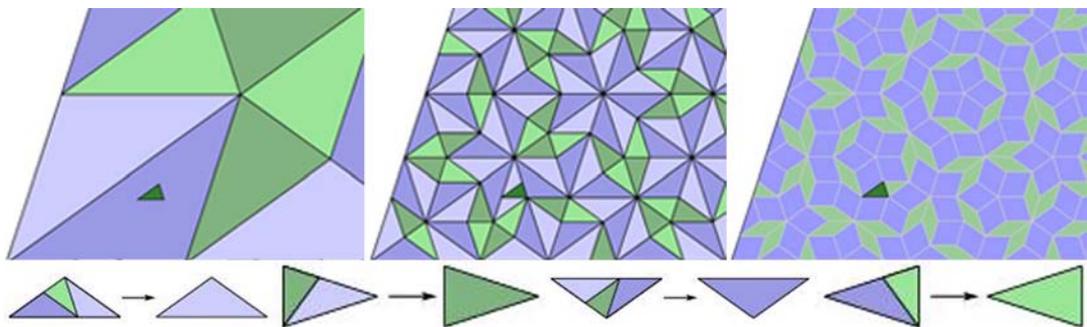


fig. 2k Example of the updown generation method for drawing a Penrose tiling
(from <http://www.ams.org/featurecolumn/archive/ribbons.html>)

For the 'Pentagrid method', a Penrose tiling can be thought as a series of parallel ribbons in five directions. The intersection between non parallel

ribbons makes a rhombus belonging to the Penrose tiling. So each rhombus can be defined from a series of intersecting straight lines. Ultimately those intersecting lines can be used to generate a Penrose tiling (*fig. 2l*).

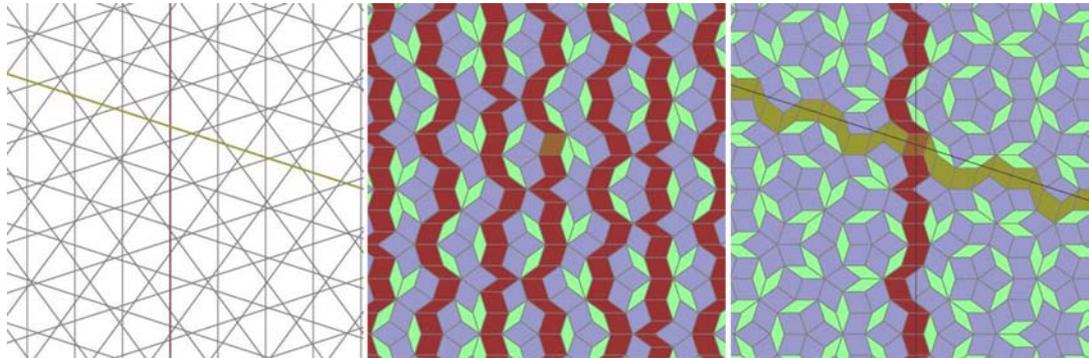


fig. 2l Example of the dual grid or Pentagrid method for drawing a Penrose tiling
(from <http://www.ams.org/featurecolumn/archive/ribbons.html>)

* *The Updown generation method is explained in more detail by David Austin at <http://www.ams.org/featurecolumn/archive/ribbons.html>*

* *The dual grid or Pentagrid method is as well explained by David Austin at <http://www.ams.org/featurecolumn/archive/ribbons.html>*

Dr. Eric R. Weeks wrote a program called Quasig using the Pentagrid method which can be found at <http://condellpark.com/kd/quasig.htm>

6.2 Architectural correlations

Modularity

Some architects remain captivated by the idea that elements can be effectively repeated to make a whole. Along with the aid of an ever evolving scientific knowledge and technology, the modernist project continues spreading its mission for mass production in architecture, although increasingly at the brink of retinal and procedural exuberance. “Less is more” has now the potential to be toppled by a pinch of “more for less”. At what expense? This should be revisited towards the end of this thesis, but undoubtedly this phenomenon has been facilitated by the tools employed?

Repetition of differences

As one of Greg Lynn latest projects, the “Blobwall” best epitomizes the above, an initial module is repeated into wall-type patterns generated in the computer and causing each one of its modules to digitally overlap with its neighbours (*fig. 2m*). To permit the components to physically interlock and the wall-type to stand, all individual and redundant intersections have to be trimmed off by a well behaved six-axis robot (*fig. 2mb*). The manufacturing process is here individualized (individual trimmings) while the rough material (unit) is the same to begin with. It is clear to say that if the overlap for each module would have had to be traced and trimmed by a human hand (manageable), an effort to make the process more efficient would have had to be pondered on further. However, if a robot can provide for some relative easiness, why not exploiting it? In fact, during a lecture last year at the Architectural Association in London,

he was supportive of an architecture being a singular object built out of many variant components. In regards to the “Blobwall”, a more central question persists: does this forceful process make up conceptually for a more interesting wall and if so, what are the criteria to judge it? History may help.



fig. 2m “Blobwall” by Greg Lynn (from <http://www.glforn.com/blobwall.html>)

Repetition of same

For the last 50 years, the Austrian born artist Erwin Hauer has been developing tectonic screens which main intentions are to temper with visual movements while still allowing diffused light to go through. His screens (*fig.2n*) are made of the same repeated module but are two-sided. Each module is precast and mass produced manually. Using a *suture curve** as its main spatial articulation, they disclose an interstitial voided space continually fleeing away from the moving eye. An ever changing presence that dynamically makes up for the fact each cast module is the same throughout.



Fig 2n example of a screen by Erwin Hauer (from *Continua--Architectural Screens and Walls*, by Erwin Hauer, 2007)

* *A suture curve is “a way of partitioning the surface of a sphere into 2 identical halves... The makers of baseballs and tennis ball have chosen it to fabricate their products.”* (From *Continua--Architectural Screens and Walls*, p 84, 2007)

Repetitions of difference

The last modular precedent (*fig.2o*) developed by Gramazio/Kohler at the ETH in Zurich, Switzerland differs from the previous two by bringing out a stronger sense of rationality. It accepts the bricks as primitive artefacts and contrasts their limitations as such by extending the ways they can literally form a wall.

In their work the robot has taken charge for the bricklayer. It sequentially positioned the bricks row by row and is informed by sets of coordinates from created patterns in the digital realm. The process emphasizes more on a series of instructions bounded by the brick size and its consideration to gravity and experiment on how far it can be articulated as a formal standing wall in perplex interplay between mass and void.



fig. 2o "Programmed wall" by Gramazio/Kohler (from <http://www.dfab.arch.ethz.ch/web/e/lehre/index.html>)

The reasons why these three examples were presented here was not so much to validate one successful approach over the other two, but more to reflect on the field of already pre-existing possibilities and on there respective premises in tackling modular systems.

Quasi furniture

Aranda/lash, a young architectural practice based in New York City has been interested in quasicrystalline structures since a few years now. They are currently testing their ideas mainly in small scale prototypes (*i.e. furniture in fig. 2m*) and believe these aperiodic structures can challenge new variations in form and surface that they can further transform into architecture. Although it is not very clear how they generate the (3d) tilings themselves, their results portray an immediate resemblance to the scientific material presented before. Nonetheless they are quite effective as pieces of furniture. While serving their purposes, as objects they neither come out as lavishly decorative: A concern that appears harder to conceal when having to deal with architectural space.



Fig 2m quasi cabinet
(from <http://www.johnsontradinggallery.com/>)

quasi table

quasi chair

The final precedent proposes an alternative take to the friction already brought up earlier (motif recognition) and a more ventured approach to space, further away from a restricted bounding box.

Orgone Reef (by Philip Beesley)

“Orgone Reef is a speculation of what the skin of a building could be like in the future. Orgone Reef is a technical exercise in construction and fabrication. The project relates to geotextiles, a new class of materials used for reinforcing landscapes and buildings. A minimal amount of raw material is expanded to form a network forming a porous volume. A **Penrose tessellation**, a non-repeating geometrical system, is used to create the hybrid fabric. This structure acts like an artificial reef that could support a living skin.”
(from Philip Beesley: <http://www.philipbeesleyarchitect.com/index.html>)

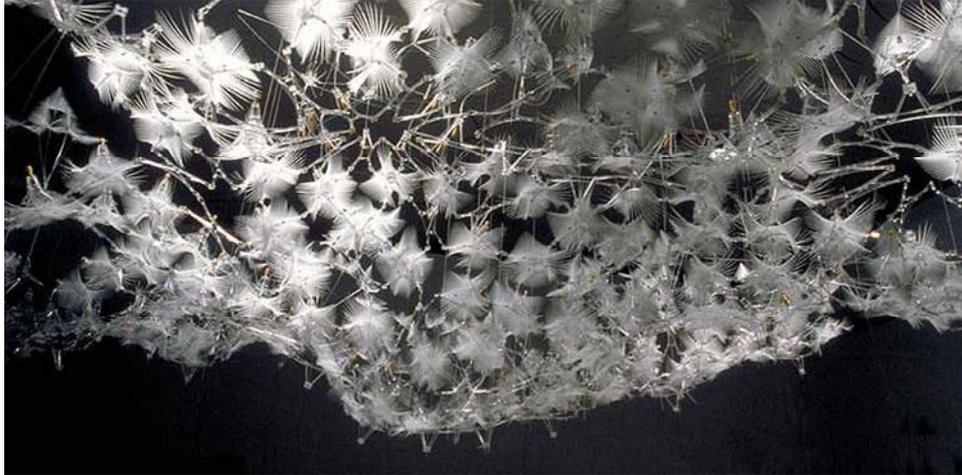


Fig. 2o View of Orgone Reef at the London Building Centre (<http://www.philipbeesleyarchitect.com/index.html>).

Philip Beesley is a Canadian architect whose experimental kinetic structures have been widely commended. Orgone Reef uses a Penrose tessellation (fig. 2nd) to organize its intricate hybrid fabric. One contrasting aspect of his approach is how on one hand he rigorously overlays and organizes his structure along the lines of the Penrose tiling and of its associated assembly rules (fig. 2na, 2nd) and yet manages to spatially depart away from the tiling strong graphical presence and at last allowing its structure to become autonomous (fig 2o). Indeed, he uses the Penrose tiling only as an efficient organizational scaffold and not any longer as a dominant and decorative feature.

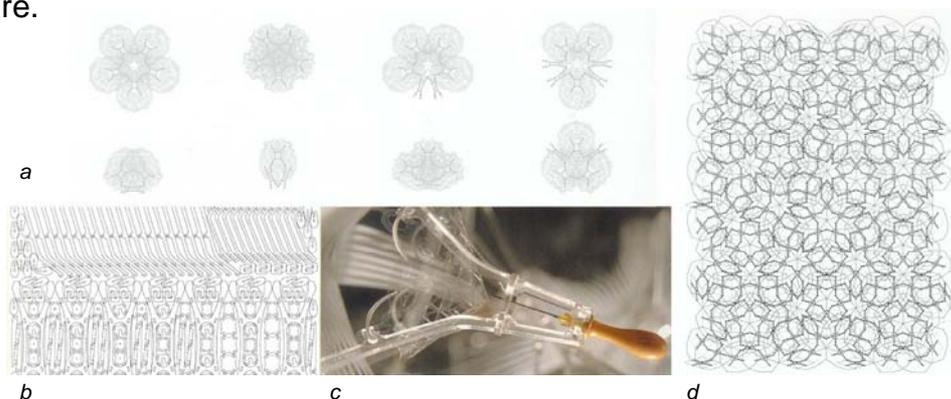


Fig 2na Penrose tessellation assembly rule: alternate configurations for rhombic structural units
Fig 2nb nested laser-cutting production layout, showing cutting paths for snap-fit assembly elements
Fig 2nc units are positioned within the membrane filter layer.
Fig 2nd unfolded meshwork structure showing a self-generating pattern of interlinked rhombic units
(Images and captions taken from AD “design trough Making”, July/August 2005, p. 49, 50)

3. Research objectives

The central research question was:

Can quasiperiodic tilings be employed as efficient means of producing complex and visually irregular patterns with a minimal amount of reusable formwork involved?

The thesis is structured into four major sections in undertaking the research question:

1) *The projection method from arbitrary rotations*

The main part of the thesis was to create a parametric program to see if alternative tiling to those already familiar like the Penrose tiling could be found. Two and three dimensional projections of higher dimensional grids (4d, 5d, 6d...) from any arbitrary rotations, was the method used to generate aperiodic structures.

2) *Adjacency analysis*

The second phase was to analyse the organizational structure of some of these findings (tilings) from various dimensions, such as the number of different tiles a tiling will be made of (from a specific dimension) as well as their inherent adjacency rules regulating how such tiling is put together.

3) *Adaptation*

The next objective was to investigate how alternative geometries and more specifically NURBS* surfaces could be created in accord with those adjacency rules (shared edges and tangencies from both sides of an edge, overall curvature of tile, etc) while still making up for a field with long range order and with certain degrees of adaptive freedom.

4) *Fabrication*

The ultimate aspiration for this research was to arrive to some spatial and physical evidences of how with few a tiles, a large field of diversity can be efficiently mass produced. This was accomplished by milling the necessary moulds required for each type of tiles, by casting them in a repetitive manner and by assembling them according to the rules of the overall (found) tiling. For this phase an attention to details was inevitable in defying gravity (reinforcements/connections) and avoiding *leakages*.

* **Non-uniform rational B-spline (NURBS)** is a mathematical model commonly used in computer graphics for generating and representing curves and surfaces (wikipedia.org).

8. Methodology

The main tools used in the research were processing (*processing.org*) for the programming part, generative components (*Bentley parametric software*) for the adaptation part and a 3 axis CNC (*stands for Computer Numerical Control*) milling machine for the fabrication part.

8.1 The projection strip method

In general, the projection method consists of selecting for an n -dimensional grid all the points falling within an area defined by a *clipping* boundary (*or the strip*) of a one unit of the n dimensional grid and projecting those selected points onto at least an $n-1$ dimensional (hyper) space. In this research, the projection is done onto a two dimensional plane or within a three dimensional space for any higher dimensional grid (i.e. From 3 to 2, from 4 to 3 or 2, from 5 to 3 or 2, from 6 to 3 or 2, etc).

When speaking of higher dimensions we quickly enter a domain where visibility is quite limited. Their representation is often difficult to imagine. Nonetheless, transformations can be performed on these **extra** parameters ($x, y, z, \mathbf{a}, \mathbf{b}$) and their perception brought back in three dimensions in various ways; projections (in $n-1$ or more) is one of those. The following section illustrates a quick experiment of how higher dimensional transformations prove to be quite powerful and on how those higher dimensions can be brought back to some sort of visualization.

8.1.1 Transformation matrices

At this stage the concept of transformation matrices (essential in working with higher dimensions) have to be presented as they became a central operator in this research. It is well known in the field of computer graphics that matrices are used to transform vectors in space or more precisely their coordinates ($\mathbf{x}, \mathbf{y}, \mathbf{z}, a, b, c$, etc). Examples of those possible transformations are translations, mirroring, rotations, orthogonal projections and 1, 2, 3 points perspectives. In 3d, we are accustomed to three possible rotations: Around the X-Y plane, the Y-Z plane and the Z-X plane. For each rotation exists a specific rotation matrix which when multiplied by a vector, will cause it to rotate. The same thing applies for the other types of transformations.

A generalized 4x4 transformation matrix in 3d is as follow:

The extra column and row (\mathbf{a}) are added to account for the other types of transformations.

$$\begin{array}{l} \begin{array}{c} x \\ y \\ z \\ a \end{array} \begin{array}{|c|c|c|c|} \hline \mathbf{Ra} & \mathbf{Rb} & \mathbf{Rc} & \mathbf{px} \\ \hline \mathbf{Rd} & \mathbf{Re} & \mathbf{Rf} & \mathbf{py} \\ \hline \mathbf{Rg} & \mathbf{Ri} & \mathbf{Rj} & \mathbf{pz} \\ \hline \mathbf{tx} & \mathbf{ty} & \mathbf{tz} & \mathbf{os} \\ \hline \end{array} \end{array} \text{ times vector } [\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{a}]$$

Where, the \mathbf{Ra} to \mathbf{Rj} group is for rotations, $\mathbf{tx}, \mathbf{ty}, \mathbf{tz}$ for translations, \mathbf{px} for 1 point perspective, \mathbf{px} and \mathbf{py} for 2 points perspective, $\mathbf{px}, \mathbf{py}, \mathbf{pz}$ for 3 points perspective and \mathbf{os} for overall scaling, etc. (*Computer Graphics: Mathematical First Steps : Patricia A. Egerton, William S. Hall: Books, p.124.*)

In 4 dimensions (x, y, z, a, b) we have 6 possible rotation matrices (5x5) around each plane: x-y, x-z, y-z, x-a, y-a, z-a.

In 5 dimensions (x, y, z, a, b, c) we have 10 possible rotation matrices (6x6) around each plane: x-y, x-z, y-z, x-a, y-a, z-a, x-b, y-b, z-b, a-b.

In 6 dimensions (x, y, z, a, b, c, d) we have 15 possible rotation matrices.

In 7 dimensions (x, y, z, a, b, c, d, e) we have 21 possible rotation matrices.

Here is an example of a rotation matrix around the Y-axis or the X-Z plane, in 3d. Angle α can parametrically range from 0 to 360 degrees.

(See appendix A for an example of the all rotation matrices to a given dimension)

$$(x, y, z, 1) \cdot \begin{matrix} & \begin{matrix} x & y & z & a \end{matrix} \\ \begin{matrix} x \\ y \\ z \\ a \end{matrix} & \begin{bmatrix} \cos\alpha & 0 & -\sin\alpha & 0 \\ 0 & 1 & 0 & 0 \\ \sin\alpha & 0 & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{matrix} = (x \cdot \cos\alpha + z \cdot \sin\alpha, y, -x \cdot \sin\alpha + z \cdot \cos\alpha, 1)$$

8.1.2 The 4 and 5d hypercube experiment

A first program was created in order to comprehend better how diverse transformations in higher dimensions interrelate while being represented in 3-dimensional space. This clarified for instance what effect a rotation in 5d around an A-B plane had upon the other dimensional vectors. All calculations were performed first on vectors in high dimensions and then each one was brought down to a 3-dimensional vector in order to be drawn in a 3d Cartesian space following this formula:

$$\begin{aligned} \text{In 4d for } (x, y, z, a) &= (x \cdot da / (a+da), \\ & y \cdot da / (a+da), \\ & z \cdot da / (a+da), \\ & a \cdot da / (a+da)) \sim (x, y, z, 0) \\ \text{In 5d for } (x, y, z, a, b) &= ((x \cdot db / (b+db)) \cdot da) / ((a \cdot db / (b+db)) + da), \\ & (y \cdot db / (b+db)) \cdot da) / ((a \cdot db / (b+db)) + da), \\ & (z \cdot db / (b+db)) \cdot da) / ((a \cdot db / (b+db)) + da), \\ & (a \cdot db / (b+db)) \cdot da) / ((a \cdot db / (b+db)) + da), \\ & (b \cdot db / (b+db)) \cdot da) / ((a \cdot db / (b+db)) + da)) \sim (x, y, z, 0, 0) \end{aligned}$$

da and db are scalars of a certain adjustable range.

A vector and a matrix class were implemented for this program partly to facilitate the multiplication of vectors by matrices. Even though all the rotation matrices (6 in 4d, 10 in 5d) are all multiply into a single one before operating on vectors, they can still be individually manipulated within the interface. This offers the user to interact parametrically with any singular vectorial transformations (rotations, perspectives, orthographic projections, scaling) while simultaneously witnessing the effect it has on the overall geometry.

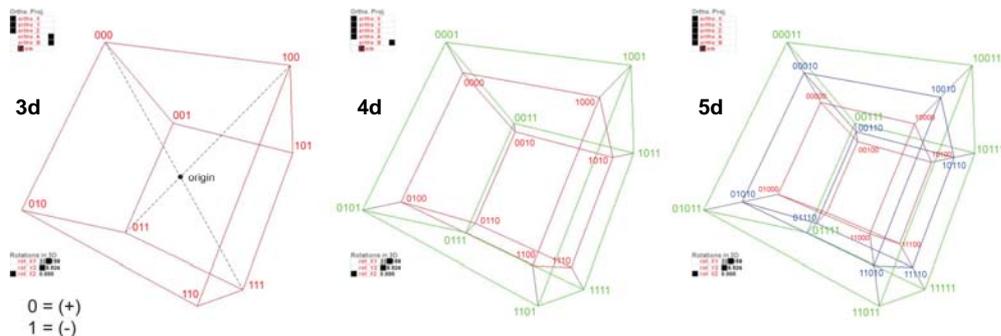


fig. 8a 3d, 4d and 5d representations of vectors in 3d space

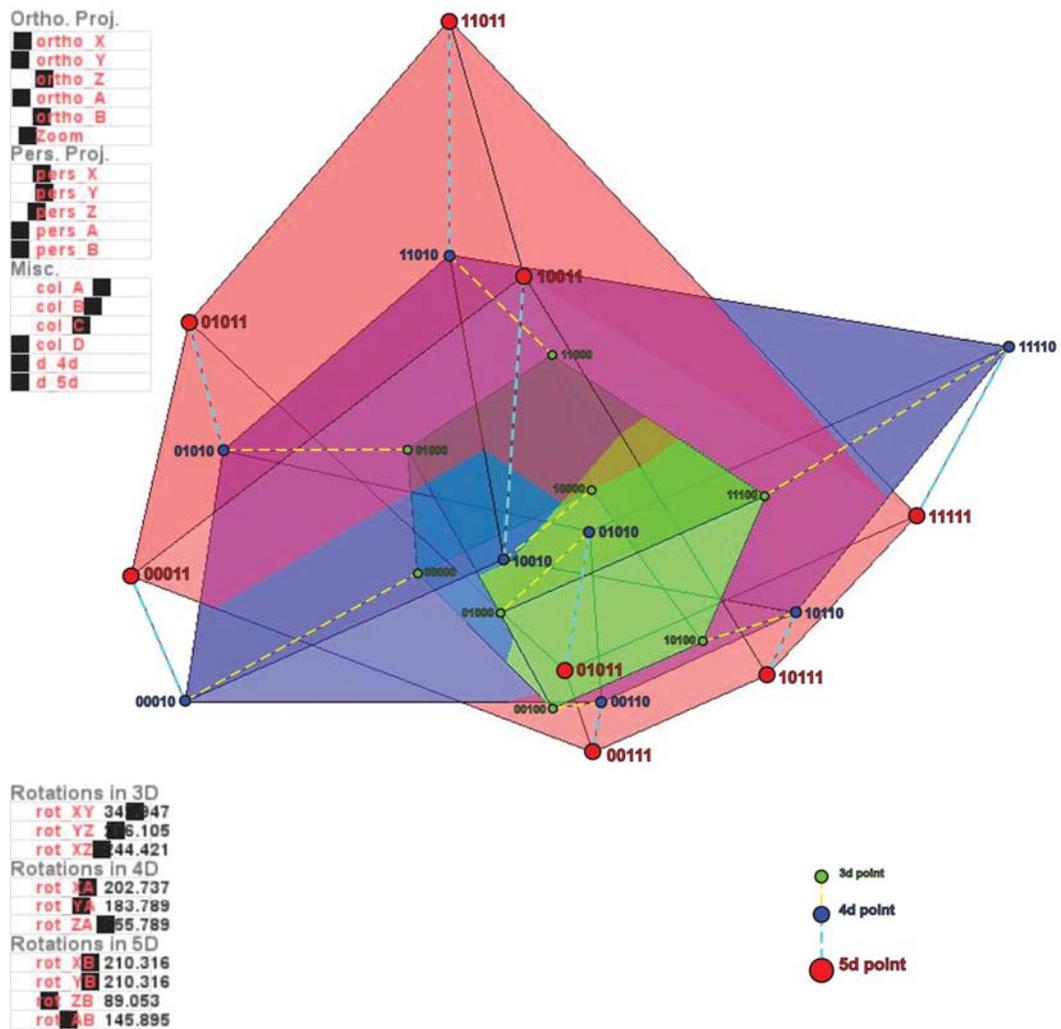


fig. 8b Snap shot from the interface of a 5d hypercube under specific transformation matrices

The development of this small interface turned out to be quite useful in disclosing parametrically the various transformations at work and was later directly employed for the projection method. This set literally the apparatus to search for aperiodic tilings from any compounds of individual and arbitrary rotations. In place of mathematically determining the content of an overall rotation matrix to unveil its corresponding tiling, this procedure permitted to discover alternative tilings along non determinate routes and without any preconceived destination.

By freely altering rotation's angles one by one and in non prescriptive orders, the contact sheet in the next page illustrates some specific encounters. For instance, initially when none or only the rotations in 3d were manipulated, the 5d, 4d and 3d (hyper) cubes were inside of each other respectively (*fig 1a: green, blue and red*). Once some 4d rotations were activated the 4d hypercube was extroverted from the 3d cube (*fig 1c: blue, green and red*). The same occurred for the 5d hypercube under 5d rotations (*fig 1e: red, blue and green*). Now the various rotations can be controlled in any prescriptive orders, causing this interplay of higher dimensional representations to intertwine in many surprising ways.

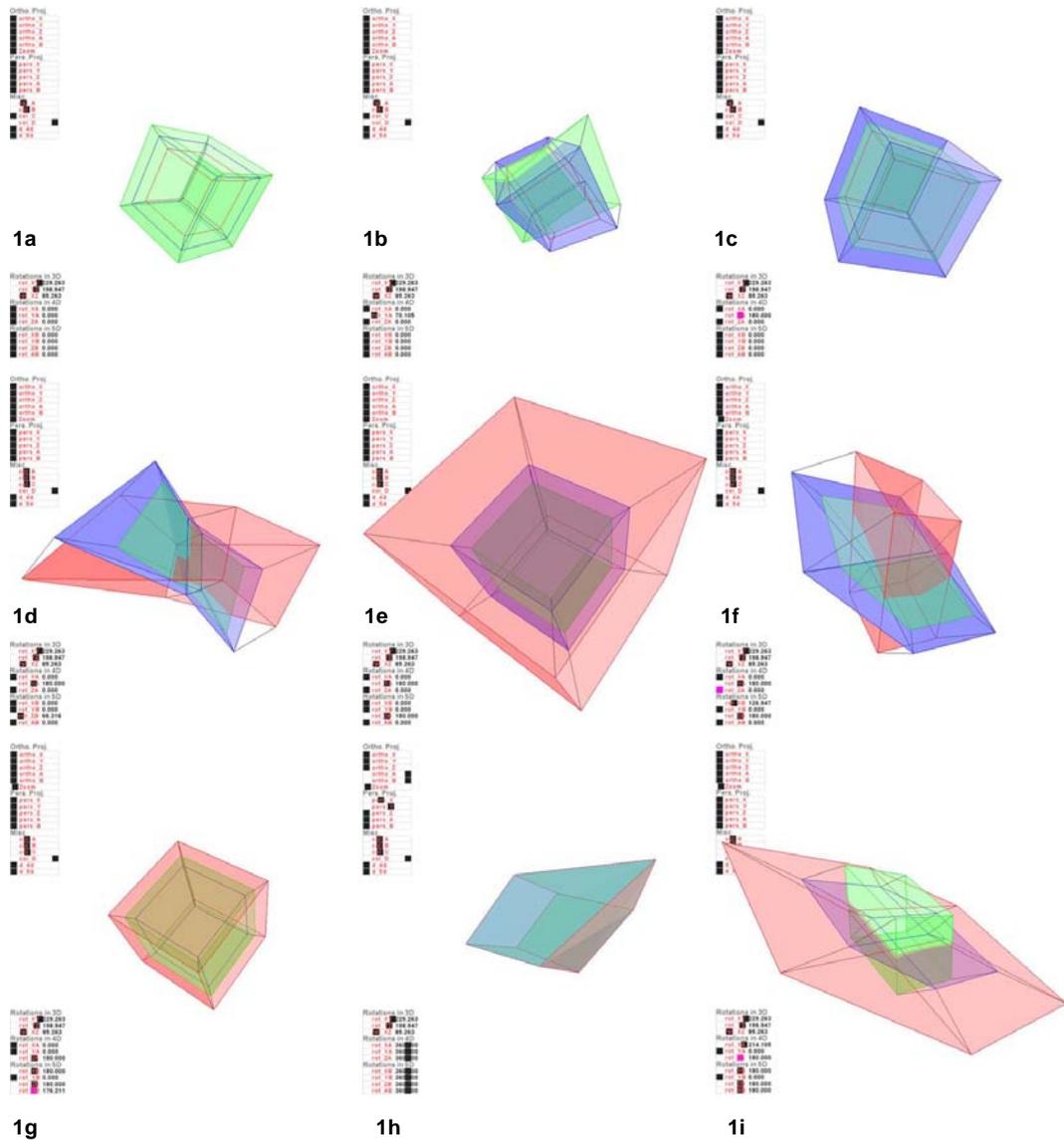


Fig. 8c contact sheet of one-pass through the hypercube interface
 Order of successive transformations: from 1a, 1b, 1c, 1d.....to 1i

8.1.3 The strip or clipping boundary

The strip projection method is a two-step process: one of projection and beforehand one of selection.

The selective method involved what is called a strip; a width between two lines in two dimensions or a space between two planes in three dimensions. This range defined the boundary inside which a set of points were first located, selected and then projected to create a tiling. The range for this method had to be just wide enough to only incorporate all the points falling within one unit of the n -dimensional lattice, no more, no less

(<http://www.geom.uiuc.edu/apps/quasitiler/>). But as the figure below shows, this width (black strip) had to increase or decrease when the lattice was rotated in order to constantly maintain this one unit rule. A first issue was how to make the width of the strip shift to the correct amount when the lattice was subjected to rotations.

8.1.4 Jumping down one dimension

From 2d to 1d (x, y)

For a 2d lattice of a 1 unit, the width of the strip ranges

- from minimum: 1 (the side of the square at rotation x-y = 0 deg)
- to maximum: 1.414 (the diagonal of the square at rotation x-y = 45 deg)

In the case below, the strip is defined by 2 boundary lines:

- 1) $y_1=0$; fixed at the origin (the viewing plane).
- 2) $y_2= [\text{min. to max.}]$; varies upon x-y rotation.

In solving this, a *depth* vector from (0, 0) to (1, 1) was created and multiplied by the rotation matrix. To define the second boundary line, one checks that the points of the lattice are smaller or equal to its y-value.

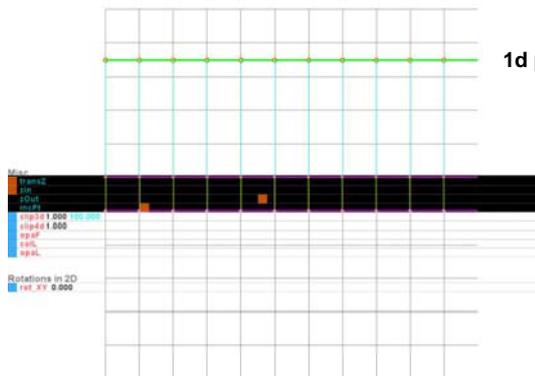
$$(1, 1, 1) * \begin{array}{c|cc} x & y & a \\ \hline \cos a & \sin a & 0 \\ -\sin a & \cos a & 0 \\ 0 & 0 & 1 \end{array} = (\cos a - \sin a, \cos a + \sin a, 1)$$

2) $y_2= [\text{min. to max.}] = \cos a + \sin a$

If $a=45$ deg, $y_2=\cos 45+\sin 45= 1.41$ (= sqrt of 2)

as in fig 8d, if $a=0$ deg, $y_2=\cos 0- \sin 0= 1$

as in fig 8e, if $a =125$ deg, $y_2= 1.39$



1d proj.

y1

y2

Rotations in 2D
rot. XY 0.000

Rotations in 2D
rot. XY 125.492

fig. 8d 2 dimensional lattice projected in 1 dimension

fig. 8e same but rotated disclosing aperiodic structure

Notice the aperiodic structure of the 1d projection of figure 8e. If the magenta segments were 0 and the green ones 1, we would have a series of this sort: ...0101001010010101001...., making it non repetitive as opposed to the periodic structure in figure 8d (11111111111) disclosing only translational symmetry.

Pseudocode

If the y-value of a point $X > 0$ and \leq y-value of the depth vector
($y_1 \text{ line} < X_y \leq y_2 \text{ line}$)

Then point X is inside the clipping boundary and is selected.

After selection only its x-value is drawn (1d projection)

From 3d to 2d (x, y, z)

The same method presented for the 2d to 1d projections applies to the 3d to 2d projections. The *depth* vector goes from (0, 0, 0) to (1, 1, 1) with three rotations matrices around x-y, y-z, z-x.

depth_Vec=(1,1,1)

z1 = 0 (fixed)

M_rotall= M_rotxy * M_rotyz * M_rotzx (multiply the 3 rotation matrices into a single one)

z2 = z_val of depth_Vec * M_rotall (varies upon rotations)

z2 min = 1

z2 max = 1.732 (= sqrt of 3)

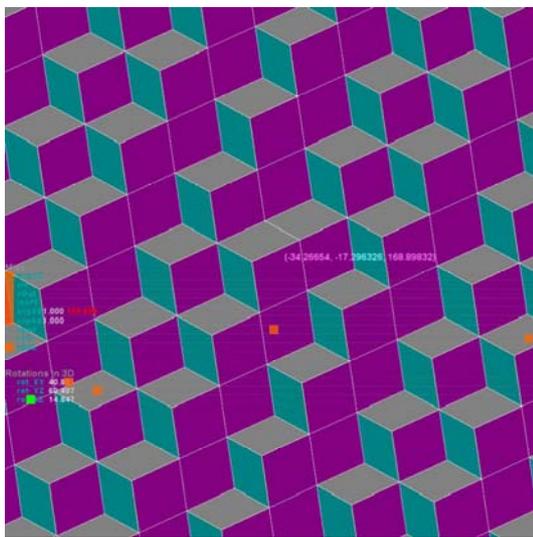


fig. 8f 2d projection on x-y plane of a 3d grid

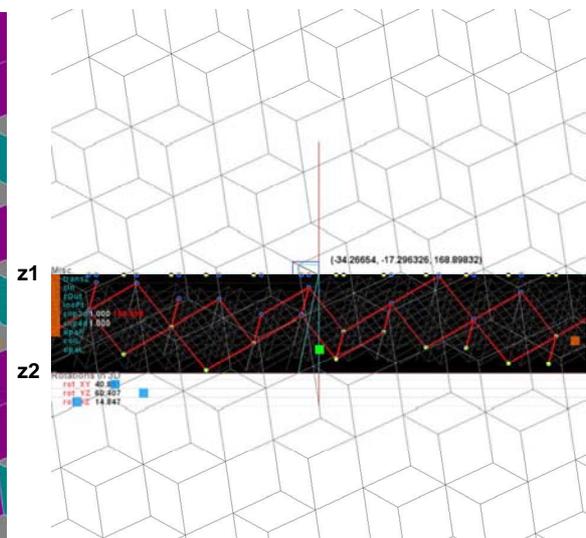


fig. 8g strip width defined by z1, z2 planes

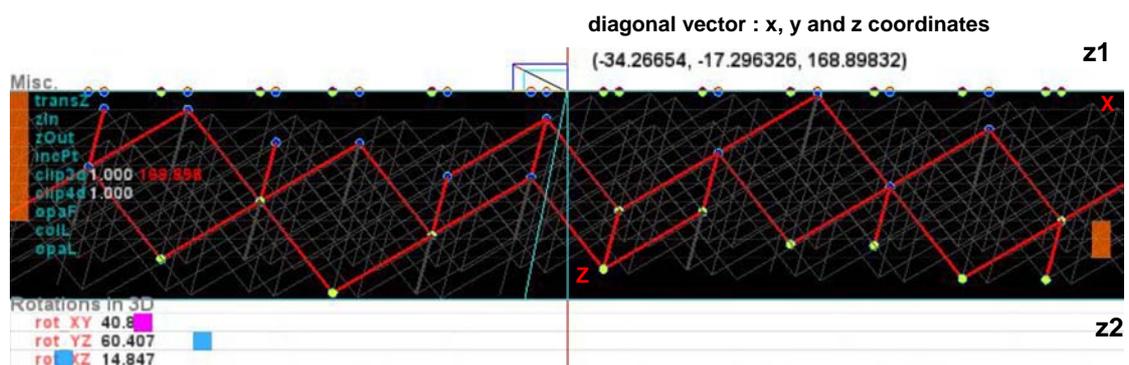


fig. 8h X-Z projection view of a 2d projection tiling from a 3d lattice showing the space and its selection between the two boundary planes (z1,z2).

This method wasn't yet working properly. As the next 2 figures indicate (fig. 8i, 8j), some holes were occurring in the tiling. The selection boundary was not as large as it should have been causing it to not properly select all the points needed to make a complete 2d tiling under the 3 specific rotations.

The problem originated when the *depth* vector was rotated past a certain amount of degrees from any rotation(s). The quadrant in which the *depth* vectors resides has a direct effect on the signs and values of its x, y, z coordinates and therefore on the width of the boundary itself. What needs to be avoided is a negative depth vector (- sign(s) of any of its coordinates caused by the three rotations when changing quadrant). A pre-emptive measure to this problem was to check first if the coordinates of the *depth* vector were negative before it was actually rotated. This was done by checking against its z-value (0, 0, 1) as clipping occurs along the z-axis and by multiplying it by the *inverse** of the overall rotation matrix. If any of its coordinates turned out negative (i.e. (0.651, -0.781, -0.265)), then flip the sign(s) for the same coordinates of the original depth vector (1, -1, -1). Now the depth vector could be correctly rotated from 0 to 360 degrees along with its complete tiling (fig 8k, 8l).

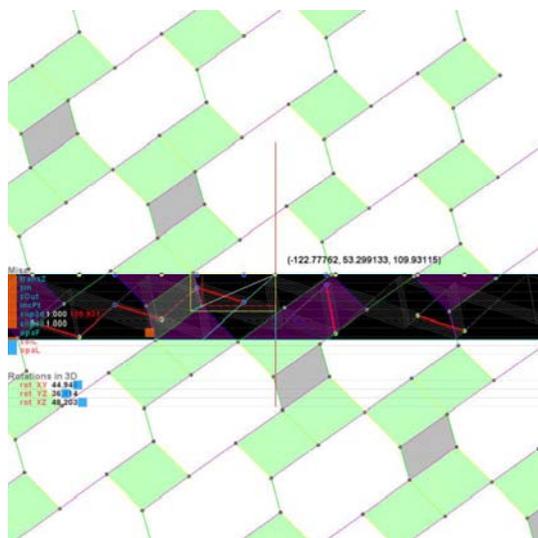


fig. 8i incomplete tiling; z2= 1.09 units

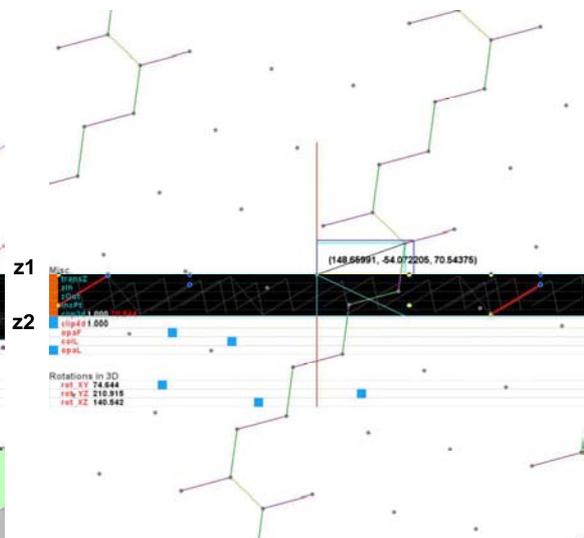


fig. 8j incomplete tiling; z2= 0.70 units

Pseudo code

```
test_Vec = (0, 0, 1)
(0,0,1) * inv.M_rotall (multiply the test_Vec by the inverse of the overall rotation matrix)
//This results in flipping quadrant when coordinate(s) are negative upon rotation angles.
```

If x_val of test_Vec < 0, then make x_val of depth_Vec < 0

If y_val of test_Vec < 0, then make y_val of depth_Vec < 0

If z_val of test_Vec < 0, then make z_val of depth_Vec < 0

depth_Vec * M_rotall (Only then multiply the depth_Vec by the overall rotation matrix)

```
void invRot(Matrix inv) {
    rows=inv.rows;
    cols=inv.cols;
    for(int i=0; i<rows; i++){
        for(int j=0; j<cols; j++){
            matrix[i][j]=inv.matrix[j][i]; }}}
```

*The inverse of a rotation matrix is its transpose (flip along its diagonal)

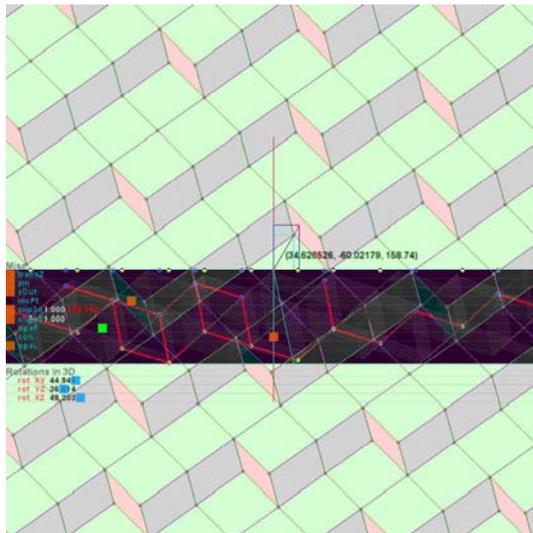


fig. 8k corrected depth vector and tiling from fig. 8i

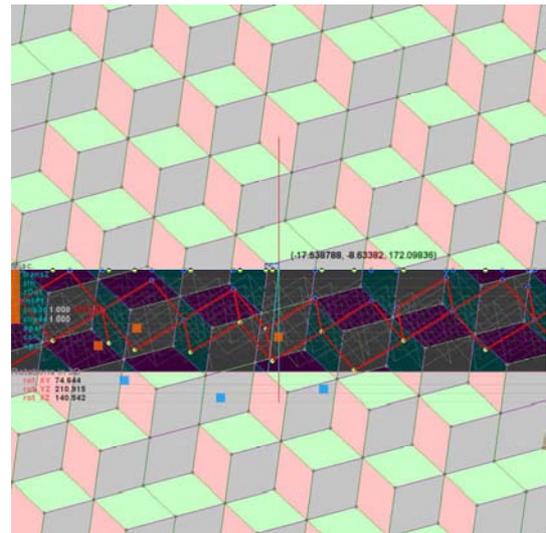


fig. 8L corrected tiling from fig. 8j

In fig. 8i, z_2 was **1.09** units and its depth_Vec was in the **SW** quadrant
 In the corrected fig. 8k, z_2 is **1.58** units and its depth_Vec is now in the **NE** quadrant
 In fig. 8j, z_2 was **0.70** units and its depth_Vec was in the **NE** quadrant
 In the corrected fig. 8L, z_2 is **1.72** units and its depth_Vec is now in the **NW** quadrant

These revised examples illustrate how substantial this boundary error was. Since, one of the premises of this research was to search for tilings from any arbitrary rotations ranging from 0 to 360 degrees, it was crucial for the projection to fully emerge under any rotations but also to progress in accordance with the change of rotations.

8.1.5 Jumping down two dimensions

Until now, the method for stepping down one dimension was satisfactory. Could the same strategy be applied successively when jumping down 2 dimensions from any higher dimensional grids?

From 4d to 2d (x, y, z, a)

As mentioned before in 4 dimensional space we have 6 possible rotations around six planes: x - y , y - x , z - x , x - a , y - a , z - a .

Two dimensions need to be clipped against (z, a) in order to project in 2d (x, y). The key issue was to elucidate if the clipping boundaries in those two dimensions are checked against individually (one after the other) or in a pre-combined manner.

The successive method or 'square' method is a continuation of the way 2d projections were tackled previously with three dimensional grids (fig.8m).

Pseudocode

If the z -value of point $X > 0$ and \leq z -value of the depth vector
 ($z_1 \text{ plane} < X.z \leq z_2 \text{ plane}$)
And if a -value of point $X > 0$ and \leq a -value of the depth vector
 ($a_1 \text{ plane} < X.a \leq a_2 \text{ plane}$)
Then point X is inside the clipping boundary and is selected.

The combined method or ‘circular’ method calculates the projection boundary in both dimensions simultaneously (fig.8n).

Pseudocode

If (the square (z-value of point X) + (the square (a-value of point X)) <= ((square (z-value) + square (a-value))*0.25) of the depth vector
 (z1 plane <(sq(X.z) + sq(X.a))*0.25) < z2 plane)
 Then point X is inside the clipping boundary and is selected before being projected.

The z-a reference plane is used in the following diagrams to visualize the differences between the two approaches.

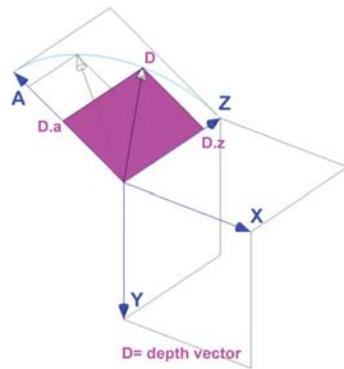


fig. 8m the ‘square’ method where the depth vector is individually calculated in each dimensions (z_depth, a_depth).

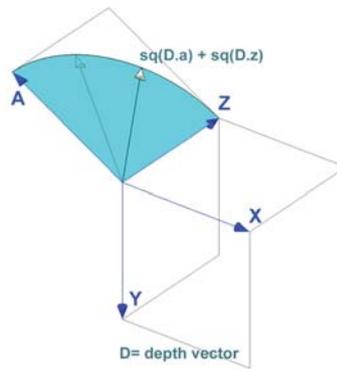


fig. 8n the ‘circular’ method where the depth vector is calculated together for both dimensions (z_depth + a_depth).

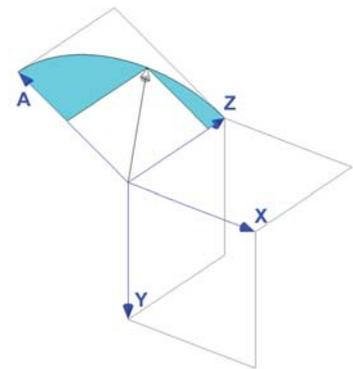


fig. 8o area differences between both methods for the depth vector.

The ‘circular’ method has always a larger selection area upon rotations than the ‘square’ method. For instance, as the depth vector in the ‘square’ method is approaching the a-axis or the z-axis, the area is progressively being reduced to a line. This will inevitably create holes till no points at all will be found within the clipping boundary. The following figures indicate the main problem caused while using the ‘square’ method.

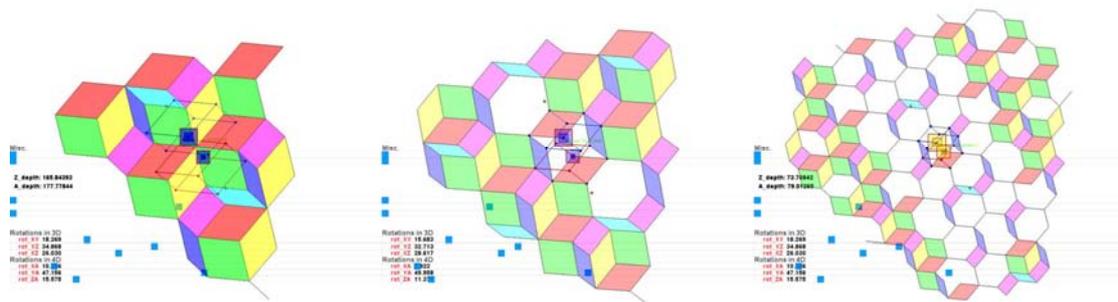


fig. 8p 2d projection of 4d tiling using the ‘square’ method revealing holes in the tiling

This method was computed by initializing the 16 points (x, y, z, a) making up a one unit 4d hypercube and by updating the clipping planes positions as the 4d points were rotated. Two vectors (z_depth, a_depth) constantly spanned 2 of those 16 points in making sure they always incorporated the smallest z_min and a_min and the largest z_max and a_max from those 16 points (fig 8q).

Pseudocode

```

void update clip(){
.....
for( int i = 0; i < c; i++ ) {
    if( block[i].vector[2] >= zmax ) {
        if( block[i].vector[3] >= amax ) {
            zmax = block[i].vector[2];
            amax = block[i].vector[3];
        }
    }
}
if( block[i].vector[2] <= zmin ) {
    if( block[i].vector[3] <= amin ) {
        zmin = block[i].vector[2];
        amin = block[i].vector[3];
        a_min = block[i];
    }
}
CLIP_POZ = z_max;
CLIP_PIZ = z_min;
CLIP_POA = a_max;
CLIP_PIA = a_min;
}

```

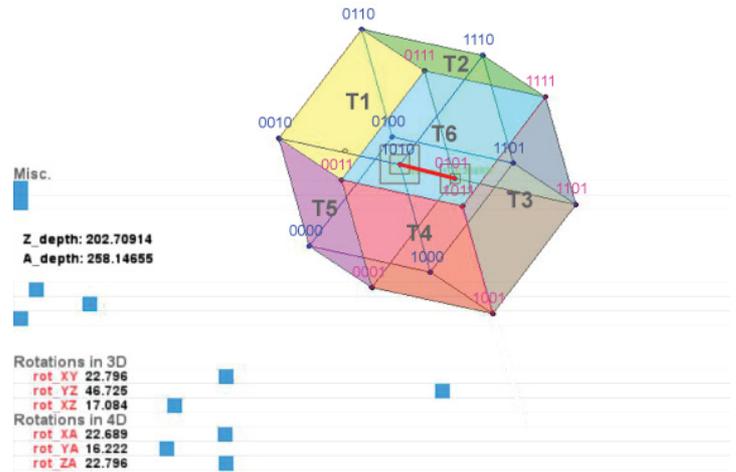


fig. 8q 4d hypercube and its corresponding clipping plane positions subject to a set of rotations. The red line is the depth vector

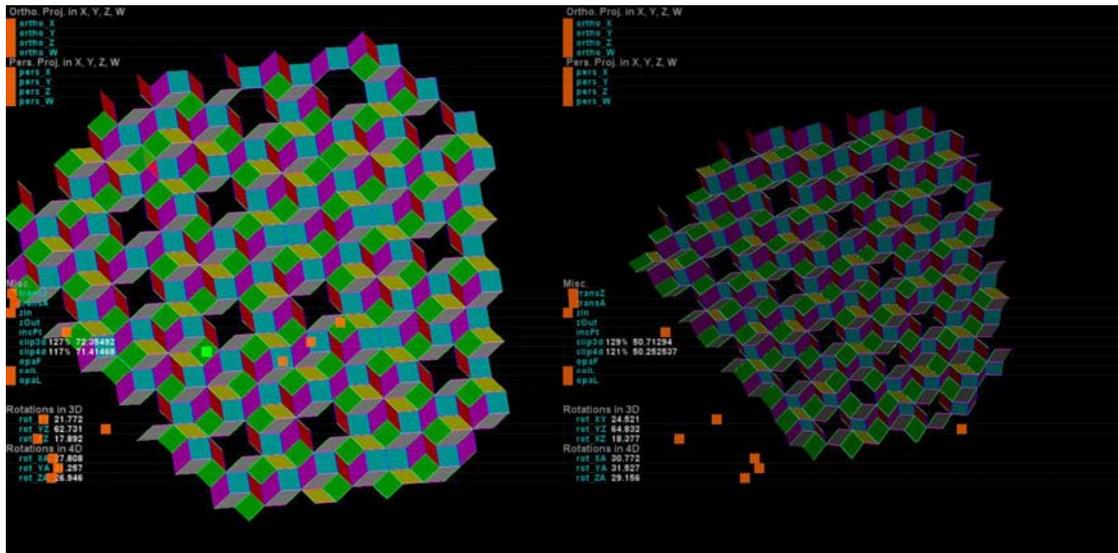


fig. 8p 2d and 3d projection of 4d tiling using the 'square' method

From the 'square' method, a more or less complete tiling can still emerge by manually incrementing the clipping planes (fudging) on top of a finely tuned set of rotation angles. Figure 8p gives an indication as to how far the clipping planes in this instance had to be widened from the originally calculated z and a_depth vector to include the missing points of the tiling; 129% for the former and 121% for the later. This percentage can vary greatly depending on rotation angles and the tiling search process can become painstaking.

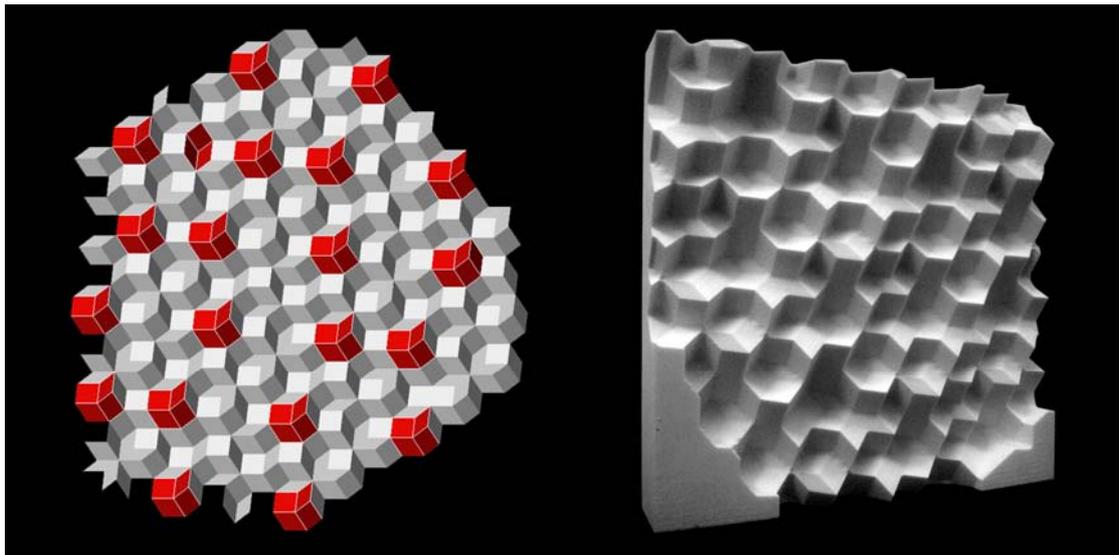


fig. 8q digital and physical (CNC milled) model of a 3d projection of 4d tiling using the 'square' method

It emerged that a 4d tiling is composed of six tiles or less if some of those tiles become identical under specific rotations (rotational symmetry). Special cases of tiling have been widely documented (Penrose tiling) and do not constitute the focus of this research. In figure 8q, in order to complete the tiling the holes were filled (in red) with some of the 6 existing tiles from the 3 dimensional tiling.

To prove that the 'circular' method was potentially the correct one, a comparative test was carried out in three dimensions where it is easier to visualize what is at work. A cubic grid of points in three dimensions was directly projected onto a one dimensional line (3d to 1d) using each method separately and jumping down two dimensions at once. The diagrams below explain each method and their differences.

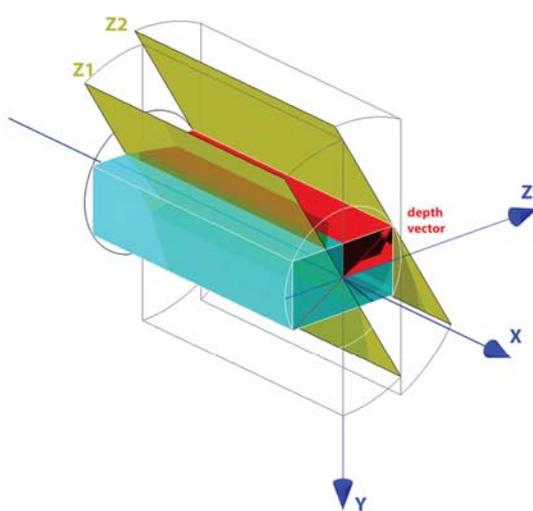


fig. 8r diagram of the square method to define the clipping boundary for a 1d projection (line on the x-axis) of a cubic grid of points.

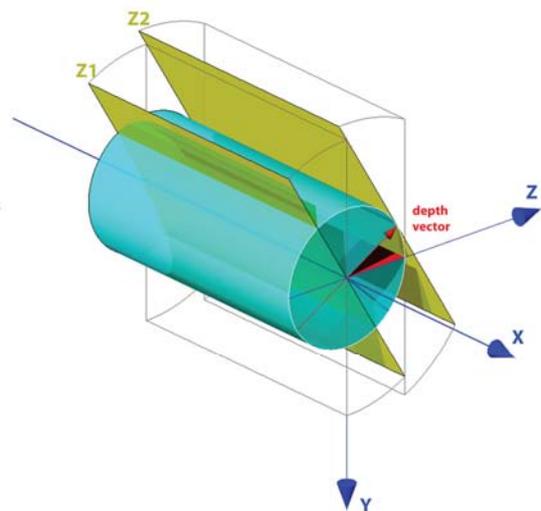


fig. 8s diagram of the circular method to define the clipping boundary for a 1d projection (line on the x-axis) of a cubic grid of points.

Closer the depth vector (red arrow) is to the y-axis or the z-axis in the 'square' method, smaller the clipping boundary becomes (the square is literally compressed towards a line). The gray rectangles are the initial clipping planes for a y-z rotation of 0 degree. The yellow planes are the same clipping planes for y-z rotation of 330 degree (z1, z2). The clipping plane located at the origin (z1) is actually the viewing plane for the interface.

The findings illustrated in figure 8t and 8u are quite self explanatory. These show that only the 'circular' method makes a complete tiling. Should it work as well in jumping two dimensions from higher dimensional grids, this would solve the overall problem.

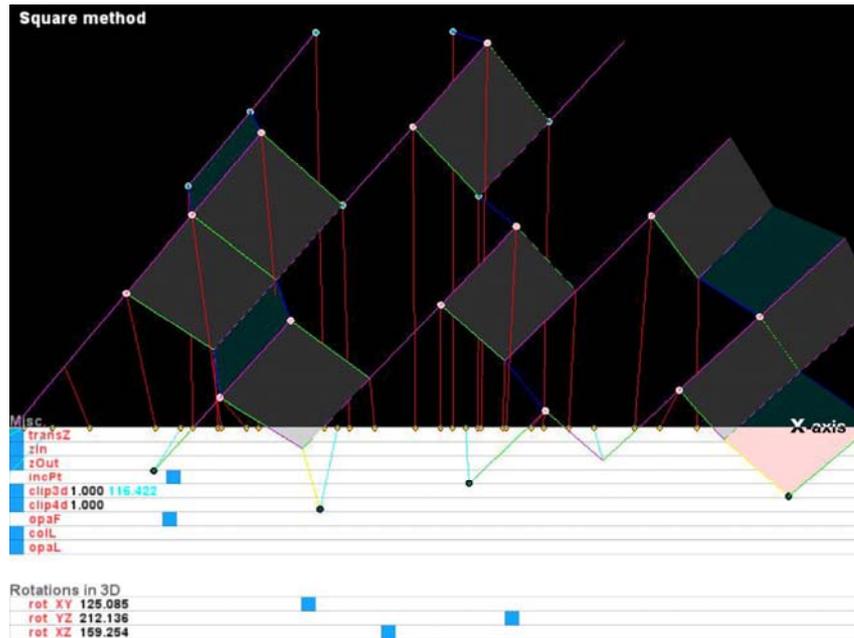


fig. 8t 'square' method from 3d to 1d with missing points in 3d

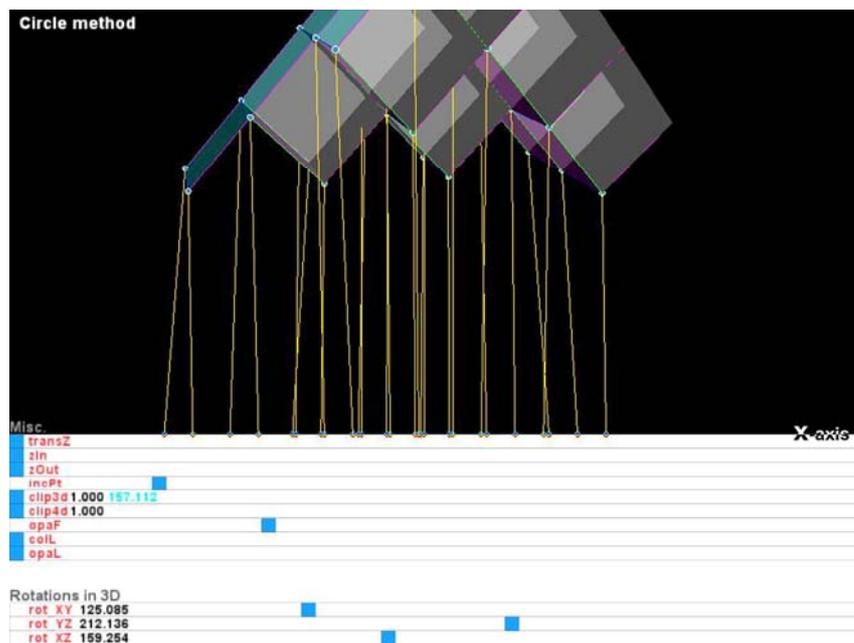


fig. 8u 'circular' method with equal amount of matching points in 3d and 1d

It is worth noting that the existing *literature** on the subject is somewhat ambiguous about the description of the clipping offset for higher dimensions. It often presents the projection method with the aid of a 2d to 1d projection diagram (as in fig. 8e) and then stipulates that the same concept applies for higher dimensions, yet not quite. If clipping occurs indeed within an area between two lines from 2d to 1d, from 3d to 1d that area is actually delimited by a rotating line in 3d and not by a linear offset between two planes. A hyperplane is here a 2d plane wrapped around (3d line rotated) to form a cylinder.

The working 'circular' projection method for 4 dimensional grids

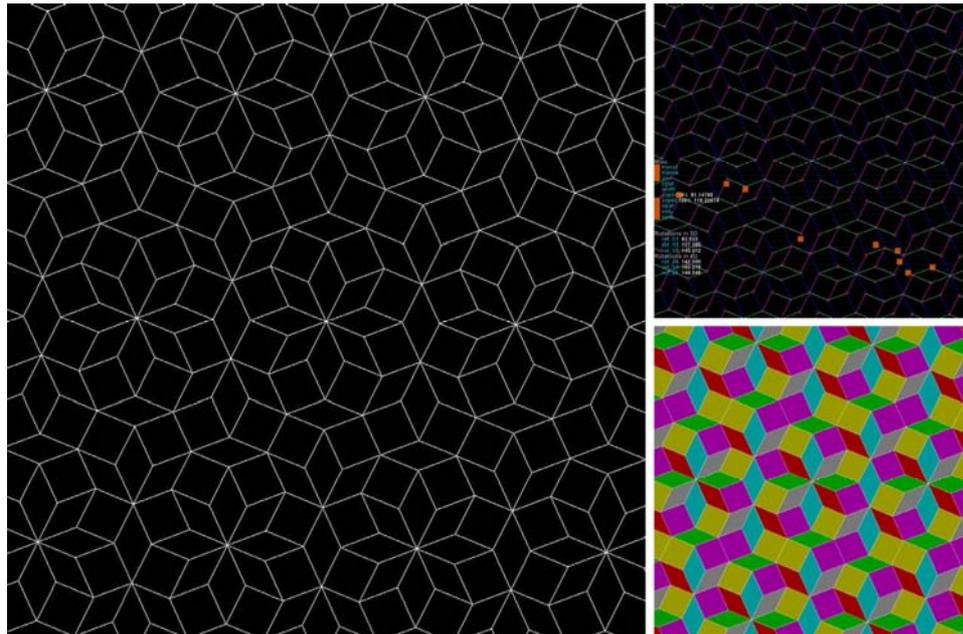


fig. 8v example of a 2d tiling using the 'circular' projection method from 4d (6tiles)

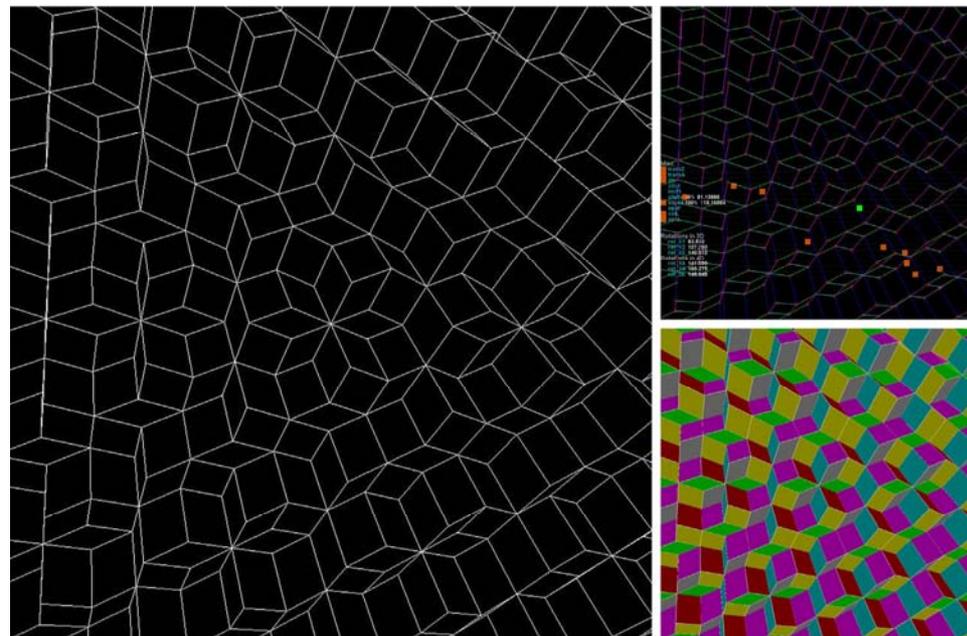


fig. 8w same example but for a 3d tiling using the 'circular' projection method from 4d (6tiles)

* Among others: <http://www.geom.uiuc.edu/apps/quasitiler/>

8.1.6 Jumping down more than two dimensions

From 5d to 2d (x, y, z, a, b)

The same approach used for four dimensional grids was successfully tested for five dimensional grids while this time jumping down three dimensions, disclosing a maximum of 10 different tiles and employing ten possible rotations (around x-y, y-z, x-z, x-a, y-a, z-a, x-b, y-b, z-b, a-b plane).

Pseudocode

If (the square (z-value of point X) + (the square (a-value of point X) + (the square (b-value of point X)) \leq ((square (z-value) + square (a-value) + square (b-value))*0.25) of the depth vector ($z1 \text{ plane} < (sq(X.z) + sq(X.a) + sq(X.b))*0.25 < z2 \text{ plane}$)

Then point X is inside the clipping boundary and is selected before being projected.

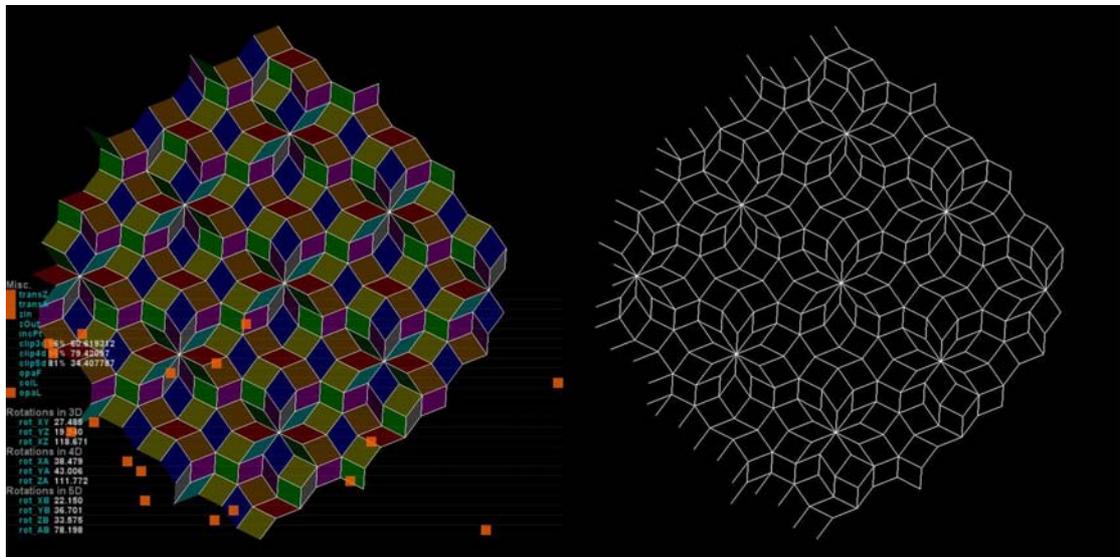


fig. 8x example of a tiling from a 5d grids projected in 2d (x, y, z, a, b)

From 5d to 3d

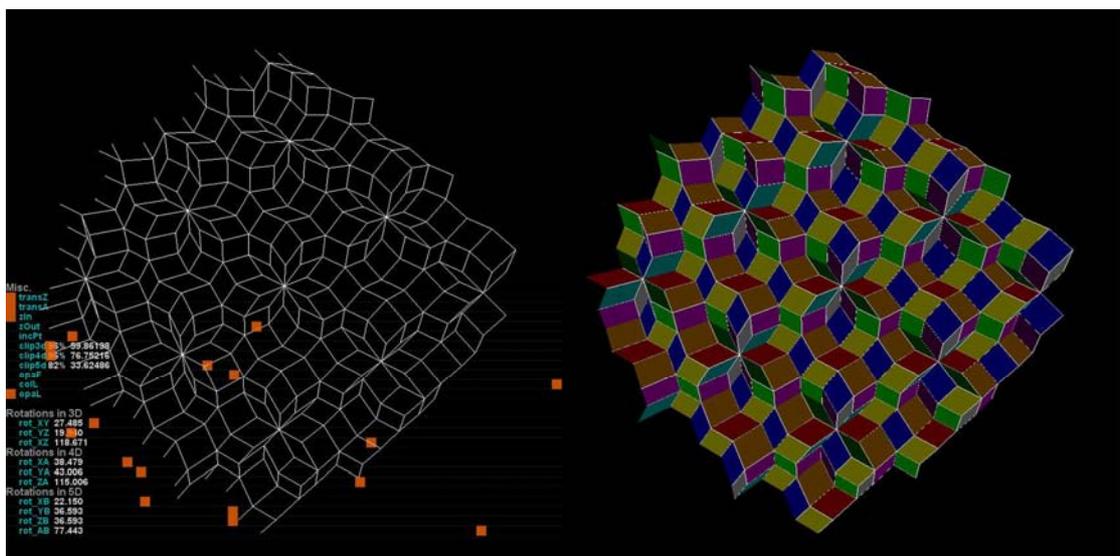


fig. 8y example of a tiling from a 5d grids projected in 3d (x, y, z, a, b)

From 6d to 2d and from 6d to 3d (x, y, z, a, b, c), see case study #5, p.32.
15 different tiles, 15 possible rotations (around x-y, y-z, x-z, x-a, y-a, z-a, x-b, y-b, z-b, a-b, x-c, y-c, z-c, a-c, b-c plane).

From 7d to 2d and from 7d to 3d (x, y, z, a, b, c, d).

21 different tiles, 21 possible rotations (around x-y, y-z, x-z, x-a, y-a, z-a, x-b, y-b, z-b, a-b, x-c, y-c, z-c, a-c, b-c, x-d, y-d, z-d, a-d, b-d, c-d plane).

8.2 Adjacency analysis

A further incentive for this research was to employ those tilings and their organizational structure and apply other types of geometries to them. This also stresses that the investigation was less concerned about the aesthetic values of the found tilings but more about their inherent assembly rules by which very few tiles could be repeated to form a field of a long range order and with a certain amount of specificities.

The type of geometry that was experimented with were *NURBS** surfaces. Each NURBS surface corresponded to a specific tile (6 types in 4d, 10 types in 5d, etc) and was delimited by the tile's edges. In order to constitute a continuous field and maintain proper tangencies with its neighbouring tiles, a specific NURBS surface had to know which edges it shared with which other tile(s) at other locations within the field.

This is precisely what the next case studies (#1, #2, #3, etc) set to establish. For any given set of tiles and their inherent organisation, how many *degree(s) of freedom* does the tiling have? Or in other words how many different line types at the tiles' edges can be individually manipulated while still providing for a fluent propagation of tiles. The example below shows how the initial appearance of a tiling's structure can be considerably altered under the new geometry and become much less discernable even though the same structure still governs the organization and repetition of the (NURBS) tiles.

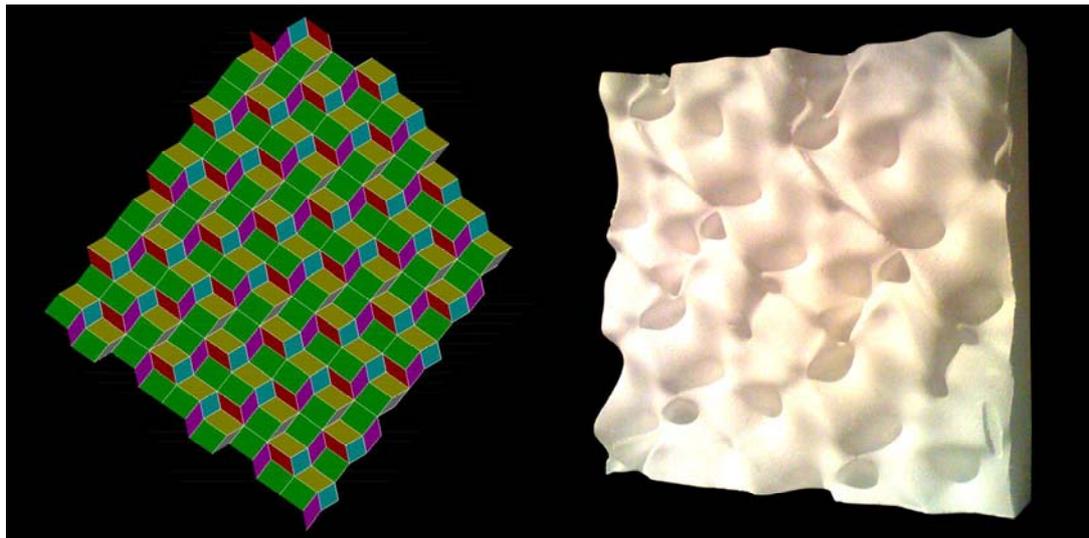
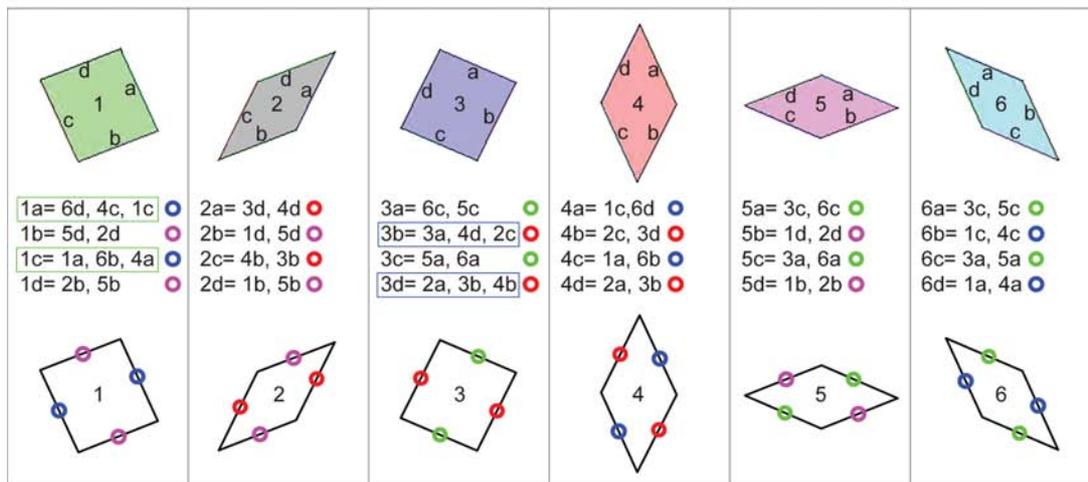
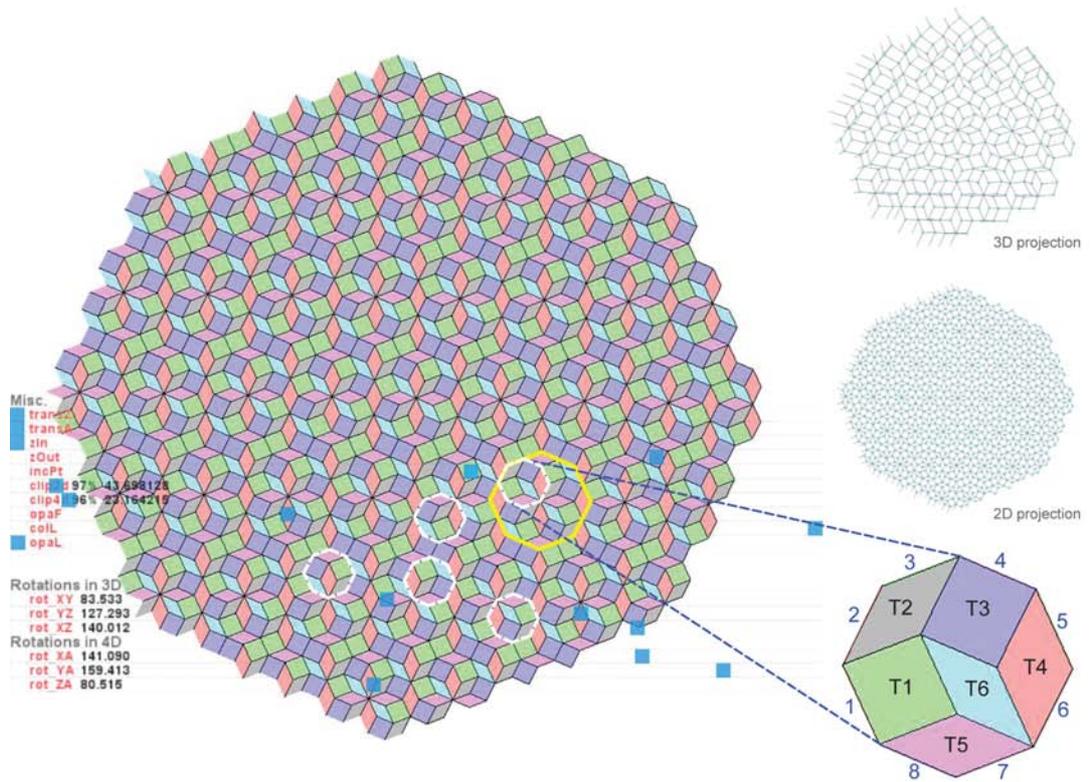


fig. 8za example of a clearly defined 4d tiling

fig. 8zb same tiling's structure but with NURBS surfaces
(physical milled model)

* *NURBS* stands for Non-Uniform Rational B-Spline.

Case study #1 (4d tiling same as figure 8v)

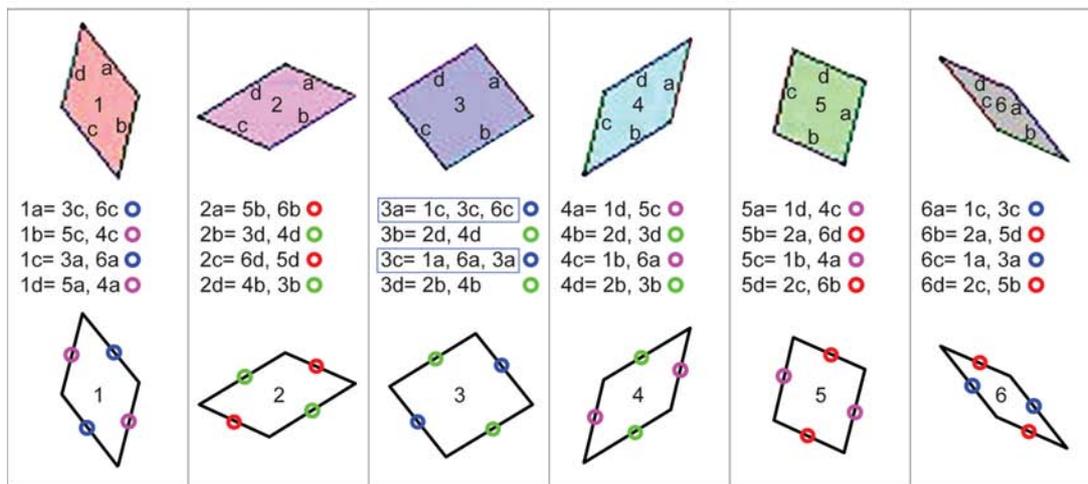
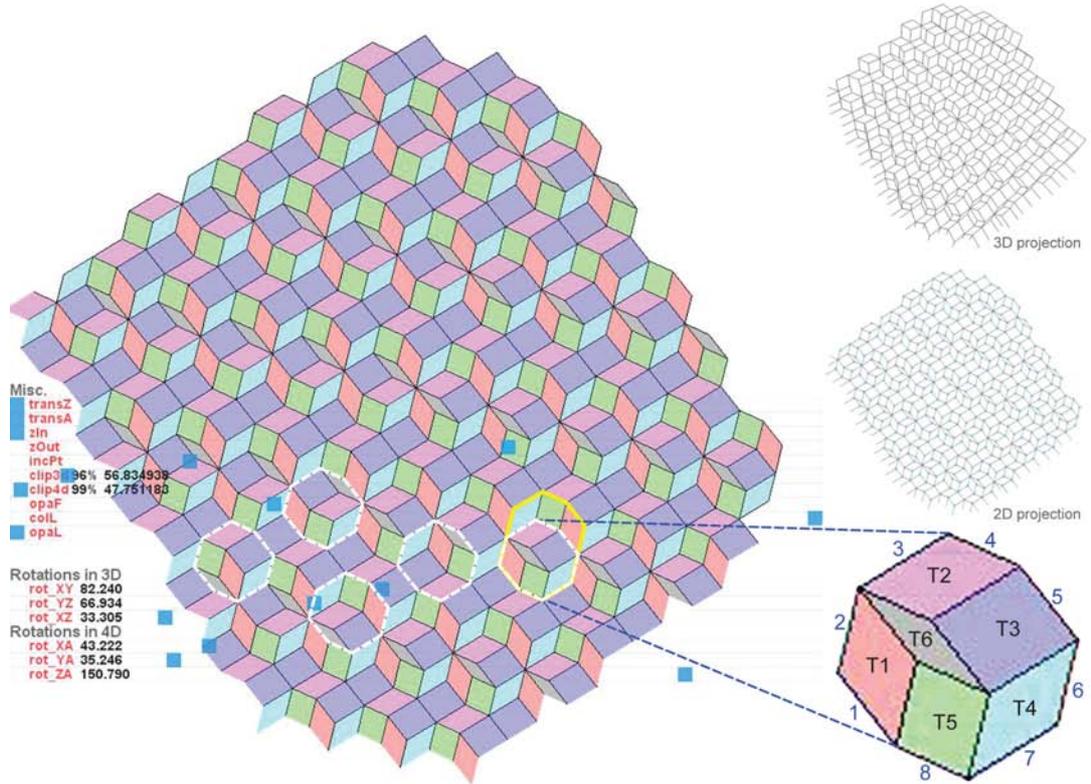


- Line type 1: 1a, 1c, 4a, 4c, 6b, 6d
- Line type 2: 1b, 1d, 2b, 2d, 5b, 5d
- Line type 3: 3a, 3c, 5a, 5c, 6a, 6c
- Line type 4: 2a, 2c, 3b, 3d, 4b, 4d

This first adjacency analysis unveiled 4 potentially different line types that would be explored in more details in the next section. The white lines compounds in the figure above are showing alternative configurations for the 6 tiles (rotational symmetries within the same group of tiles) and therefore already highlight that a specific edge will have more than one neighbouring edge in common.

Finally, edge 1a and 1c (in rectangle above) have an extra edge in common because in the overall tiling two tiles of type 1 are repeated next to one another causing 1a to be equal to 1c (the same occurs with edge 3b and 3d).

Case study #2 (alternative 4d tiling for comparison with one (blue tile) double only)

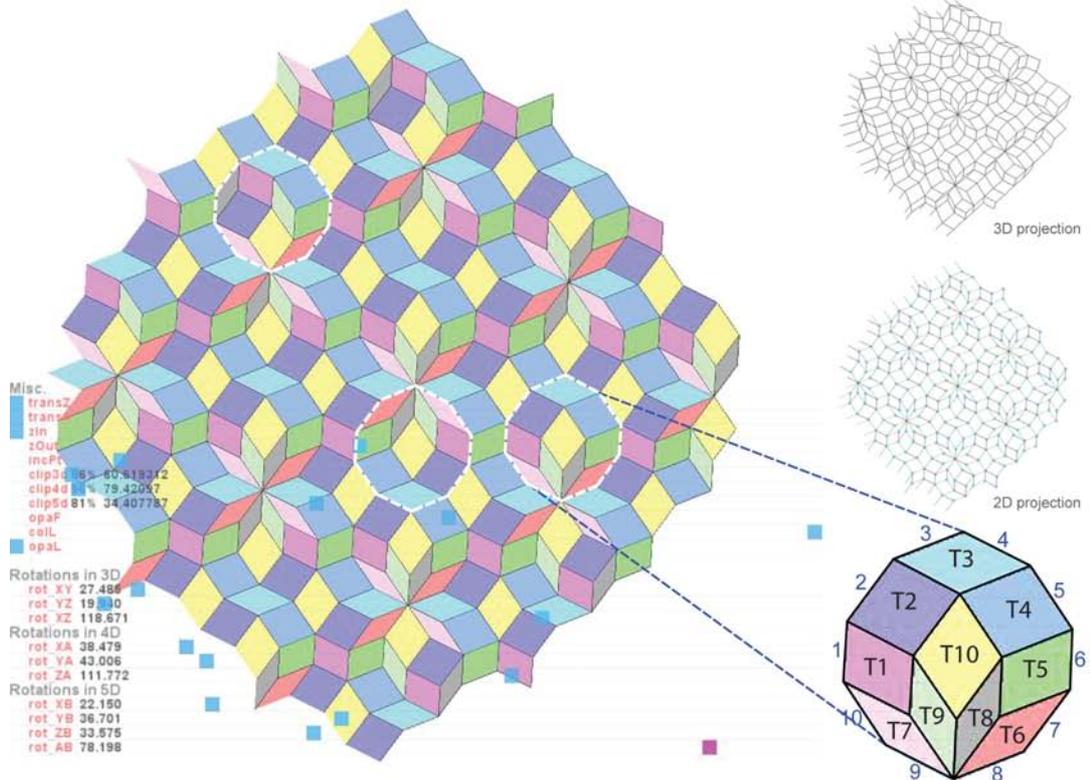


- Line type 1: 1a, 1c, 3a, 3c, 6a, 6c
- Line type 2: 1b, 1d, 4a, 4c, 5a, 5c
- Line type 3: 2b, 2d, 3b, 3d, 4b, 4d
- Line type 4: 2a, 2c, 5b, 5d, 6b, 6d

In this example above, only edge 3a and 3c have more than 2 corresponding edges (3 in total) because tile #3 is adjacently doubled in places in the tiling. For a 4d tiling that has no double tiles left, every edge (24 in total) would have

a corresponding 2 edges in common and 4 different line types of 6 edges each (6 edges x 4 line types= 24 edges).

Case study #3 (5d tiling same as figure 8x)

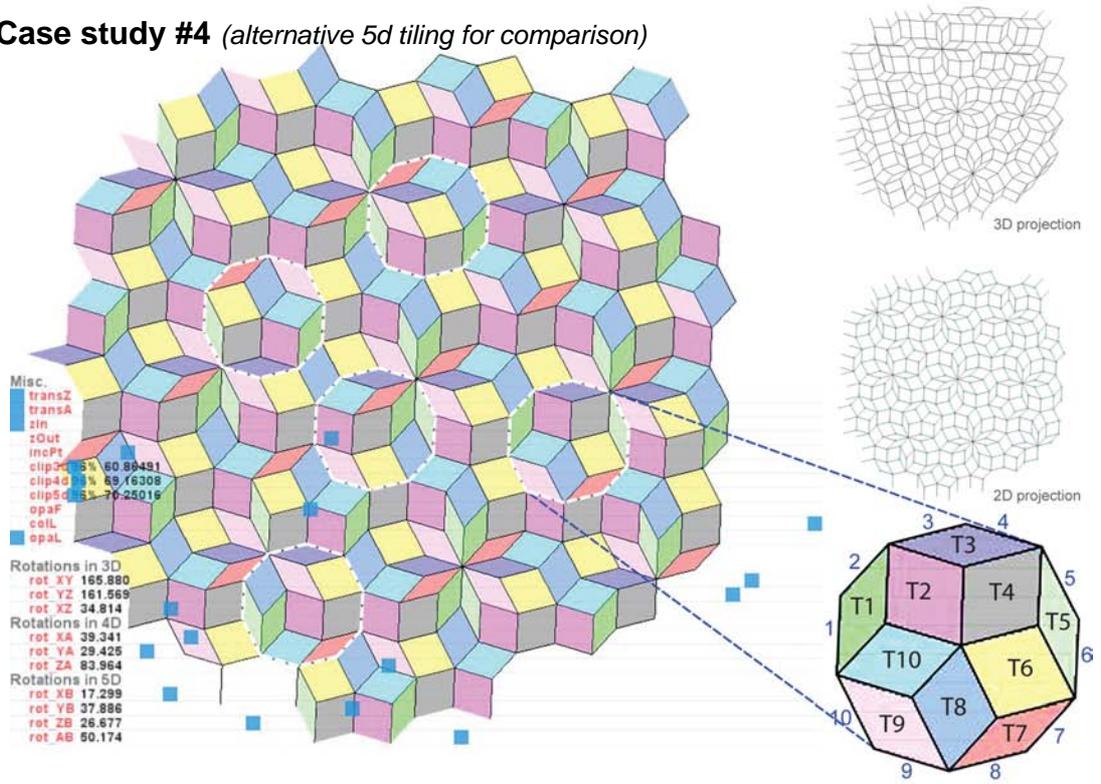


<p>1a= 2c, 7c, 3c 1b= 9d, 5c, 8c 1c= 7a, 2a, 3a 1d= 5a, 9b, 8a</p>	<p>2a= 1c, 3c, 7c 2b= 10d, 6d, 8d 2c= 1a, 3a, 7a 2d= 10b, 6b, 8b</p>	<p>3a= 1c, 2c, 7c 3b= 4d, 6a 3c= 1a, 2a, 7a 3d= 4b, 6c</p>	<p>4a= 7d, 10c, 9c 4b= 3d, 5d 4c= 10a, 7b, 9a 4d= 3b, 5b</p>	<p>5a= 1d, 9d, 8c 5b= 4d, 6a 5c= 1b, 8a, 9b 5d= 4b, 6c</p>
<p>6a= 3b, 5b 6b= 2d, 10d, 8d 6c= 3d, 5d 6d= 10b, 8b, 2b</p>	<p>7a= 1c, 2c, 3c 7b= 4c, 9c, 10c 7c= 1a, 2a, 3a 7d= 4a, 9a, 10a</p>	<p>8a= 1d, 5c, 9d 8b= 2d, 6d, 10d 8c= 1b, 9b, 5a 8d= 2b, 6b, 10b</p>	<p>9a= 4c, 7d, 10c 9b= 1d, 5c, 8c 9c= 4a, 7b, 10a 9d= 1b, 8a, 5a</p>	<p>10a= 4c, 7d, 9c 10b= 2d, 6d, 8d 10c= 4a, 7b, 9a 10d= 2b, 8b, 6b</p>

- Line type 1: 1a, 1c, 2a, 2c, 3a, 3c, 7a, 7c
- Line type 2: 1b, 1d, 5a, 5c, 8a, 8c, 9b, 9d
- Line type 3: 2b, 2d, 6b, 6d, 8b, 8d, 10b, 10d
- Line type 4: 3b, 4d, 5b, 6a
- Line type 5: 3d, 4b, 5d, 6c
- Line type 6: 4a, 4c, 7b, 7d, 9a, 9c, 10a, 10c

A 5d tiling has a maximum of 40 possible edges (10 tiles x 4 edges).
 In this specific case study 6 possible line types were encountered ((4 x 8 edges) + (2 x 4 edges)) = 40 edges.

Case study #4 (alternative 5d tiling for comparison)



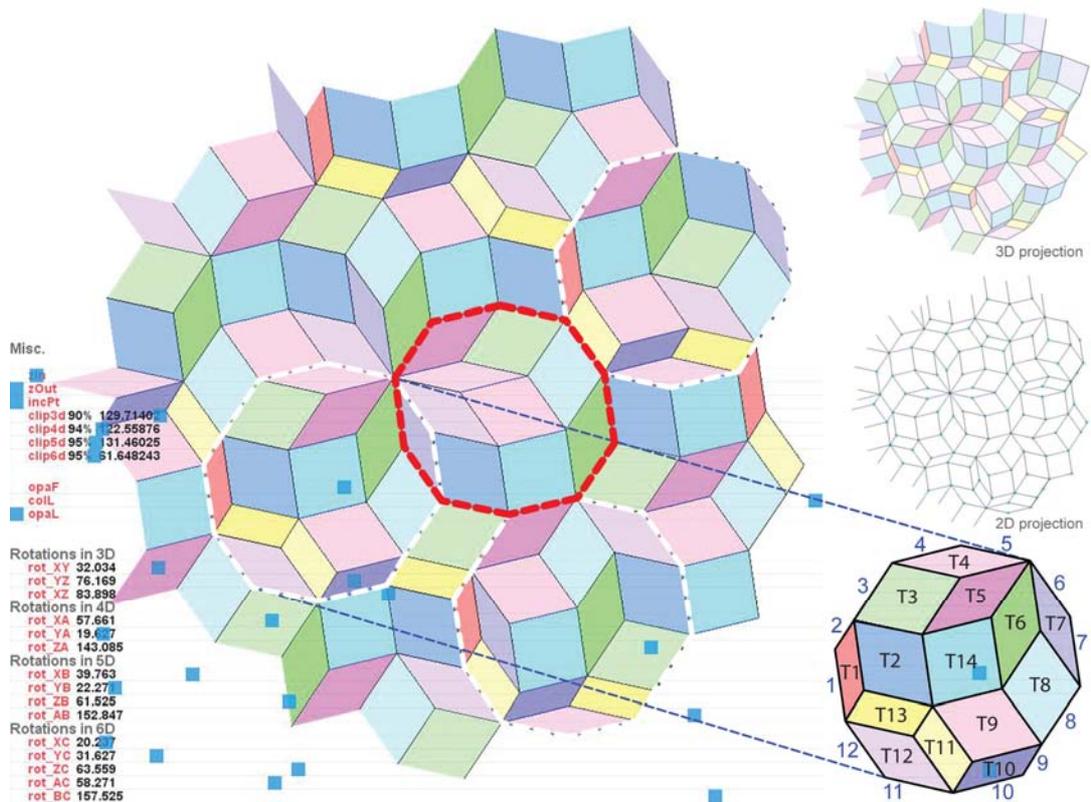
<p>1a= 2d, 4c, 5d 1b= 10d, 7c, 8d 1c= 5b, 4a, 2b 1d= 8b, 7a, 10b</p>	<p>2a= 3c, 10c, 9b 2b= 1c, 4c, 5d 2c= 10a, 9d, 3a 2d= 1a, 7a, 10b</p>	<p>3a= 2c, 10c, 9b 3b= 4d, 6d, 7d 3c= 10a, 9d, 3a 3d= 6b, 7b, 4b</p>	<p>4a= 1c, 2d, 4d 4b= 3d, 6d, 7d 4c= 1a, 2b, 5b 4d= 3b, 7b, 6b</p>	<p>5a= 9c, 8c, 6c 5b= 1c, 2d, 4c 5c= 6a, 9a, 8a 5d= 1a, 2b, 4a</p>
<p>6a= 5c, 8c, 9c 6b= 3d, 4d, 7d 6c= 5a, 8a, 9a 6d= 3b, 4b, 7b</p>	<p>7a= 1d, 10d, 8d 7b= 3d, 4d, 6d 7c= 1b, 8d, 10b 7d= 3b, 4b, 6b</p>	<p>8a= 5c, 6c, 9c 8b= 1d, 7c, 10d 8c= 5a, 6a, 9a 8d= 1b, 7a, 10b</p>	<p>9a= 5c, 6c, 8c 9b= 3a, 2a, 10a 9c= 5a, 6a, 8a 9d= 2c, 3c, 10c</p>	<p>10a= 2c, 3c, 9b 10b= 1d, 7c, 8d 10c= 2a, 3a, 9d 10d= 1b, 7a, 8b</p>

- Line type 1: 1a, 1c, 2b, 2d, 4a, 4c, 5b, 5d
- Line type 2: 1b, 1d, 7a, 7c, 8b, 8d, 10b, 10d
- Line type 3: 2a, 2c, 3a, 3c, 9b, 9d, 10c, 10d
- Line type 4: 3b, 3d, 4b, 4d, 6b, 6d, 7b, 7d
- Line type 5: 5a, 5c, 6a, 6c, 8a, 8c, 9a, 9c

This case study is very close to the Penrose tiling

In the example above, 5 different line types were categorized (5 x 8 edges each = 40 edges in total).

Case study #5 (6d hybrid tiling)



This 6d case study disclosed just 14 tiles instead of the expected 15 tiles. Firstly, it does still perform as an aperiodic tiling as well as its adjacency analysis. The actual ‘missing’ tile is there but has been reduced to a line overlapping with an existing tile’s edge.

The fact that rotations are articulating the emerging tiling, by rotating in space selected vertices within a specific region, does not guaranty that for a higher dimension all the concerned vertices will be accordingly activated to make up for the correct amount of tiles. A 5d type of tiling can be created within a 6d environment even though the 15 rotations are all activated. More crucially, arbitrary rotations offered the possibility for hybrid tilings to emerge. The ones comprised with moments of 5d, 4d, 3d and/or 2d tile projections within the same tiling. The red (contour) line above shows a 5d assortment (10 tiles) inside the 6d tiling.

Hybrid tilings would prove to be spatially richer in the next experiments. Lines linking vertices behave somewhat like ‘rubber bands’ within a dimension and across dimensions. The default setup is when all rotations are at zero degree angles and when for any dimensions the selected vertices are all overlapping on top of one another, which perception from one of its clipping plane is a simple 2d grid. Appendix B presents the geometry making up faces from vertices in and across dimensions.

It is now safe to assume that for a tiling of n dimensions; a minimum of n different line types are possible.

In 4d: for a max. of 6 repeated tiles, that is 24 edges: a min. of 4 line types of 6 edges each

In 5d: for a max. of 10 repeated tiles, that is 40 edges: a min. of 5 line types of 8 edges each

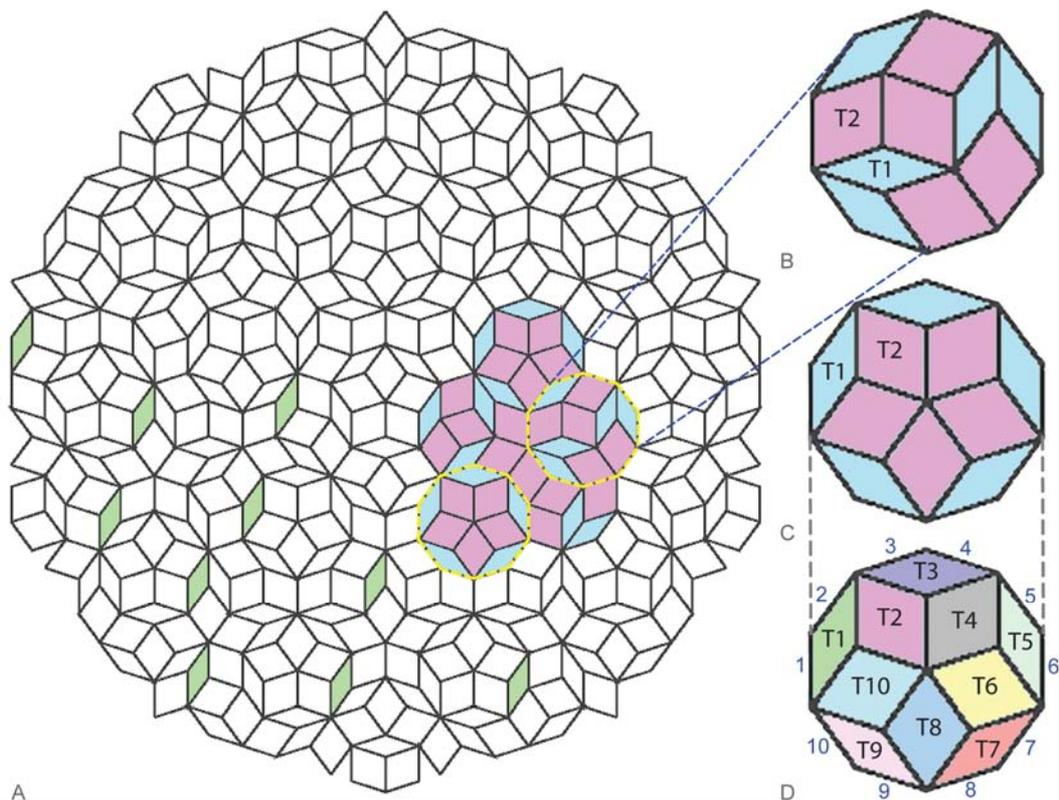
In 6d: for a max. of 15 repeated tiles, that is 60 edges: a min. of 6 line types of 10 edges each

In 7d: for a max. of 21 repeated tiles, that is 84 edges: a min. of 7 line types of 12 edges each
Etc.

Back to the Penrose

Case study #6 (5d tiling)

In 5d: min. of 2 repeated tiles, that is 40 edges: a min. of 5 line types of 8 edges each?



The Penrose tiling above can be generated by the 2d projection of 5-dimensional grid. It is aperiodic and made of only two different rhombi (fig 6A, 6B). As such the adjacency analysis would require all the edges to have the same line type in order to mass produce two types of tile only. One line type for the overall tiling is very efficient but does limit the amount of irregularity another geometry could have when adapted to the adjacency rules (see next section). Alternatively it can also be thought as a 5d tiling comprised of a maximum of 10 different tiles as in fig. 6D, even though as a 2d shape, it actually portrays as 5x2 different tiles. This carries the possibility of 5 different line types for 10 tiles to make up for a greater diversity in a field formation from the exact same tiling, similar to case study #4. As 2 tiles, they have to be rotated to make the appropriate tiling, but as 10 tiles, they are just translated to make the same tiling (no rotations in individual tiles, only as compounds of 10 tiles; see how the green tiles above have the same orientation). This distinction would become useful in a later experiment (p.40).

5. Adaptation

This chapter is in some sort a continuation of the methodology pursued previously. Nonetheless a separation wanted to be drawn to better distinguish the findings (tiling from the projection method) against how their organizational effectiveness can be further tested past a tiling graphical arrest. This section also instigated the discussion part.

Towards the Quasi Roller

Once the organization of the edges for a tiling had been determined (i.e. case studies carried out), a line type, accounting for a group of edges distributed throughout the field, became its own being. This meant each line type had the potential to be manipulated individually while readjusting the overall field. This part of the process was tested in generative component, where each line type was first programmed parametrically and then mapped appropriately onto the field. In doing so, any changes made locally (on a line type itself) would instantly altered and update the overall tiling. Appendix D1 shows examples of different graph variables created to parametrically alter a 4d tiling.

NURBS line type

A primary experiment was carried out by changing the Z-values of the line types in two locations (yellow dots in fig. 5a) along the edges of the tiles in order to create a NURBS curve in elevation (a sinusoidal-like curve). Figure 5a and 5b describe the geometry employed. The tiling used is a 2d projection of a 4 dimensional lattice (for its adjacency analysis see appendix C).

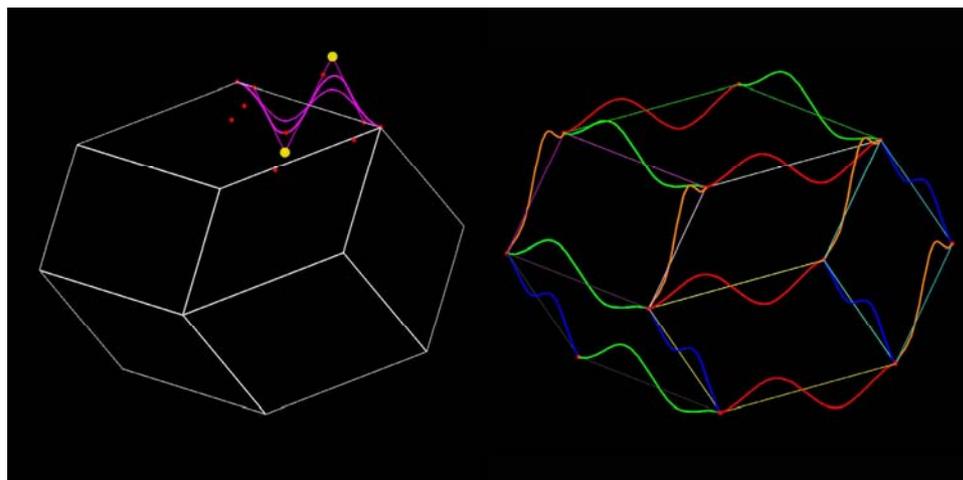


fig. 5a axonometric view showing a 2nd, 3rd, 4th degrees order of NURBS curves (a linetype) along a tile's edge.

fig. 5b axonometric showing the 4 different linetypes (green, red, orange, blue) from adjacency analysis onto a combination of 6 tiles.

Every tile is rhombic (4 edges) but have different line types making up a NURBS surface from the initial tile (*fig. 5c shows 2 types of NURBS curves for a single tile*).

Tangencies

The construction of a Nurbs tile alone does not account for how it must communicate with its neighbouring tiles. In fact, for a NURBS surface to smoothly translate into its neighbours, tangencies had to be resolved so that creases could be avoided along the tiles' edges. Tangential offsets were created away from the surfaces' respective common edges, onto which a series of points were added to become part of the overall definitions of the surfaces and act as tangency points. These offsets are represented in the two figures below by dotted lines (*fig. 5c, 5d*). Each point along a dotted line has a corresponding point perpendicular to a tile's edge and belonging to the other surface sharing the same edge. Figure 5e best illustrates this geometrical model in plan view.

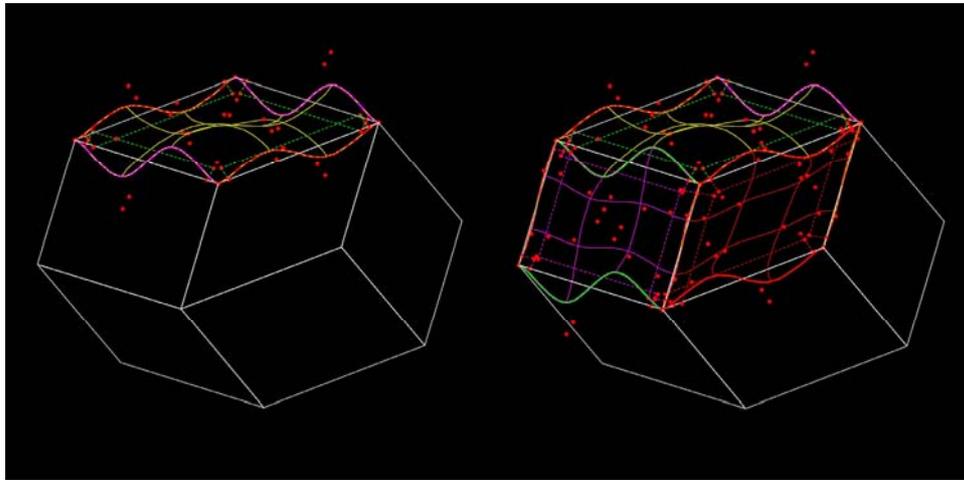


fig. 5c axonometric view showing a Nurbs surface from 4 Nurbs curves located onto the tile's edges

fig. 5d axonometric view showing the combination of 3 NURBS tiles with proper tangencies between them

One of the difficulties encountered was how good the tangencies performed near the vertices. Each vertex is a meeting point of 3 to 8 edges (in 4d) at different angles (obtuse, acute) across the field. Although each specific case could have been tailored, the key was to arrive to a general method which would apply anywhere in the tiling under any convergences of edges (p.37).

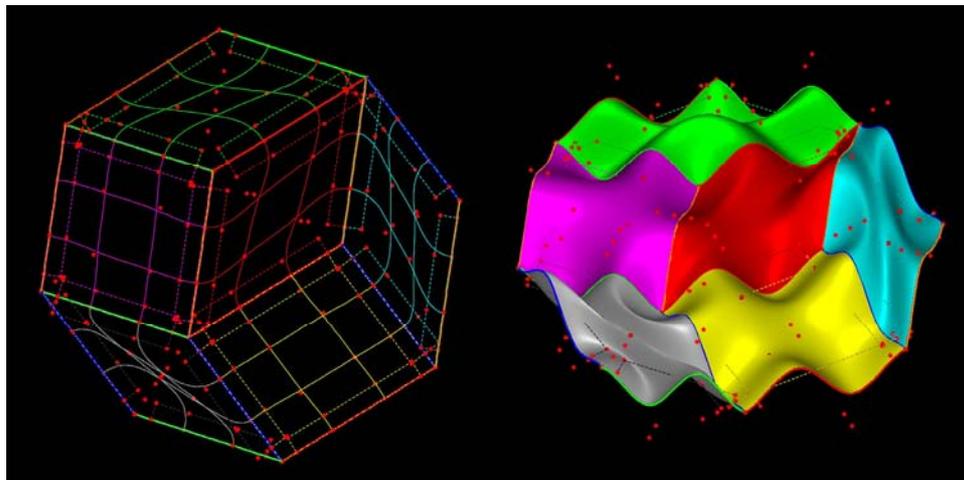


fig. 5e plan view of a combination of 6 tiles showing the NURBS curves are coincident with the tiles' edges and the offsets points perpendicular to the tiles' edges.

fig. 5f axonometric view of NURBS surfaces from changes in the curves' z-values.

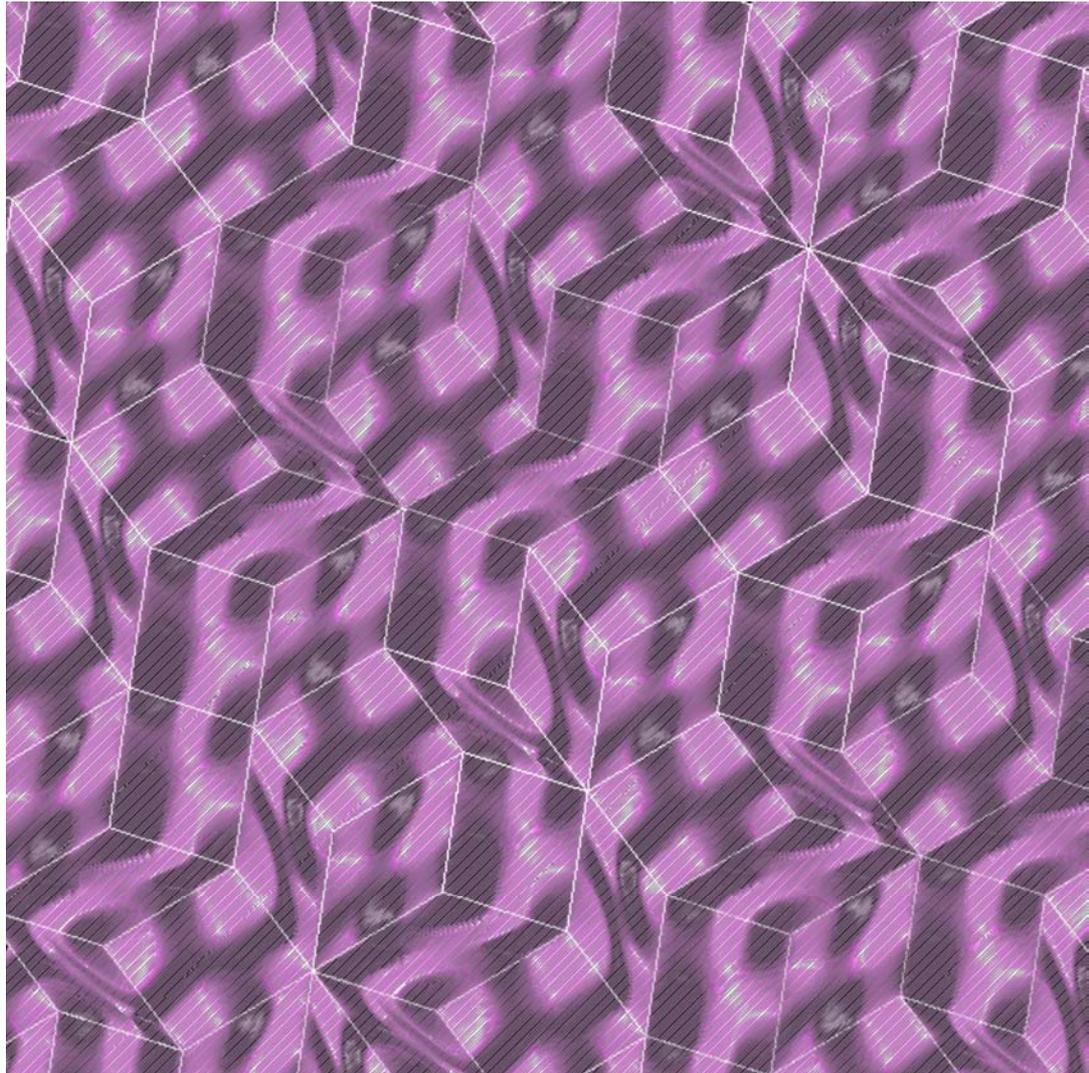


fig. 5g Plan view of a CAM simulation of NURBS aperiodic field of 6 tiles from the 2d projection of a 4d grid
 Gray lines are tool paths: 45° parallel finish, 6mm ball mill
 White lines show the structure of the original 4d tiling superimposed

The image above shows a field configuration from the example in figure 5f. Once the line types and its geometry have been tested for a combination of 6 tiles, they are then referenced as *components* to a larger field of tiles. This was scripted (in *generative components*, see appendix D2, p.63) by following the rules from the adjacencies analysis, shown previously. A corresponding physical model in appendix F2 manifested for the first time how the repetition of few tiles (6 in this case) could support a relatively irregular and complex pattern away from the tiles' immediate recognition.

Fleeing the tile's original shape

A further test was to dislocate the NURBS geometry from to the tiles' straight edges by allowing not only the Z-values of the curves but also the X and Y-values to vary. Simultaneously it was essential to ensure that suitable tangencies between the NURBS tiles were well kept (*fig. 5h, 5i*).

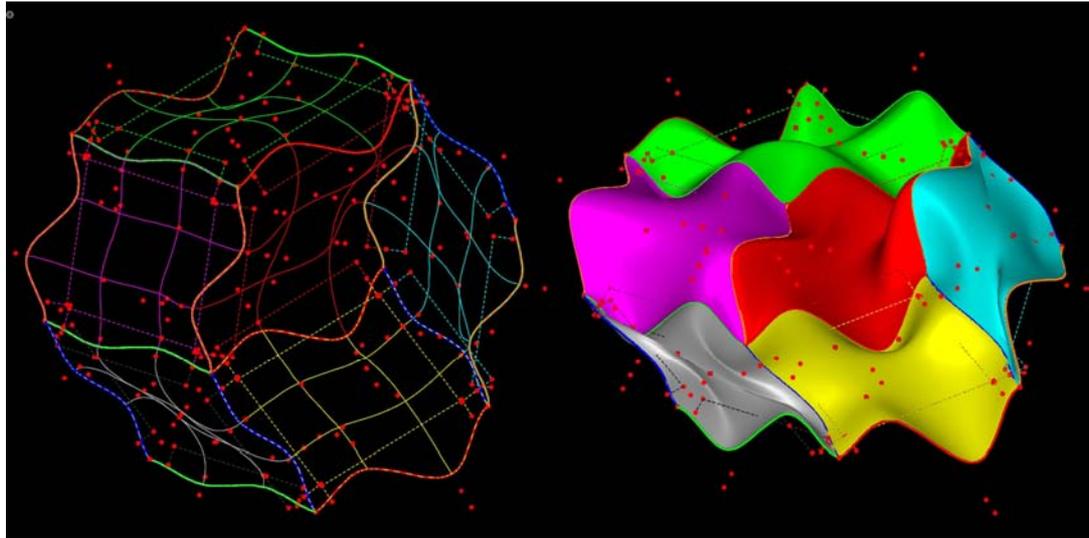


fig. 5h plan view of a combination of 6 tiles showing the overall geometry and how the NURBS line types are changes in x and y.

fig. 5i axonometric view of NURBS surfaces from changes in the curves' x, y and z-values.

In returning to the tangencies, the general method is a combination of two principles and was constructed as follow:

a) Around the vertices (red lines and points in fig. 5j)

Equidistant points from a vertex along each edge were defined from which transversal lines to each edge were drawn in order to create series of 2 tangential points belonging to the NURBS surfaces' definitions. This made sure a tangency between two NURBS surfaces near the vertices were properly matching.

b) Along the edges (yellow lines and points in fig. 5j)

Additionally in between vertices, series of equidistant points are defined on the tiles' edges to transversally draw lines on which offset points are created to also be part the NURBS surfaces so that proper tangencies between tiles are settled. The general method was later tested effectively on a 5d tiling (fig.5k).

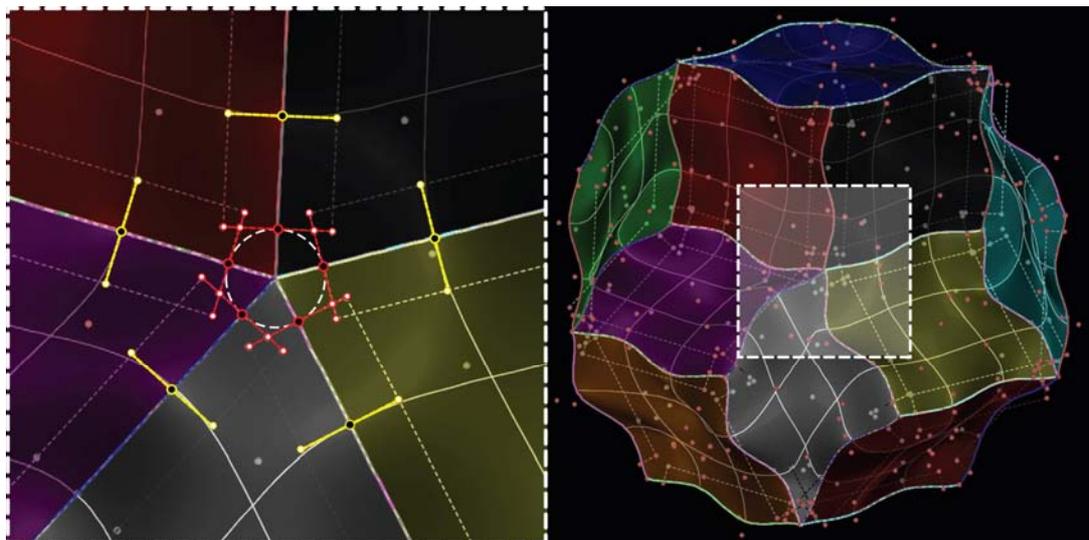


fig. 5j plan view detail of 5d NURBS tiling default setup for tangencies

fig. 5k plan view of 5d NURBS tiling where x, y and z-values of the different line types were altered

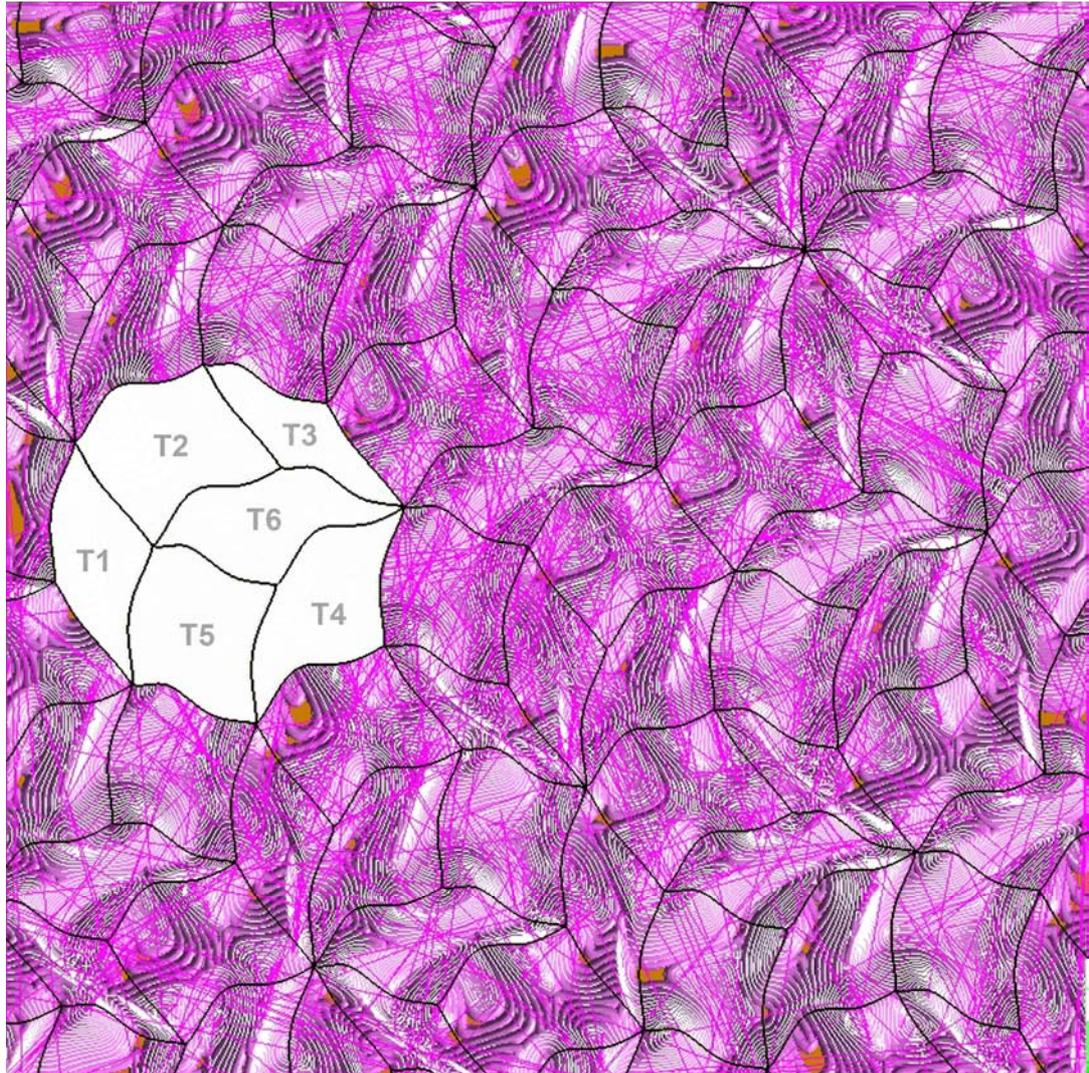


fig. 5l Plan view of a CAM simulation of NURBS aperiodic field of 6 tiles from the 2d projection of a 4d grid
 White lines show the tool paths between levels: horizontal finish, 6mm ball mill by plateaus of 3mm
 Magenta lines are the cut transfer paths)
 Black lines show the altered structure (x and y values) of the original 4d tiling superimposed
 See appendix F3 for the corresponding physical milled model

Porosity

Until now the two-sidedness of the field has been left latent, even though it always existed. One way the interrelation of the two sides (recto, verso) of the overall tiling can be enhanced is by allowing some voids throughout the field of tiles upon the same initial assembly rules. These voids work as vehicles between the two sides in settlement with their prior duality. A hole came about by the incursion of part of a bottom tile into a top tile. Their intersection is then voided to make an opening. Two families of holes were investigated; one around the middle of a tile (*fig. 5k*), the other around the corners of a tile (*fig. 5l*). A void outline emerged specifically characterized by the partial overlap of the bottom and top tiles NURBS geometry.

The figure 5m uses a tiling from a 2d projection of a 4d lattice, previously analysed (case study #7, appendix C). The porosity factor contributed in transforming what was previously a wall-type into a more dynamic screen.

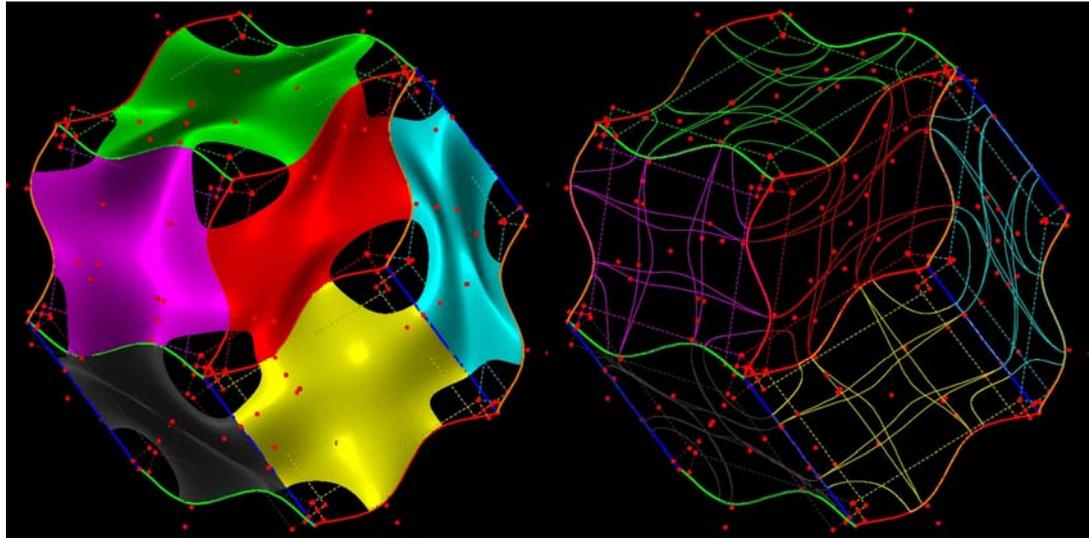


fig. 5m plan view of 6 tiles with holes at the vertices (tiles' corners)

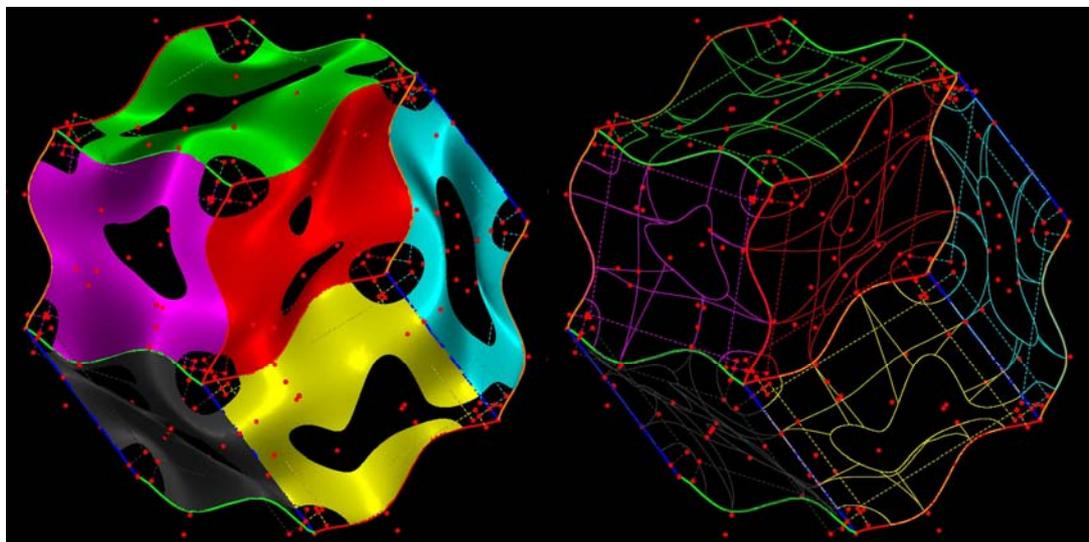


fig. 5n plan view of 6 tiles with holes at the vertices (tiles' corners) and in middles of the tiles

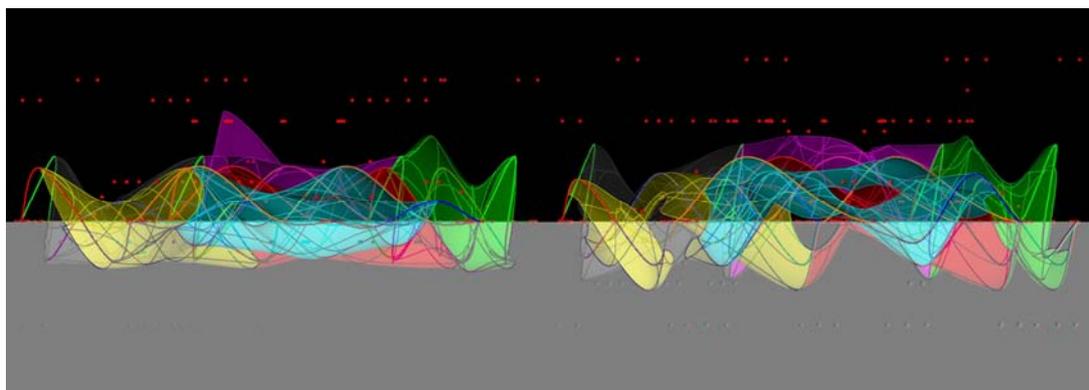


fig. 5o Elevations of fig. 5k and 5l showing the amplitude of the z-values for each tile
 The boundary line between the black and the gray zone defines the plane onto which all original vertices from the tiling are located (2d plane).

See appendix F4 for the corresponding physical two-part mould from a larger field of tiles

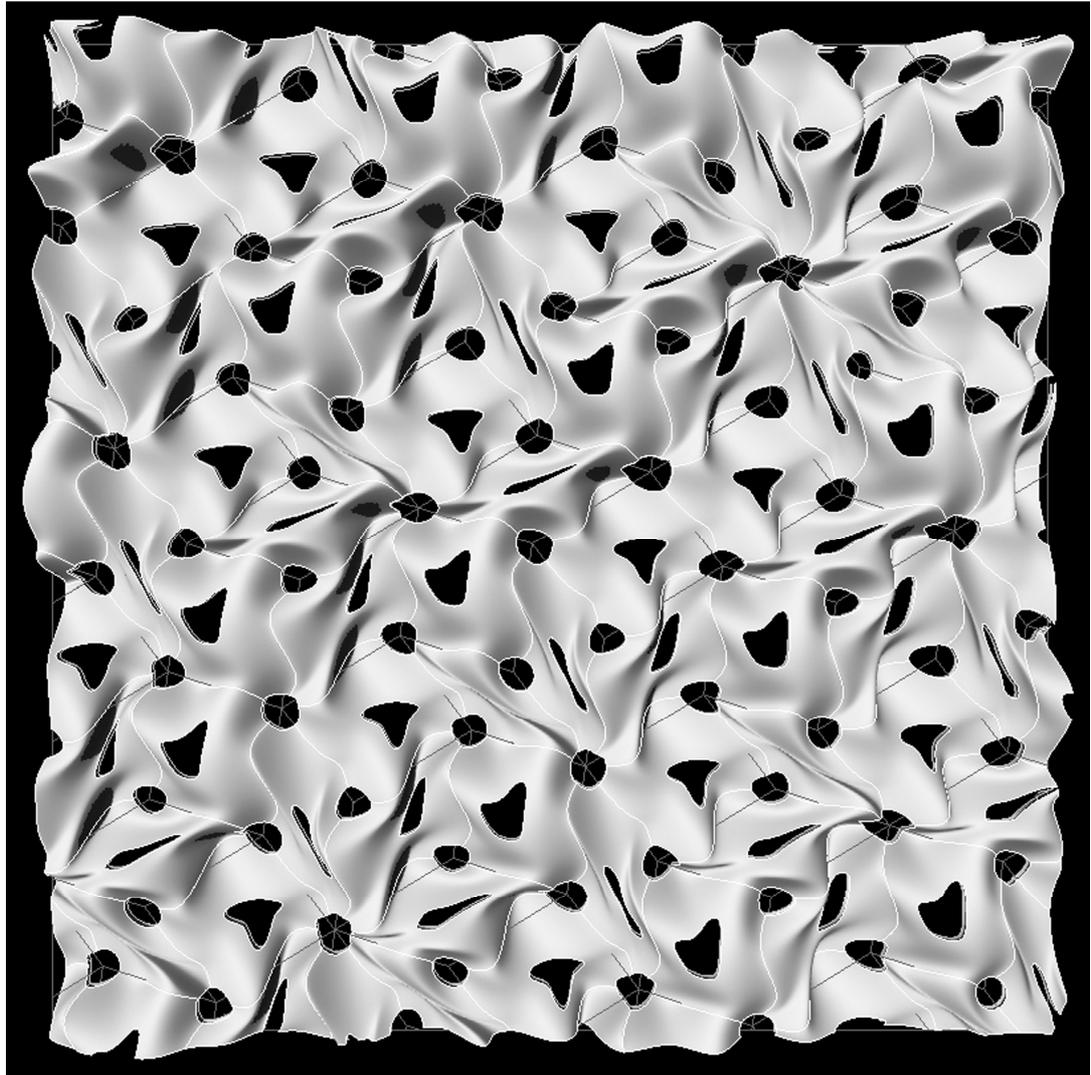


fig. 5o Top view of field configurations from 6 tiles (4d) with holes at vertices and at middle of tiles

Quasi roller

At last, an experiment was to liberate any tiling and the subsequent geometry from its projected plane (2d projection of vertices) and roll it into a cylinder (fig 5q). Since each individual tile (6 in 4d, 10 in 5d, etc) is repetitively translated and copy throughout the tiling, its relationship with regards to the curvature of the cylinder was not altered, that is a *quasi roller* could be built without increasing the amount of different tiles (still 6 tiles in 4d). None of the individual tile was in fact rotated anywhere else. Only groupings of 6 tiles (in 4d) had various types of configurations across the field. On the contrary, this efficiency of means would collapse in trying to map a sphere, a cone or any volume changing in more than one direction; each tile would indeed have to become individualized.

All the previous experiments can now be amplified from the new cylindrical configuration where interior spatially interacts with the exterior and vice versa (image on title page).



fig. 5q rolling of a 4d tiling into a cylinder with close match

The length of a *quasi cylinder* (along its rotational axis as well as transversal to it) for any n-dimensional tiling is limited for the same reason an aperiodic tiling is finite in space. Regions of translational symmetry are indeed finite by definition on such tilings. How long a cylinder could be made from a tiling was an exploration beyond the scope of this research but more time may come later to ponder further on the question.

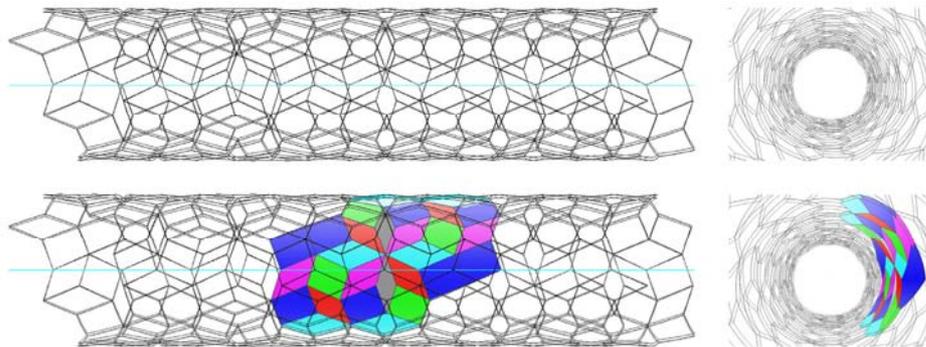


fig. 5r Elevations and interior views of a quasi roller from case study #7 (4d tiling) in appendix C

A rolling of a 5d (or more) tiling is as well effective. Nonetheless tiles need to be repeated across the tiling by translations only. If they are repeated by translations and rotations, the cylindrical curvature for the same tile would be different and would increase the amount of tiles required initially. The 5d Penrose tiling would have to be first generalized as 10 tiles and not just 2 in order to avoid the problem of various curvatures for one tile. Or if it is considered as a two tiles' aperiodic structure, under the rules of a cylinder, it will return to a 10 tiles' structure in order to retain the same curvature for each tile. Also, from the perspective of one tile because it is a parallelogram and it is located askew (not transversal) to the rotation axis of the cylinder, it is made of a surface which has a double curvature.

Geometrical development

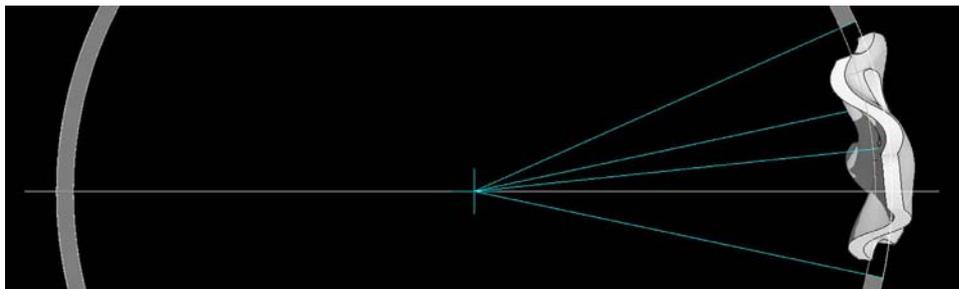


fig. 5s Front elevation of one NURBS tile within a cylindrical geometry showing the offset converging towards the centre in 2 directions (x and y not z)

By looking at one tile (*fig.5s*) with acceptable thickness beyond the paper thin model and in regards to the new spatial configuration (cylindrical), the tiles' edges raised some new geometrical challenges. Not only, had they tapered in opposite direction in sets of two parallel lines (creating undercuts) but they were also found warped (*fig.5t*) in order to meet the overall geometry of the cylinder (convergence to a central axis). This caused more of a problem in making it to the physical realm (next section) than it did resting inside the digital one.

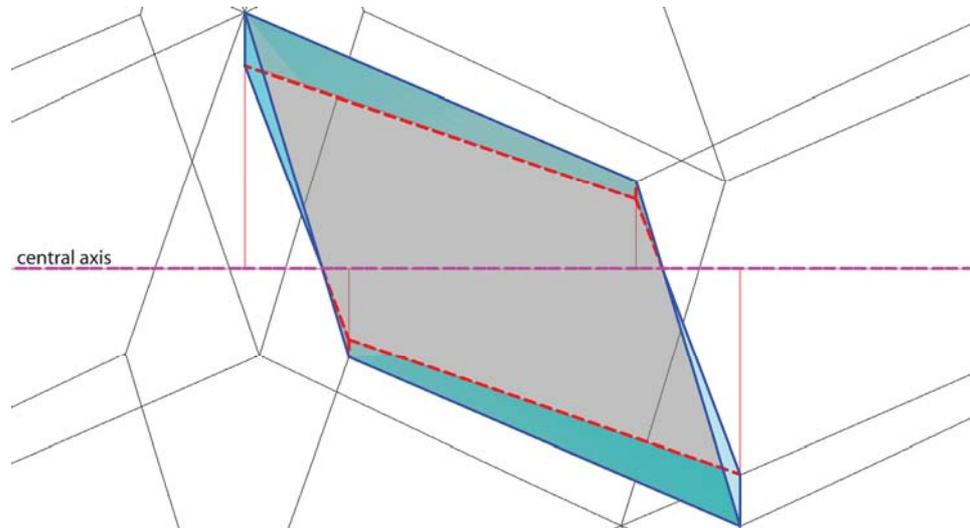


fig. 5t Side elevation of a default tile prior to NURBS adaptation showing tapered and warped edges when the tile's thickness is considered.

Similarly to the experiments carried out before with the 2d tilings, 6 NURBS tiles were later developed within the quasi roller's geometry (*fig. 5u*). All the tiles' vertices are converging perpendicularly along the central axis of rotation of the cylinder (blue line). The converging lines (orange lines), as shown in figure 5v became the basis from which all the tiles' edges were geometrically articulated. Once the 6 tiles were parametrically created, they could finally be associated (components) to a cylindrical field following the same adjacency rules from the case study in appendix C, p.62. A last experiment was to test the notion of porosity throughout the NURBS field (*fig. 5w, 5x*).

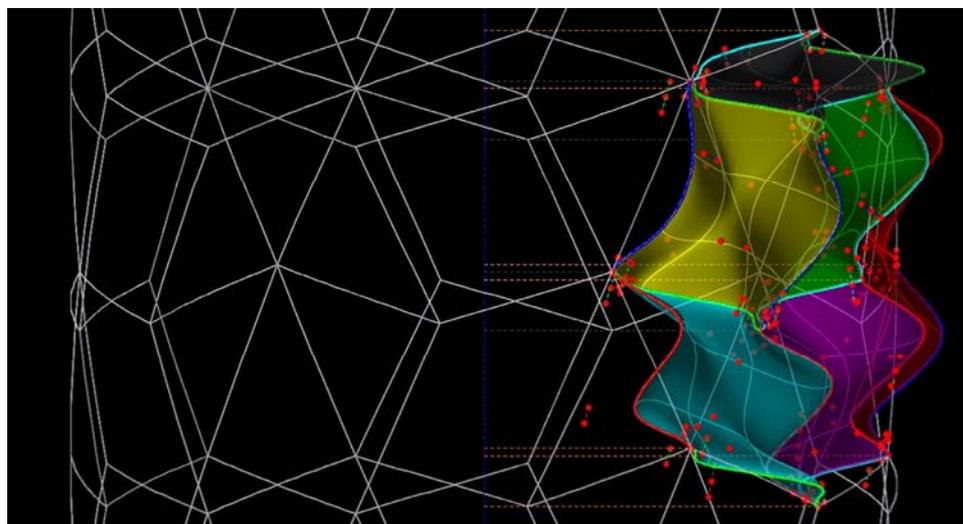


fig. 5u Side elevation of 6 NURBS tiles within the cylindrical geometry (axis convergence)

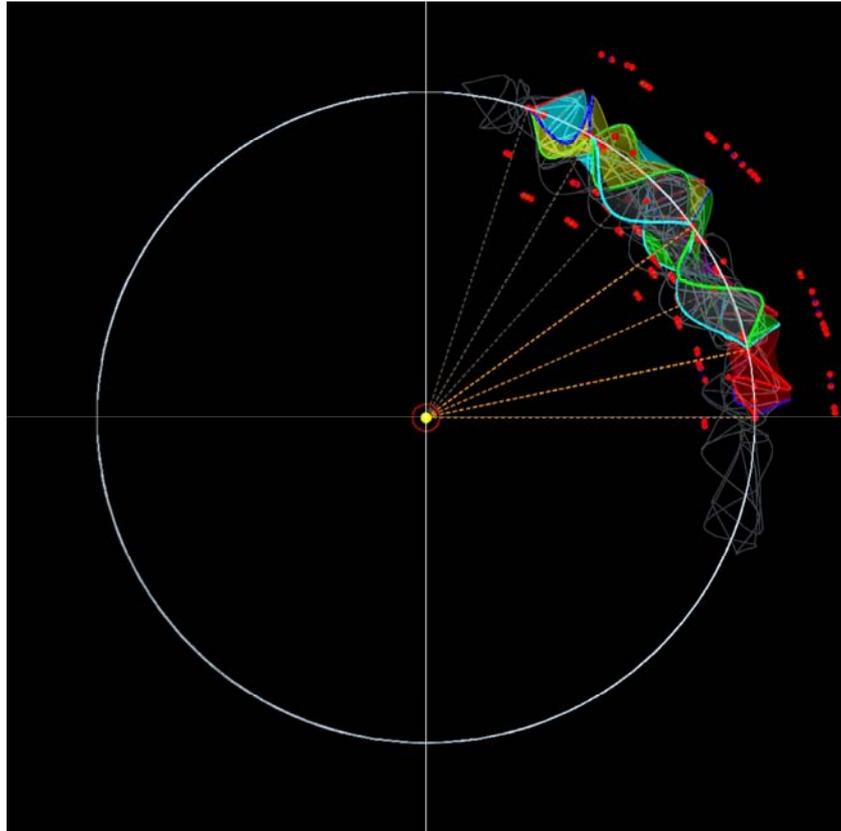


fig. 5v front elevation of 6 NURBS tiles within the cylindrical geometry (axis convergence)

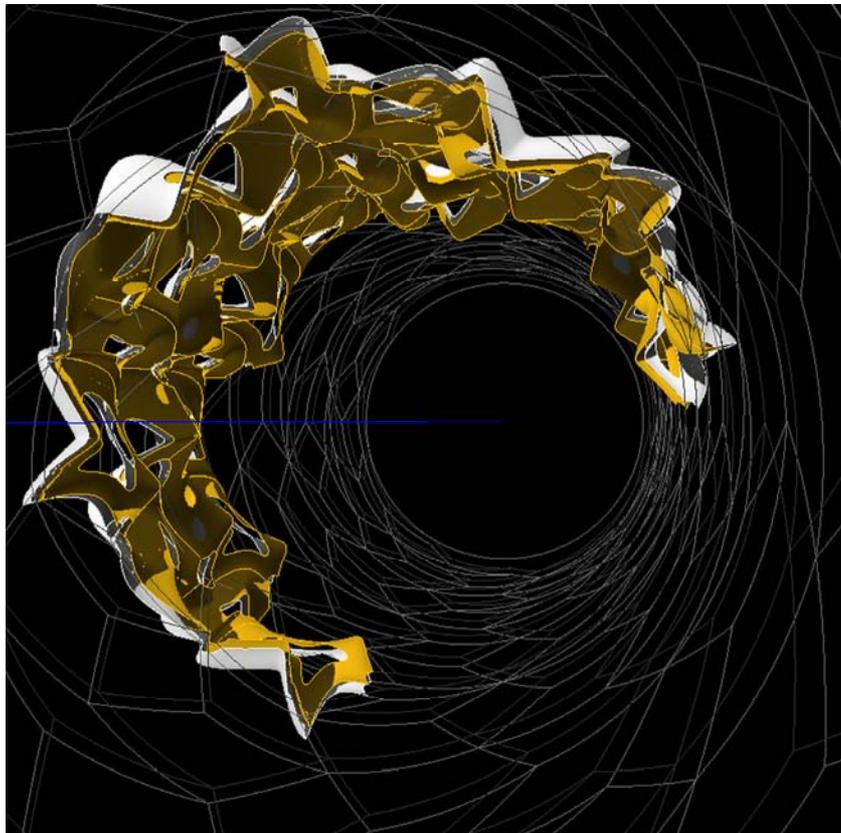


fig. 5w interior view of porous NURBS field within the aperiodic cylinder

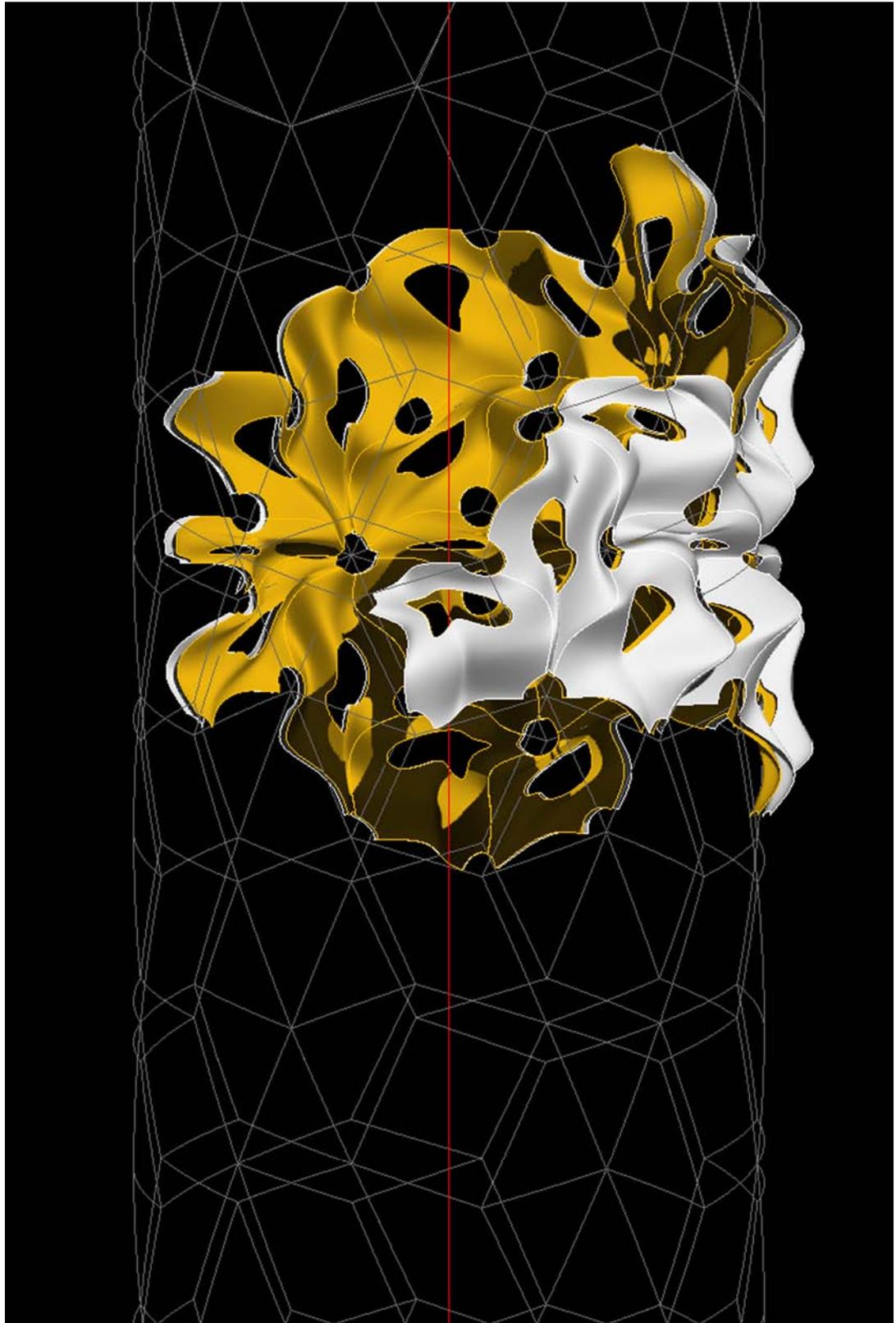


fig. 5x Side elevation of porous NURBS field within the aperiodic cylinder

6. Fabrication

As stated in objectives, fabrication was from the very start a central motivation in directing the research. At last, the aim was to mass produce some of the findings by repeatedly casting the necessary tiles with the least possible amount of formwork while not compromising on the overall ‘irregularities’ of generated patterns from the projected tilings (what is meant here by ‘irregularities’ and how it can be objectively measured must be discussed further in the next section). The fabrication process was conducted early on at two scales informing one another: the scale of the field and the 1 to 1 scale of the singular tile (*fig 6a, 6b*). Moreover attention to certain details was inescapable if physicality had to be conveyed. The main tool used was a CNC 3-axis milling machine. For the singular tiles, negative moulds were milled and positive casts (plaster, cement) were the final outputs (two-step production).

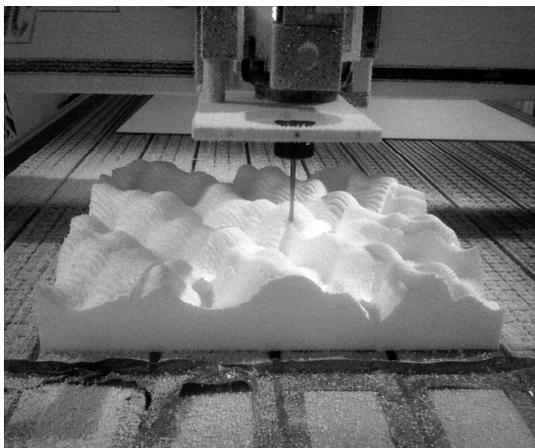


Fig 6a milling of part of a field for geometry check

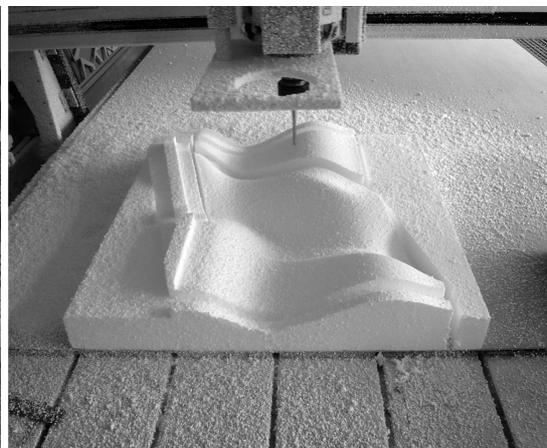


Fig 6b milling of part of a mould for a 1:1 cast of a tile

6.1 Fieldproof

The milling of some fields was already introduced (in the adaptation chapter) and was mainly conducted to verify that tangencies between NURBS tiles were working properly in providing for a continuous field as well as for testing the general amplitude of the NURBS geometry. It was often exaggerated to see how robust the tangency method was, before a tile would be milled at full scale.

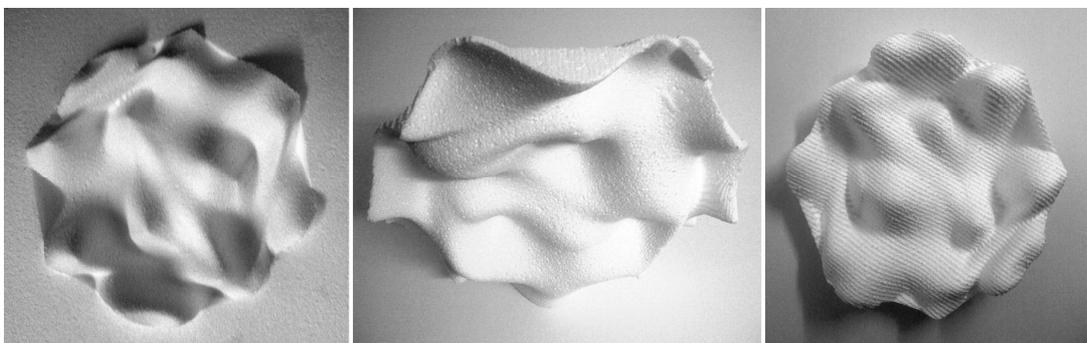


Fig 6c group of 6 tiles (two-sided) from a 4d tiling where tangency is still leaving creases on some edges

Figure 6c shows an early milling test where tangencies were not robust enough under extreme z-values. The problem led to the corrected method presented in figure 5j on page 37.

6.2 Tileproof

Some criteria for mass production (repeating many casts from the same moulds) were durability of the moulds, rapid setup prior to casting, easy mould release after casting. The full cycle had to be as efficient as possible to facilitate repetition.

Other criteria related to the performance of the finished tiles themselves were the tiles strengths upon stresses, tile reinforcements, surface finish, and most importantly how the tiles are connected to one another (no magic silicon!). These depended a lot on the spatial configuration of the tiling itself and its associated loads. For instance, it could be a 3d one-sided floor tiling, a wall tiling, a two-sided free standing wall (gravity), a porous screen, a porous quasi roller, etc.

A first prototype was developed for a free standing wall type (two sided), another for a porous quasi roller (interior/exterior).

Although the fabrication part is very much in the making and will continue to be so past this report, here follows excerpts of some of the first experiments pursued.

Quasi wall prototype (4d tiling: 6 tiles)

The following study is a two-sided NURBS tiling which has no undercuts. The first full scale prototype tested the changes in the z-values of the tiles' edges only (no x and y variations yet).

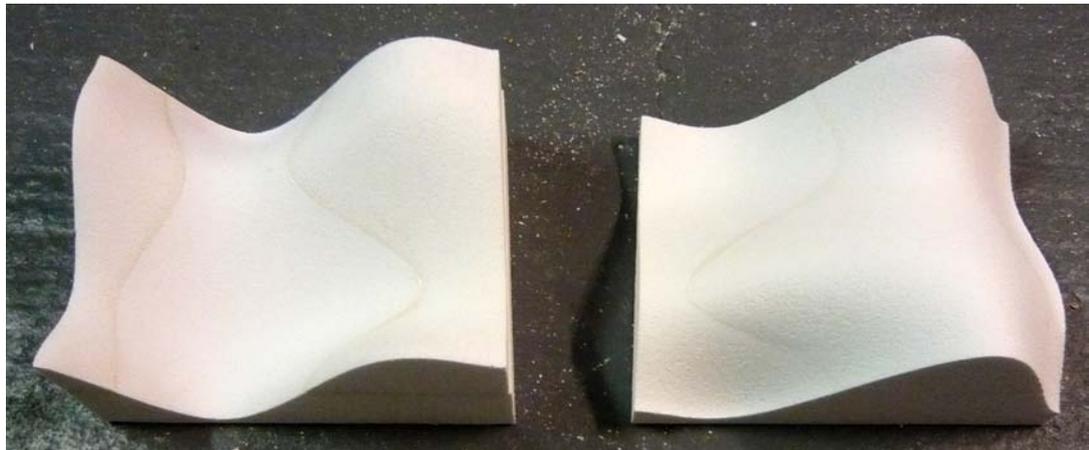


Fig. 6d one tile's corresponding two-part mould (dim: 400 x 400 x 250mm)

The 6 tiles are 3 dimensionally milled as a two-part mould (negative) matching the top and bottom of the tiles' surfaces (*fig. 6d*). Each half mould is laminated into two horizontal sections to reach the necessary height (z-value). They were made of EPS boards 30kg/m³ (Expanded Polystyrene) and glued together.

The strategy for connecting the tiles to one another was first to position in the moulds two pairs of hollow plastic tubes surrounded by plaster rings (spacers).

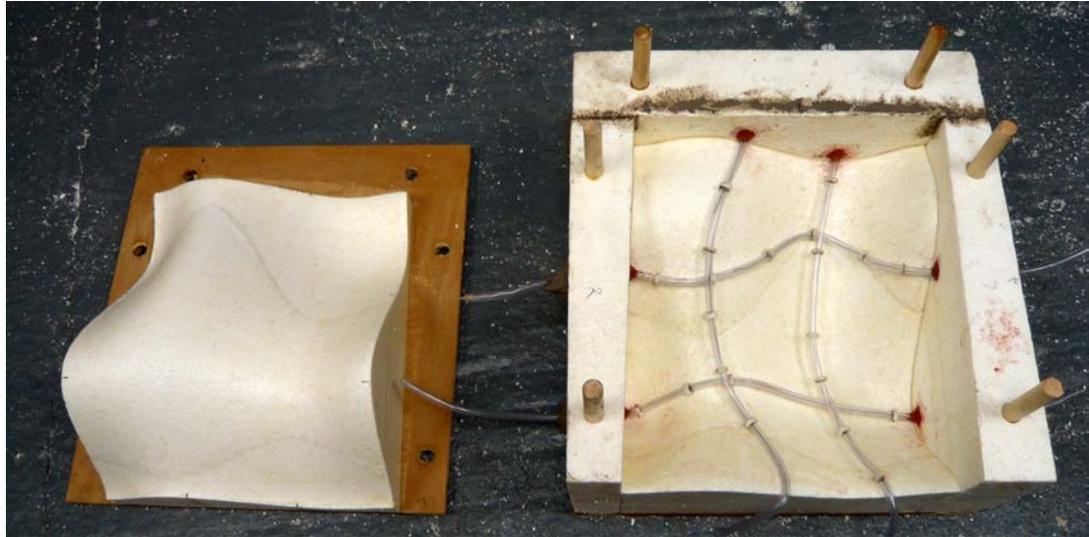


Fig. 6e same mould showing the addition of reinforcements, connection system and removable sides

The rings ensured the tubes would be kept away from the tile's finished surfaces and remain properly located within the tile's thickness (*fig.6e*). After the tiles were cast and dry, steel cables were threaded inside the hollow tubes to travel from tiles to tiles until an end was finally reached, at which they were tensioned and tightened (similar to the directions of the NURBS' isocurves in *fig. 5e*, p.35). The technique accounted for a non permanent site type of assembly and disassembly (*appendix F6*, p.76 for an assembly of 6 tiles). For permanency, the tubes would be replaced by continuous steel bars reinforcements (pre-bent) and the moulds would travel along their paths to make progressive pours.

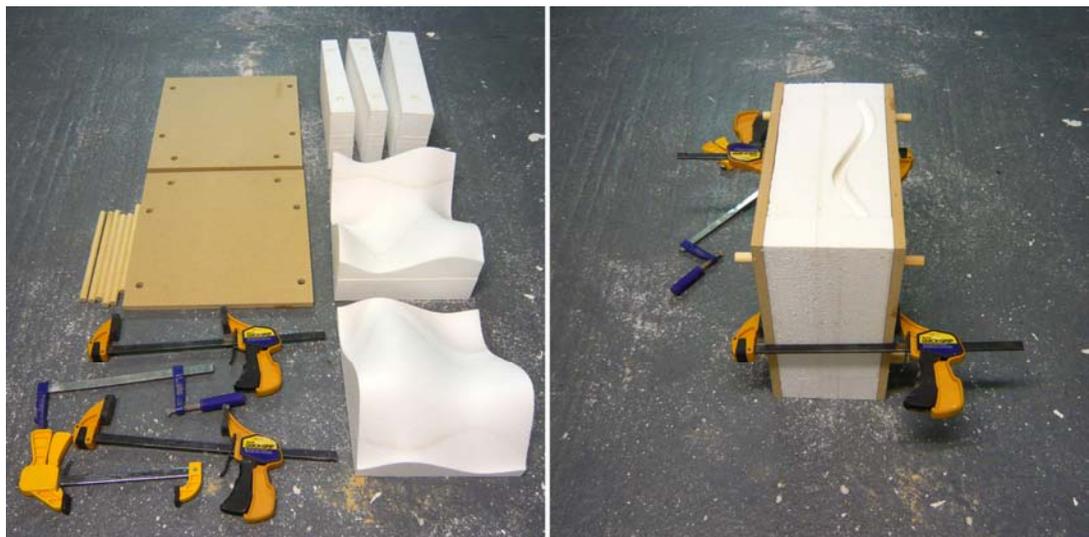


Fig. 6f the parts making a mould for easy assembly and disassembly

The figure above presents at this stage how a mould was devised in all its parts to allow for fast reuse and easy mould release.

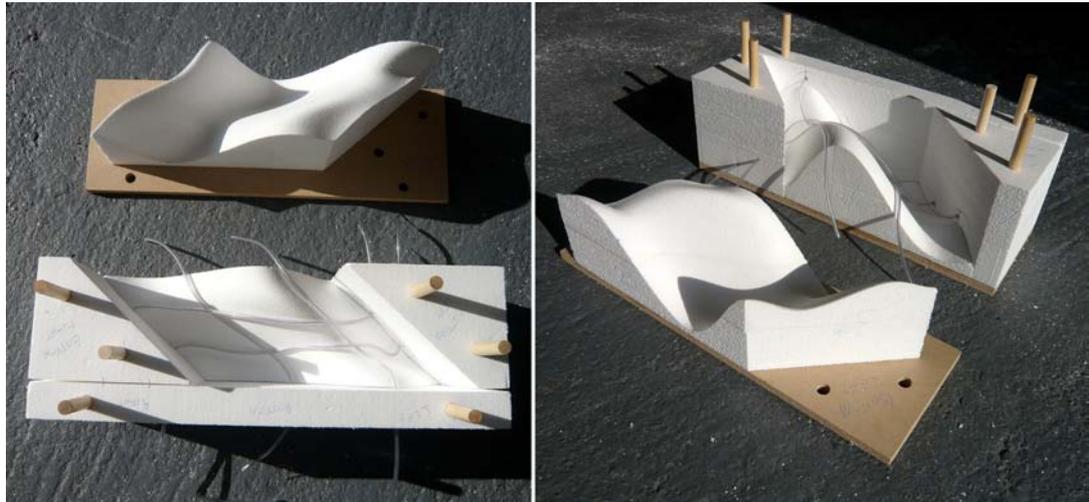


Fig. 6g other mould type showing variation in the tile' shape and in the z-amplitude of the edges



Fig. 6h mould preparation during and after casting

EPS was not as durable a material as desired, especially when involving reuse. After the technique was better perfected, fibreglass moulds were made instead to counter that problem. The milled moulds nonetheless had to accommodate for the change (see next prototype). Other fabricated moulds are documented in appendix G, p.79.

Quasi roller prototype (4d porous tiling)

The last prototype only recently began and was initiated by the making of one tile fitting a NURBS quasi cylinder (*same tile as in fig. 5s, p.41*). It further involved solving for the x, y and z changes in the NURBS' edges, holes at middle and corners of the tile and not the least, cylindrical curvatures disclosing undercuts (*as in fig. 5t*). The moulds' geometry for the milling was also revised to provide for the fibreglass change. It was applied as a layer directly onto the EPS moulds and then later released.

To solve for the double curvature of the tile and the edges' undercuts, the mould had to be conceived in three composite parts (prior to fibreglass application): Two negative beds (*fig. 6i*) and the actual positive tile (*fig. 6l*). The two beds accounted for the top and bottom surfaces of the tile but also served as supports in the process. The positive tile was flip-milled from one bed to the other so that the undercuts could be avoided (*fig. 6j, 6k*).



Fig. 6i the two beds for flip milling and for fibreglass mould



Fig. 6j the two beds on which the positive tile is being milled in 2 horizontal laminations

Once the three pieces were completed, the positive was placed back onto one of its bed (appendix G2) and then coated with fibreglass to make one half of

the final mould. The other half is done by coating the other bed alone. Once the resin is cured, the EPS is fully removed and the fibreglass mould ready for casting. The process turned out to be more robust as well as more flexible in working with this type of geometrical complexity.

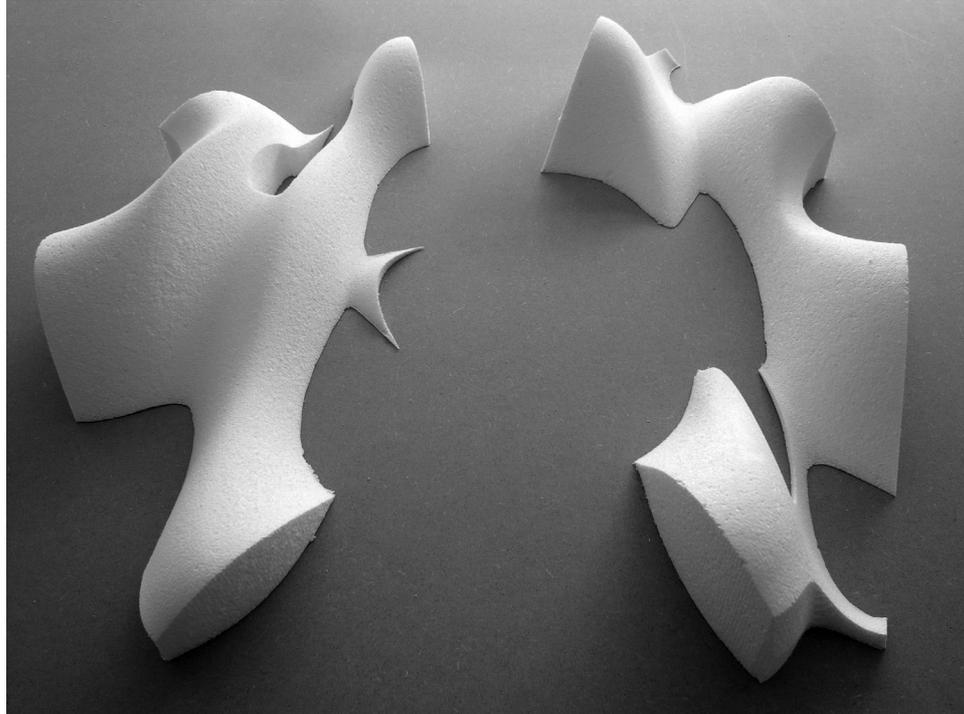


Fig. 6k views of the two halves of the positive tile (recto/verso) after milling and before lamination

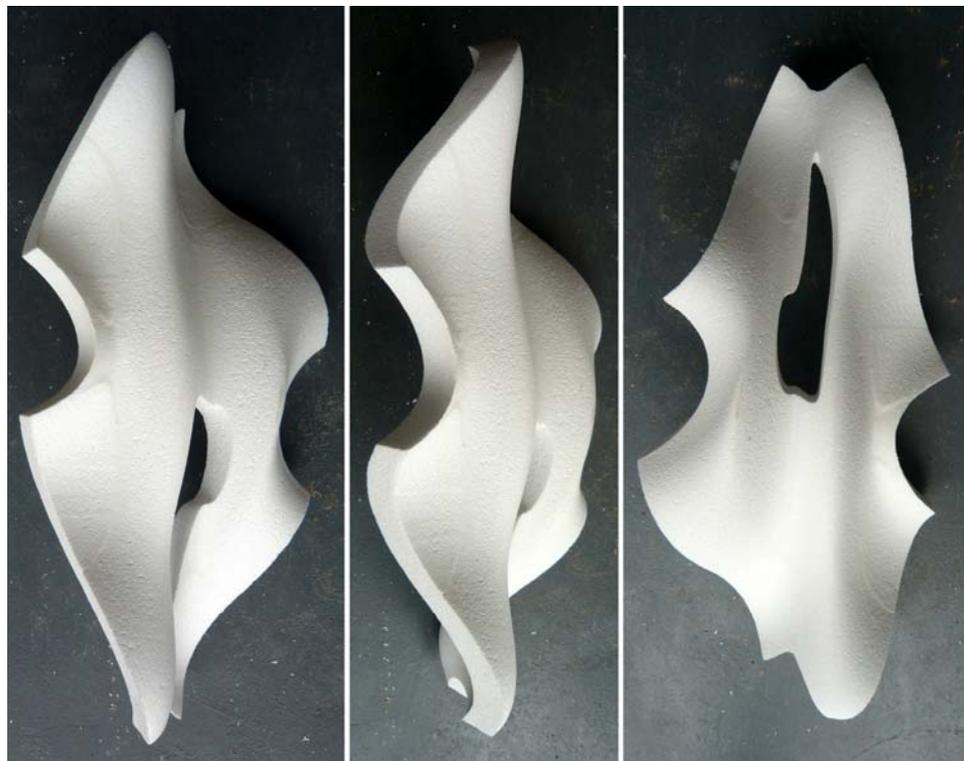


Fig. 6l views of the final positive tile after laminations

7. Discussion

Overall impressions

The word 'Irregularity' had been mentioned at various places in this report as a mean of measuring some of the premises of the thesis. One of which being how far away can a generated pattern (parametrically defined) distance itself from a comprehensive tiling (projection of n-dimensional grids) without abandoning its efficient assembly or adjacency rules, that is its ability to repeat with the least amount of tiles. 'Irregularity' here became the length of a scale stretching from an initial aperiodic tiling to a generated NURBS pattern to mainly gauge itself against that initial tiling: the **most** possible variations within a generated field with the **least** possible amount of tiles. The scale's units defined not only a level of recognition one had over the other but also etched along its length a series of parameters autonomous from the original tiling. Such parameters were, for example the ones defining the geometrical amplitude of a NURBS tile, the amount the porosity a field would have, the size of individual holes, etc. At this stage they were established manually in arbitrary ways, but they have the potential to be further conditioned meaningfully by external forces. For instance, could such parameters become more in tune with a specific environment? In a screen type or in the case of the *quasi roller*, could its structural envelope perform as a mediator between the amounts of natural (diffused) light is let in versus the amount of visibility is offered through to a participant?

Even though the research had not directly addressed these types of questions, it had freed up a ground for them to be explored extensively. Under which processes though would still have to be determined.

The exploration presented may also be perceived rather self-referential despite an effort to depart from the restrictions of the aesthetic appearances of the found tilings (in contrast to the RMIT example in the Background section). By comprehending how such tilings came about (the projection method), how their adjacencies were related and less how they graphically looked like, alternative geometries were able to 'economically' emerge spatially from it (adaptation chapter). The self-referential criticism could as well bring up to discussion a certain temptation of falling for the ornamental for the sake of retinal exuberance. The reader should decide if that is the case. Although in defence, it would be argued that to stand up, the tiles both in themselves and connected together had no choice but to become a structural entity (performing against gravity) and not merely an add-on decoration.

It often occurs throughout the scientific literature* on the subject that looking for aperiodic structures (quasicrystals) becomes a final goal in and of itself. Only sometimes, brief speculations are made on what they can be used for, often derived from their appearances.

* *"Things to do with Quasitiler: Redecorate the bathroom!"*
(From <http://www.geom.uiuc.edu/apps/quasitiler/>)

In stopping short to analyse what rules tilings more broadly are really founded on, the opportunity to think of them as mere organizers capable of effectively articulating remote geometries is often missed out or by default too promptly associated to the well known Penrose tiling and its specific assembly rules.

The strip projection method by arbitrary rotations

A principal characteristic of the proposed method was to render wide open the searching space in which tilings emerged. This was achieved by breaking apart an overall rotation matrix with specific values into a compound of individual rotations (around various planes) where each angular value could be parametrically adjusted (*see appendix A, p.58 for example in 6d*). Out of this implementation, a user had more chances to arrive to a *hybrid* tiling than a predefined case (Penrose). Because any dimensions could be activated in no specific order (top-down, bottom-up or any subjective orders in fact), the searching process was as valuable in cutting across projected dimensions as the moment when a tiling was completed. By hybrid, it is meant here a tiling made of a crossover between dimensions where for instance zone of 3d or 4d tiles are mixed with zone of 5d tiles, etc (*see case study #5, p.32*). A hybrid tiling had the extra potential to make a pattern (later developed) spatially richer by externalizing greater irregularities from the same amount of tiles. Despite the infinite quantity of tilings the method can disclose, it does have some limitations. It is known by now that for a tiling to perfectly emerge it requires a well tuned compound of rotation angles. The more rotations higher dimensions offer, the more the space to entirely arrive to a well tuned group of angular values is shrunk. The fact that they are operated manually in the interface does not facilitate the task at hand (approximations and rounding errors as explained by U. Vogg and P.L. Ryder, 1996, *A general algorithm for generating quasiperiodic lattices by the strip projection method*, Journal of Non-Crystalline Solids 194 134-144).

In theory it works absolutely fine but in practice it became harder and harder to arrive to a complete tiling as higher dimensional grids increased. For instance, the interface accommodates for the search for tiling from 7 dimensional grids, but its hunt can turn out painstaking and long. Could there be some other types of processes capable of fine tuning and directing more accurately this search? One dilemma is that there isn't a real beginning or end in determining a trajectory to pursue in the method. How to define what is being search for away from special cases with special sets of angles? It is similar to juggling at once with 15 balls (in 6d) in any prescriptive order. In other words, the searching criteria are a bit ad hoc and in the method are highly relying on step by step human intuition and curiosity (handling the rotations). Could that sort of intuition be algorithmically substituted or maybe only complemented?

Furthermore the hypercube example demonstrated that higher dimensions can be transferred back in three dimensions with different representations. In this research, with the help of transformations matrices (linear algebra) of

higher dimensions, rotations were used prior to 2d and 3d projections. Other types of transformations than rotations could have been operated and higher dimensional vectors represented in 3d space while still focusing on aperiodic structure. In the method just the x, y and z values of vectors were kept. It is important to stress that an abundance of parameters (x, y, z, a, b, c, d, etc) and computation are used here and a lot of it remained *latent* or to use a stronger term, wasted. Much more can be demanded from these extra vectorial variables than rotations. For instance, a 5 dimensional lattice could project a three dimensional tiling while its fourth parameter (coordinate: a) is being utilized and represented in Cartesian space to regulate the size or the amount of apertures upon a moving source of light. A not so great example merely to illustrate (out of a certain consciousness) that these extra dimensions were being disposed of too swiftly throughout the process.

Adaptation to fabrication

In the research, not as many explorations had been put into the three dimensional projections (of n-dimensional grids) as it went into the two dimensional ones. A lack of time was more the reason rather than one favoured over the other. Although the method provided for three dimensional projected outputs (see fig. 8w, 8x, 8y, etc) for any found tilings, they had yet to be experimented with in the adaptation of other possible geometries. The model presented in appendix F1 (p. 71) shows that 3 dimensional tiling came out in a plate-like structure due to the strip selection from the projection method. Within a plate unfinished aggregates (half sections) are embedded, nonetheless they contained enough information to complete them into full aggregate structures (formations of similar kinds to the presented water cube and quasi furniture in the background section). Also in the interface the region between the clipping planes can be parametrically incremented to incorporate complete aggregates. The distinction from two to three-dimensional projection raises a few queries on the choice of alternative geometries for adaptation and on the volumetric limitations the 2d projection has over the 3d. NURBS surfaces are known to support continuity and flow; therefore they were thought to be an appropriate choice for adaptation in competing against the strictly defined edges of tiles in 2d projections. Would this type of geometry be suitable to test the rules of 3d tilings remains to be seen? Faces forming 3d aggregates are in some places relatively acute and less desirable for tangential continuity. In addition, 3d aggregates have the potential to be much more flexible in making up volumes in space than when starting from 2d aperiodic structures. The further the experiments went from 2d projection was to the cylindrical configuration where the 2d vertices were not anymore contained within a 2d plane. Numerous experiments would still need to be carried out from 3d projections in order to test how far modularity can be challenged beyond the initial aggregates' identifications.

From the point of view of fabrication, a (full scale) single tile should integrate all the necessary details to make up for the whole without requiring a comprehensive notion of it. In reference back to a point made in the introduction, the concept of going from the building element or the part

towards the whole (in contrast to the usual reversal) makes the edge detail of a tile in this sense very crucial and informative. Their assembly into an entity is indeed governed locally by the manner a tile's edge meets its neighbours and from the original adjacency analysis of a tiling regardless of its overall 3 dimensional articulation. For instance in return to the example of the quasi roller, by assembling any of the 6 different tiles in the correct arrangement, it incrementally evolves towards a cylinder on the basis of an edge to edge correlation. The proposed route is additive rather than one based from divisions where an emphasis had to be put first on structure before enclosure. The thesis generally devoted itself to investigate how far an initial 2d tiling could make it to the 3-dimensional realm without breaking into unique pieces. At last, a direct way to evaluate how well a modular system performed is to judged it from the ratio between the **most** (generated spatial field) against the **least** (amount of different tiles required).

Further Work

In the short term, a porous quasi wall prototype will be carried out at 1 to 1 scale in the frame of a short workshop, followed by the further development and construction of a complete quasi roller with all the detailing involved. In the longer term, investigations in the geometry of n-dimensions will want to be pursued and translated to the field of generative design and architecture. But prior to that the missing experiments mentioned previously in tackling 3d projections of aperiodic aggregates would have to bear to some more fruition.

8. Conclusion

Repetitions of Differences

To put it simply, it is quite gratifying for a designer to know beforehand that because only few building elements would be reproduced over and over into a relatively intricate assemblage, more attention and endeavour can be channelled on just the few pieces accounting for it all.

By exposing and probing the space inside which aperiodic structures emerged, the research virtually extended the quantity of already known tilings to an infinite amount of possibilities for any higher dimensions. The computing performance of the machine became one of its limits. The pursued explorations not only detracted from the gravitation in and around the special cases (i.e. Penrose tilings) but essentially proposed through analytical means to liberate the organizational potentials of the found tilings from their strong graphical appeals.

Once comprehended as mere scaffolds, alternatives geometries could suddenly plot their courses past the tiles' boundaries and present relatively complex pattern formations without compromising on the scaffolds' efficiency.

On one hand, the position taken for the thesis was an attempt to demonstrate that one can conceive of modular systems without the restrictions frequently attached to them; *'repetitions of the same'*.

On the other hand, how far the quest for intricacy and irregularity can be stretched without falling into the gratuitous? A difficult question which answers can begin to be legitimized by the call for efficiency of means: Minimum amount of formwork for maximum overall irregularities.

This research would have reached its objectives if while interacting with a (full scale 4d NURBS) *quasi roller* for instance, the initial tiling its adjacency rules are based from and the fact it is made of only 6 tiles throughout, cannot be directly recognized; *'repetitions of differences'*.

This motto summarizes an approach which may or not partially contribute to the advance of modular systems, but which nonetheless highlights the fact that modularity has plenty of space left to be explored under conducted guidelines. In this regard, the 'modernist project' might have not yet ended.

References

Books

- D.M.Y. Sommerville, 1958, *An introduction to the Geometry of n Dimensions*, New York, Dover Publications, Inc.
- D P DiVincenzo & P J Steinhardt, 1999, *Quasicrystals: The State of the Art*, London, World Scientific.
- Marjorie Senechal, 1995, *Quasicrystals and Geometry*, Cambridge University Press.
- Roger Penrose, 1989, *The Emperor's New Mind*, New York, Oxford University Press.
- Philip Ball, 1999, *The self-Made Tapestry: Pattern Formation in Nature*, Oxford University Press.
- Patricia A. Egerton & William S. Hall, 1998, *Computer Graphics: Mathematical First Steps*, Harlow, England, Prentice Hall.
- Foley, Van Dam, Feiner, Hughes, 1996, *Introduction to Computer Graphics*, Reading, Massachusetts, Addison-Wesley Publishing Company.
- Foley, Van Dam, Feiner, Hughes, Philips, 1997, *Computer Graphics: Principles and Practice*, Boston, Addison-Wesley.
- Reas and Fry, 2007, *Processing: A programming Handbook for Visual Designers and Artists*, Cambridge, Massachusetts, Massachusetts Institute of Technology.
- H. Pottman, A. Asperl, M. Hofer, A. Kilian, 2007, *Architectural geometry*, Exton, PA, Bentley Institute Press.
- Aranda & Lash, 2006, *Tooling, Pamphlet Architecture 27*, New York, Princeton Architectural Press.
- Erwin Hauer, 2004, *Continua--Architectural Screens and Walls*, New York, Princeton Architectural Press.

Articles and Papers

- N.G.deBruijn, 1981, *Algebraic theory of Penrose's non-periodic tilings of the plane, I, II*, Nederl. Akad. Wetensch. Indag. Math. 43 p. 39-52, 53-66.
- U. Vogg, P.L. Ryder, 1996, *A general algorithm for generating quasiperiodic lattices by the strip projection method*, Journal of Non-Crystalline Solids 194 134-144.
- Jose Caceres, Roberto Marquez, 2002, *An aperiodic tiles machine*, Computational Geometry 23 171-182.
- Nagaraja S. Rao, E. R. Suryanarayan, 1994, *Quasiperiodic tilings generated by matrices*, Physica B 193 139-146.
- Philip Beesley, 2005, *Design Through Making: Orgone Reef*, AD Architectural Design, London, Wiley-Academy, p.46-53.
- Greg Lynn, 1993, *Architectural Curvilinearity: The Folded, the Pliant and the Supple*, Architectural Design, London, Wiley-Academy, p.102.
- Alison Boyle, 1997, *Quasicrystals to Kleenex*, millennium mathematics projects, University of Cambridge, <http://plus.maths.org/issue16/features/penrose/>
- Edwin Heathcote, February 23 2008, *The argument for Ornament*, London, Financial Times.

Web documents

- <http://www.geom.uiuc.edu/apps/quasitiler/> Quasitiler
- <http://www.jcrystal.com/> Quasicrystals
- <http://www.jcrystal.com/steffenweber/> Quasicrystals
- <http://steve.hollash.net/thesis/> 4D objects
- <http://www.stephencollins.net/Penrose/> Penrose Tiling Generator
- <http://condellpark.com/kd/quasig.htm> Quasig using the Pentagrid method
- <http://www.ams.org/featurecolumn/archive/ribbons.html> Updown Generation Penrose
- <http://www.giform.com/blobwall.html> Blobwall
- <http://www.dfab.arch.ethz.ch/web/e/lehre/index.html> Programmed wall
- <http://www.johnsontradinggallery.com/> Quasi Furniture
- <http://www.cgl.uwaterloo.ca/~csgk/software/penrose/> Penrose Tiling Applet
- <http://gregegan.customer.netspace.net.au/> DeBruijn Applet

Appendix A: Rotation matrices in 6d (x, y, z, a, b, c),

15 matrices around xy, xz, xa, xb, xc, yz, ya, yb, yc, za, zb, zc, ab, ac, bc plane

1)for xy

	x	y	z	a	b	c	g
x	cos(1)	sin(1)	0	0	0	0	px
y	-sin(1)	cos(1)	0	0	0	0	py
z	0	0	1	0	0	0	pz
a	0	0	0	1	0	0	pa
b	0	0	0	0	1	0	pb
c	0	0	0	0	0	1	pc
g	tx	ty	tz	ta	tb	tc	os

times vector |x, y, z, a, b, c, 1|

2)for xz

	x	y	z	a	b	c
x	cos(2)	0	sin(2)	0	0	0
y	0	1	0	0	0	0
z	-sin(2)	0	cos(2)	0	0	0
a	0	0	0	1	0	0
b	0	0	0	0	1	0
c	0	0	0	0	0	1

3)for xa

	x	y	z	a	b	c
x	cos(3)	0	0	sin(3)	0	0
y	0	1	0	0	0	0
z	0	0	1	0	0	0
a	-sin(3)	0	0	cos(3)	0	0
b	0	0	0	0	1	0
c	0	0	0	0	0	1

4)for xb

	x	y	z	a	b	c
x	cos(4)	0	0	0	sin(4)	0
y	0	1	0	0	0	0
z	0	0	1	0	0	0
a	0	0	0	1	0	0
b	-sin(4)	0	0	0	cos(4)	0
c	0	0	0	0	0	1

5)for xc

	x	y	z	a	b	c
x	cos(5)	0	0	0	0	sin(5)
y	0	1	0	0	0	0
z	0	0	1	0	0	0
a	0	0	0	1	0	0
b	0	0	0	0	1	0
c	-sin(5)	0	0	0	0	cos(5)

6)for yz

	x	y	z	a	b	c
x	1	0	0	0	0	0
y	0	cos(6)	sin(6)	0	0	0
z	0	-sin(6)	cos(6)	0	0	0
a	0	0	0	1	0	0
b	0	0	0	0	1	0
c	0	0	0	0	0	1

7)for ya

	x	y	z	a	b	c
x	1	0	0	0	0	0
y	0	cos(7)	0	sin(7)	0	0
z	0	0	1	0	0	0
a	0	-sin(7)	0	cos(7)	0	0
b	0	0	0	0	1	0
c	0	0	0	0	0	1

8)for yb

$$\begin{array}{l|cccccc}
 & x & y & z & a & b & c \\
 x & 1 & 0 & 0 & 0 & 0 & 0 \\
 y & 0 & \cos(8) & 0 & 0 & \sin(8) & 0 \\
 z & 0 & 0 & 1 & 0 & 0 & 0 \\
 a & 0 & 0 & 0 & 1 & 0 & 0 \\
 b & 0 & -\sin(8) & 0 & 0 & \cos(8) & 0 \\
 c & 0 & 0 & 0 & 0 & 0 & 1
 \end{array}$$

9)for yc

$$\begin{array}{l|cccccc}
 & x & y & z & a & b & c \\
 x & 1 & 0 & 0 & 0 & 0 & 0 \\
 y & 0 & \cos(9) & 0 & 0 & 0 & \sin(9) \\
 z & 0 & 0 & 1 & 0 & 0 & 0 \\
 a & 0 & 0 & 0 & 1 & 0 & 0 \\
 b & 0 & 0 & 0 & 0 & 1 & 0 \\
 c & 0 & -\sin(9) & 0 & 0 & 0 & \cos(9)
 \end{array}$$

10)for za

$$\begin{array}{l|cccccc}
 & x & y & z & a & b & c \\
 x & 1 & 0 & 0 & 0 & 0 & 0 \\
 y & 0 & 1 & 0 & 0 & 0 & 0 \\
 z & 0 & 0 & \cos(10) & \sin(10) & 0 & 0 \\
 a & 0 & 0 & -\sin(10) & \cos(10) & 0 & 0 \\
 b & 0 & 0 & 0 & 0 & 1 & 0 \\
 c & 0 & 0 & 0 & 0 & 0 & 1
 \end{array}$$

11)for zb

$$\begin{array}{l|cccccc}
 & x & y & z & a & b & c \\
 x & 1 & 0 & 0 & 0 & 0 & 0 \\
 y & 0 & 1 & 0 & 0 & 0 & 0 \\
 z & 0 & 0 & \cos(11) & 0 & \sin(11) & 0 \\
 a & 0 & 0 & 0 & 1 & 0 & 0 \\
 b & 0 & 0 & -\sin(11) & 0 & \cos(11) & 1 \\
 c & 0 & 0 & 0 & 0 & 0 & 0
 \end{array}$$

12)for zc

$$\begin{array}{l|cccccc}
 & x & y & z & a & b & c \\
 x & 1 & 0 & 0 & 0 & 0 & 0 \\
 y & 0 & 1 & 0 & 0 & 0 & 0 \\
 z & 0 & 0 & \cos(12) & 0 & 0 & \sin(12) \\
 a & 0 & 0 & 0 & 1 & 0 & 0 \\
 b & 0 & 0 & 0 & 0 & 1 & 0 \\
 c & 0 & 0 & -\sin(12) & 0 & 0 & \cos(12)
 \end{array}$$

13)for ab

$$\begin{array}{l|cccccc}
 & x & y & z & a & b & c \\
 x & 1 & 0 & 0 & 0 & 0 & 0 \\
 y & 0 & 1 & 0 & 0 & 0 & 0 \\
 z & 0 & 0 & 1 & 0 & 0 & 0 \\
 a & 0 & 0 & 0 & \cos(13) & \sin(13) & 0 \\
 b & 0 & 0 & 0 & -\sin(13) & \cos(13) & 0 \\
 c & 0 & 0 & 0 & 0 & 0 & 1
 \end{array}$$

14)for ac

$$\begin{array}{l|cccccc}
 & x & y & z & a & b & c \\
 x & 1 & 0 & 0 & 0 & 0 & 0 \\
 y & 0 & 1 & 0 & 0 & 0 & 0 \\
 z & 0 & 0 & 1 & 0 & 0 & 0 \\
 a & 0 & 0 & 0 & \cos(14) & 0 & \sin(14) \\
 b & 0 & 0 & 0 & 0 & 1 & 0 \\
 c & 0 & 0 & 0 & -\sin(14) & 0 & \cos(14)
 \end{array}$$

15)for bc

$$\begin{array}{l|cccccc}
 & x & y & z & a & b & c \\
 x & 1 & 0 & 0 & 0 & 0 & 0 \\
 y & 0 & 1 & 0 & 0 & 0 & 0 \\
 z & 0 & 0 & 1 & 0 & 0 & 0 \\
 a & 0 & 0 & 0 & 1 & 0 & 0 \\
 b & 0 & 0 & 0 & 0 & \cos(15) & \sin(15) \\
 c & 0 & 0 & 0 & 0 & -\sin(15) & \cos(15)
 \end{array}$$

Appendix B: Processing code snippets for 6d to 3d strip projection method

```
Matrix mult(Matrix a, Matrix b)
{
  Matrix result= new Matrix (a.cols, b.rows);
  for (int i=0; i<b.rows; i++){
    for(int j=0; j<a.cols; j++){
      result.matrix[i][j]= 0.0;
      for(int x=0; x<a.cols; x++){
        result.matrix[i][j]+= a.matrix[i][x]* b.matrix[x][j];
      }
    }
  }
  return result;
}

/////////////////////////////////////////////////////////////////

void xform(Matrix b)
{
  Vector result=new Vector(7);
  for(int i=0; i<7; i++){
    result.vector[i]=0.0;
    for(int j=0; j<7; j++){
      if(i!=6){
        result.vector[i]+=vector[j]*b.matrix[j][i];
      }
      else{
        result.vector[i] +=b.matrix[j][i];
      }
    }
  }
  for(int i=0; i<6; i++){
    vector[i]= result.vector[i]/result.vector[6];
  }
}

/////////////////////////////////////////////////////////////////

void rot_yc(float ang_yc)
{
  matrix[1][1]=cos(ang_yc);
  matrix[5][1]=-sin(ang_yc);
  matrix[1][5]=sin(ang_yc);
  matrix[5][5]=cos(ang_yc);
}

/////////////////////////////////////////////////////////////////

void invRot(Matrix inv)
{
  rows=inv.rows;
  cols=inv.cols;
  for(int i=0; i<rows; i++){
    for(int j=0; j<cols; j++){
      matrix[i][j]=inv.matrix[j][i];
    }
  }
}

/////////////////////////////////////////////////////////////////

rotA=mult(p_xy,p_yz);
rotA=mult(rotA,p_xz);
rotA=mult(rotA,p_xa);
rotA=mult(rotA,p_ya);
rotA=mult(rotA,p_za);
rotA=mult(rotA,p_xb);
rotA=mult(rotA,p_yb);
rotA=mult(rotA,p_zb);
rotA=mult(rotA,p_ab);
rotA=mult(rotA,p_xc);
rotA=mult(rotA,p_yc);
rotA=mult(rotA,p_zc);
rotA=mult(rotA,p_ac);
rotA=mult(rotA,p_bc);

rotG=new Matrix(7,7);
rotG.invRot(rotA);

/////////////////////////////////////////////////////////////////
```

```

//test with inverse rotatons to determine the clipping plane.
Vector temp_test= new Vector((int)0, (int)0, 100, 100, 100, 100);

temp_test.xform(rotG);//overall inverse rotation matrix

//find diagonal for clipping thickness
Vector temp_1= new Vector(100*zIn*zOut, 100*zIn*zOut, 100*zIn*zOut, 100*zIn*zOut, 100*zIn*zOut, 100*zIn*zOut);

if(temp_test.vector[0]<0){
    temp_1.vector[0]=-temp_1.vector[0];
}
if(temp_test.vector[1]<0){
    temp_1.vector[1]=-temp_1.vector[1];
}
if(temp_test.vector[2]<0){
    temp_1.vector[2]=-temp_1.vector[2];
}
if(temp_test.vector[3]<0){
    temp_1.vector[3]=-temp_1.vector[3];
}
if(temp_test.vector[4]<0){
    temp_1.vector[4]=-temp_1.vector[4];
}
if(temp_test.vector[5]<0){
    temp_1.vector[5]=-temp_1.vector[5];
}

Matrix p6=new Matrix(7,7);
p6.makeTrans(425,425,0,0,0,0);
trans2=mult(rotA,p6);
temp_1.xform(trans2);

float diag= clip3d* temp_1.vector[2];
float diag2= clip4d* temp_1.vector[3];
float diag3= clip5d* temp_1.vector[4];
float diag4= clip6d* temp_1.vector[5];

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

for(int i=-2-incPt; i<2+incPt; i++){
    for(int j=-2-incPt; j<2+incPt; j++){
        for(int k=-2-incPt; k<2+incPt; k++){
            for(int m=-2-incPt; m<2+incPt; m++){
                for(int n=-2-incPt; n<2+incPt; n++){
                    for(int o=-2-incPt; o<2+incPt; o++){

                        Vector temp_q = new Vector(i*100*zIn*zOut, j*100*zIn*zOut, k*100*zIn*zOut, m*100*zIn*zOut, n*100*zIn*zOut,
o*100*zIn*zOut);
                        Vector temp_x = new Vector((i+1)*100*zIn*zOut, j*100*zIn*zOut, k*100*zIn*zOut, m*100*zIn*zOut, n*100*zIn*zOut,
o*100*zIn*zOut);
                        Vector temp_y = new Vector(i*100*zIn*zOut, (j+1)*100*zIn*zOut, k*100*zIn*zOut, m*100*zIn*zOut, n*100*zIn*zOut,
o*100*zIn*zOut);
                        Vector temp_z = new Vector(i*100*zIn*zOut, j*100*zIn*zOut, (k+1)*100*zIn*zOut, m*100*zIn*zOut, n*100*zIn*zOut,
o*100*zIn*zOut);
                        Vector temp_w = new Vector(i*100*zIn*zOut, j*100*zIn*zOut, k*100*zIn*zOut, (m+1)*100*zIn*zOut, n*100*zIn*zOut,
o*100*zIn*zOut);
                        Vector temp_b = new Vector(i*100*zIn*zOut, j*100*zIn*zOut, k*100*zIn*zOut, m*100*zIn*zOut, (n+1)*100*zIn*zOut,
o*100*zIn*zOut);
                        Vector temp_c = new Vector(i*100*zIn*zOut, j*100*zIn*zOut, k*100*zIn*zOut, m*100*zIn*zOut, n*100*zIn*zOut,
(o+1)*100*zIn*zOut);

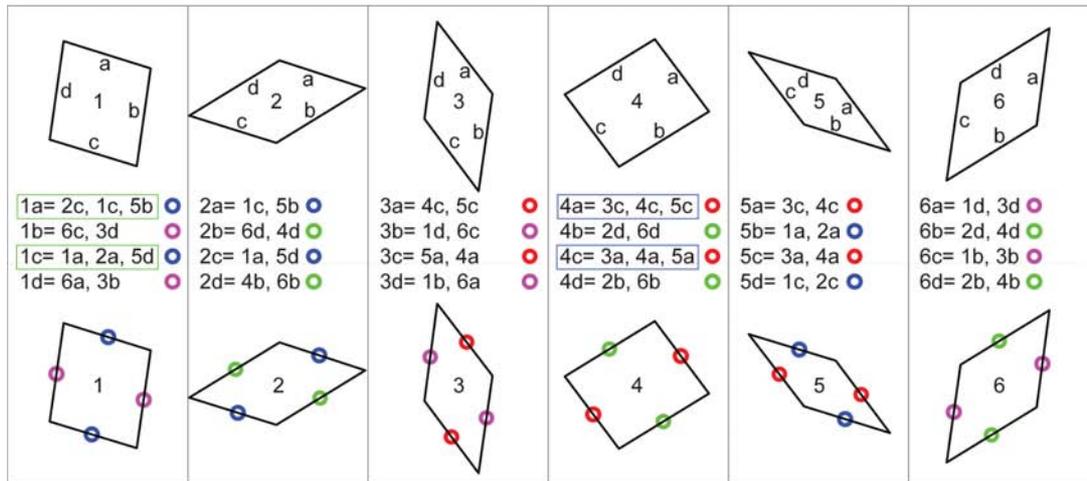
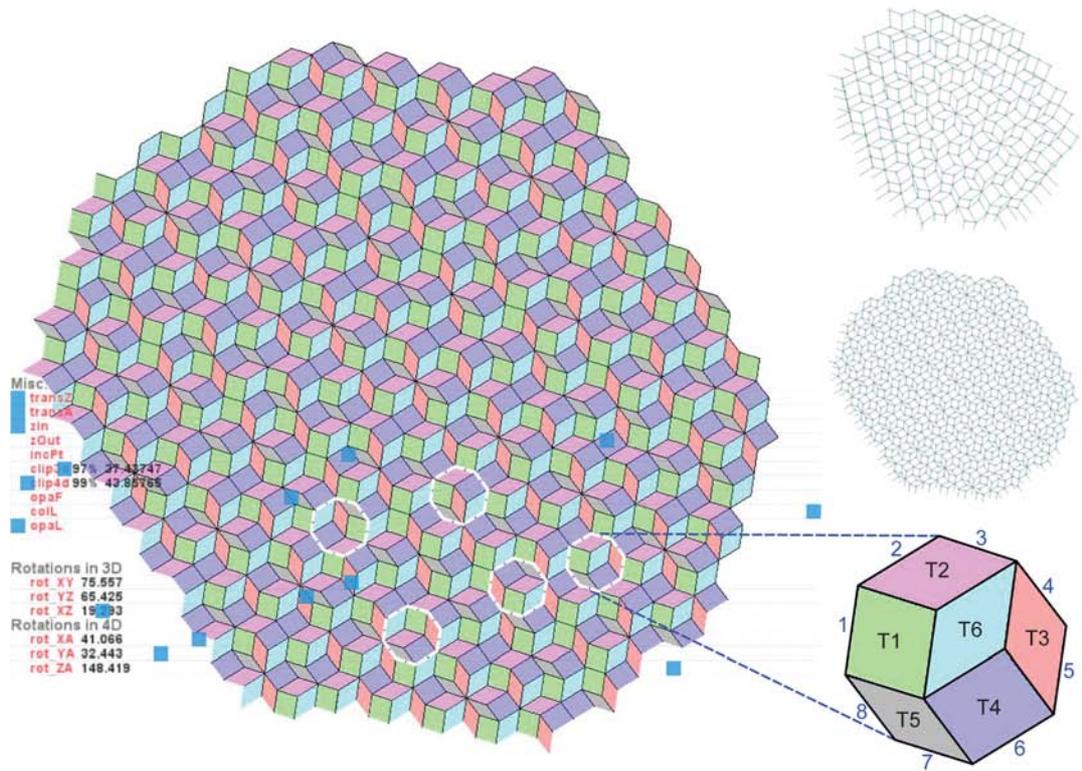
                        Vector temp_w1 = new Vector(i*100*zIn*zOut, j*100*zIn*zOut, (k+1)*100*zIn*zOut, (m+1)*100*zIn*zOut, n*100*zIn*zOut,
o*100*zIn*zOut);
                        Vector temp_w2 = new Vector(i*100*zIn*zOut, (j+1)*100*zIn*zOut, k*100*zIn*zOut, (m+1)*100*zIn*zOut, n*100*zIn*zOut,
o*100*zIn*zOut);
                        Vector temp_w3 = new Vector((i+1)*100*zIn*zOut, j*100*zIn*zOut, k*100*zIn*zOut, (m+1)*100*zIn*zOut, n*100*zIn*zOut,
o*100*zIn*zOut);

                        stroke(0,100,0);
                        Matrix p5=new Matrix(7,7);
                        p5.makeTrans(425,425,transZ,transA,0,0);
                        trans=mult(rotA,p5);
                        temp_q.xform(trans);
                        temp_x.xform(trans);
                        temp_y.xform(trans);
                        temp_z.xform(trans);
                        temp_w.xform(trans);
                        temp_b.xform(trans);
                        temp_c.xform(trans);
                    }
                }
            }
        }
    }
}

```


Appendix C: Adjacency rules

Case study #7 (4d tiling)



- Line type 1: 1a, 1c, 2a, 2c, 5b, 5d
- Line type 2: 1b, 1d, 3b, 3d, 5a, 5c
- Line type 3: 2b, 2d, 4b, 4d, 6b, 6d
- Line type 4: 3a, 3c, 4a, 4c, 5a, 5c

Appendix D: Graph variables and script snippets (Generative Components)

The image shows two side-by-side screenshots of the Generative Components software interface. Each window displays a table of variables and their values, along with a control bar at the bottom for limiting values and hiding variables.

Left Window: ELEMENTS_FIELD - GenerativeComponents

Name	Value	Analog Value
a_x1	199	
a_x2	598	
a_y1	-150	
a_y2	100	
a_z0	0	
a_z1	-500	
a_z2	700	
a_z3	a_z0 (0)	
b_x1	326.926	
b_x2	653.852 (650)	
b_y1	-100	
b_y2	-100	
b_z0	a_z0 (0)	
b_z1	500	
b_z2	100	
b_z3	a_z0 (0)	
c_x1	448.43337...	
c_x2	696.866	
c_y1	-300	
c_y2	0	
c_z0	0	
c_z1	400	
c_z2	-500	
c_z3	a_z0 (0)	
d_x1	250.16576...	
d_x2	650.33153...	
d_y1	0	
d_y2	-50	
d_z0	0	
d_z1	-200	
d_z2	200	
d_z3	a_z0 (0)	
offset	100	
offset_edges	0.2	
order	4	
symbolSize	50	

Right Window: TILES40_ELEMENT_HOLE40_05 - GenerativeComponents

Name	Value	Analog Value
c_x1	400.43337...	
c_x2	600.866	
c_y1	250	
c_y2	-250	
c_z0	a_z0 (0)	
c_z0b	a_z0b (133)	
c_z1	400	
c_z2	-400	
c_z3	a_z0 (0)	
c_z3b	a_z0b (133)	
d_x1	300.16576...	
d_x2	400.33153...	
d_y1	150	
d_y2	0	
d_z0	a_z0 (0)	
d_z0b	a_z0b (133)	
d_z1	0	
d_z2	50	
d_z3	a_z0 (0)	
d_z3b	a_z0b (133)	
offset	100	
offset_bottom	100	
offset_edges	0.3	
order	4	
symbolSize	50	
zcyan_x	325	
zcyan_y	100	
zcyan_z	275	
zgreen_x	-50	
zgreen_y	50	
zgreen_z	350	
zgrey_x	0	
zgrey_y	0	
zgrey_z	0	
zmagenta_x	-150	
zmagenta_y	0	
zmagenta_z	575	
zred_x	-150	
zred_y	75	
zred_z	100	

D1. two examples of graph variable tables for 4d NURBS tilings

transaction modelBased "GraphFunction polyBottom"

```

{
  feature polyBottom GC.GraphFunction
  {
    Definition = function (CoordinateSystem cs1 ,Polygon poly ,int start , BSplineCurve bs,double scalarZ )
    {
      //breakpoint;
      Point projPt={};
      DVector3d vect3dZ ={};
      DPoint3d ptD3d1={};
      for (int i = 0; i < poly.Vertices.Count-1;i++)
      {
        projPt[i]= new Point ();
        projPt[i].ProjectOnToCurve(bs.poly.Vertices[i]);
        vect3dZ[i] = new DVector3d();
        vect3dZ[i].Init( projPt[i].DPoint3d,poly.Vertices[i].DPoint3d);
        vect3dZ[i].NormalizeInPlace();
        vect3dZ[i].ScaleInPlace(scalarZ);
        ptD3d1[i]= new DPoint3d();
        ptD3d1[i].Init(poly.Vertices[i].DPoint3d);
        ptD3d1[i].AddInPlace(vect3dZ[i]);
      }
      pt= new Point();
      pt.FromDPoint3d(cs1, ptD3d1);
      Polygon polyOffset= new Polygon ();
      polyOffset.ByVertices(pt);
      return polyOffset;
    }
  }
}

```

```

transaction modelBased "graph function pt"
{
  feature pt GC.GraphFunction
  {
    Definition = function (CoordinateSystem cs , Polygon poly , int start , int end, int start1 ,BSplineCurve bs , double scalarX ,
                          double scalarY, double scalarZ , double scalarOff)

    {
      //breakpoint;
      iCount=0;
      Point pt= {};

      // vector X
      DVector3d vect3dX= new DVector3d ();
      vect3dX.Init( poly.Vertices[start].DPoint3d,poly.Vertices[end].DPoint3d);
      vect3dX.NormalizeInPlace ();
      vect3dX.ScaleInPlace (scalarX);
      DPoint3d pt3dBase = new DPoint3d ();
      pt3dBase.Init(poly.Vertices[start].DPoint3d);
      pt3dBase.AddInPlace(vect3dX);
      Point ptB = new Point ();
      ptB.FromDPoint3d(cs, pt3dBase);

      //vector y
      Point projPt = new Point();
      projPt.ProjectOnToCurve(bs, ptB);
      DVector3d vect3DProj = new DVector3d();
      vect3DProj.Init( ptB.DPoint3d ,projPt.DPoint3d);
      vect3DProj.NormalizeInPlace();
      DVector3d vect3dXn= new DVector3d();
      vect3dXn.Init(poly.Vertices[start].DPoint3d,poly.Vertices[end].DPoint3d);
      vect3dXn.NormalizeInPlace();
      DVector3d vect3dYCrossP = new DVector3d();
      vect3dYCrossP=DVector3d.FromCrossProduct (vect3dXn, vect3DProj);
      vect3dYCrossP.NormalizeInPlace();
      DVector3d vect3dY = new DVector3d();
      vect3dY=DVector3d.FromCrossProduct (vect3dX, vect3DProj);
      vect3dY.NormalizeInPlace();
      vect3dY.ScaleInPlace(scalarY);

      //vector z
      DVector3d vect3dZ = new DVector3d();
      vect3dZ.Init(ptB.DPoint3d ,projPt.DPoint3d);
      vect3dZ.NormalizeInPlace();
      vect3dZ.ScaleInPlace(scalarZ);

      DPoint3d pt3d= new DPoint3d();
      pt3d.Init(poly.Vertices[start].DPoint3d);
      pt3d.AddInPlace(vect3dX);
      pt3d.AddInPlace(vect3dY);
      pt3d.AddInPlace(vect3dZ);

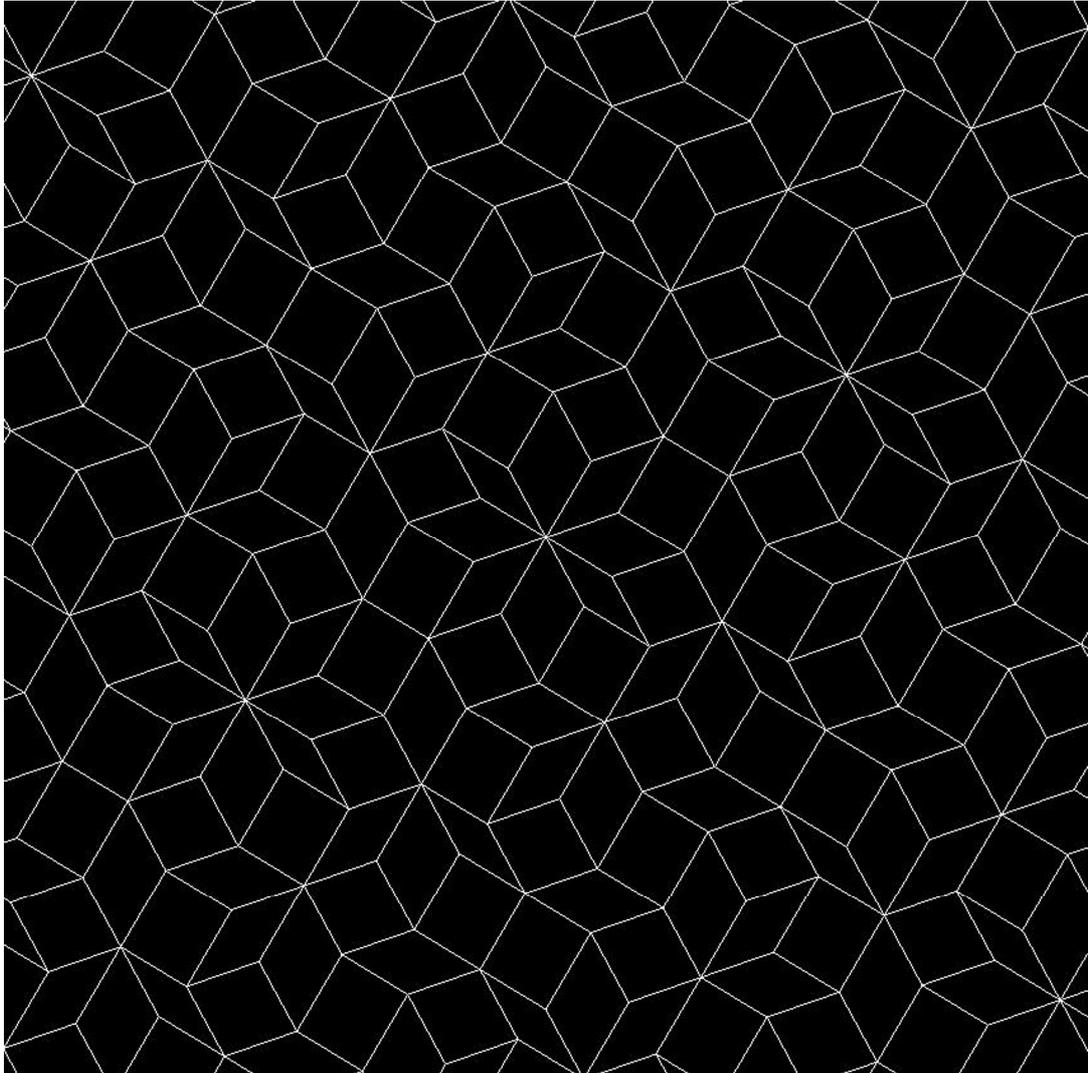
      //vector offset
      DVector3d vect3dXOff= new DVector3d();
      vect3dXOff.Init(poly.Vertices[start].DPoint3d,poly.Vertices[end].DPoint3d);
      vect3dXOff.NormalizeInPlace();
      DVector3d vect3DProjOff= new DVector3d();
      vect3DProjOff.Init(poly.Vertices[start1].DPoint3d ,pt3d);
      vect3DProjOff.NormalizeInPlace();
      DVector3d vect3dOff = new DVector3d();
      vect3dOff=DVector3d.FromCrossProduct(vect3DProj ,vect3DProjOff);
      vect3dOff.NormalizeInPlace();
      vect3dOff.ScaleInPlace(scalarOff);
      DPoint3d pt3d1= new DPoint3d();
      pt3d1.Init(pt3d);
      pt3d1.AddInPlace(vect3dOff);
      iCount=0;
      pt[0]= new Point();
      pt[0].FromDPoint3d(cs, pt3d);
      iCount++;
      pt[1]= new Point();
      pt[1].FromDPoint3d(cs, pt3d1);
      iCount++;

      return pt;
    }
  }
}

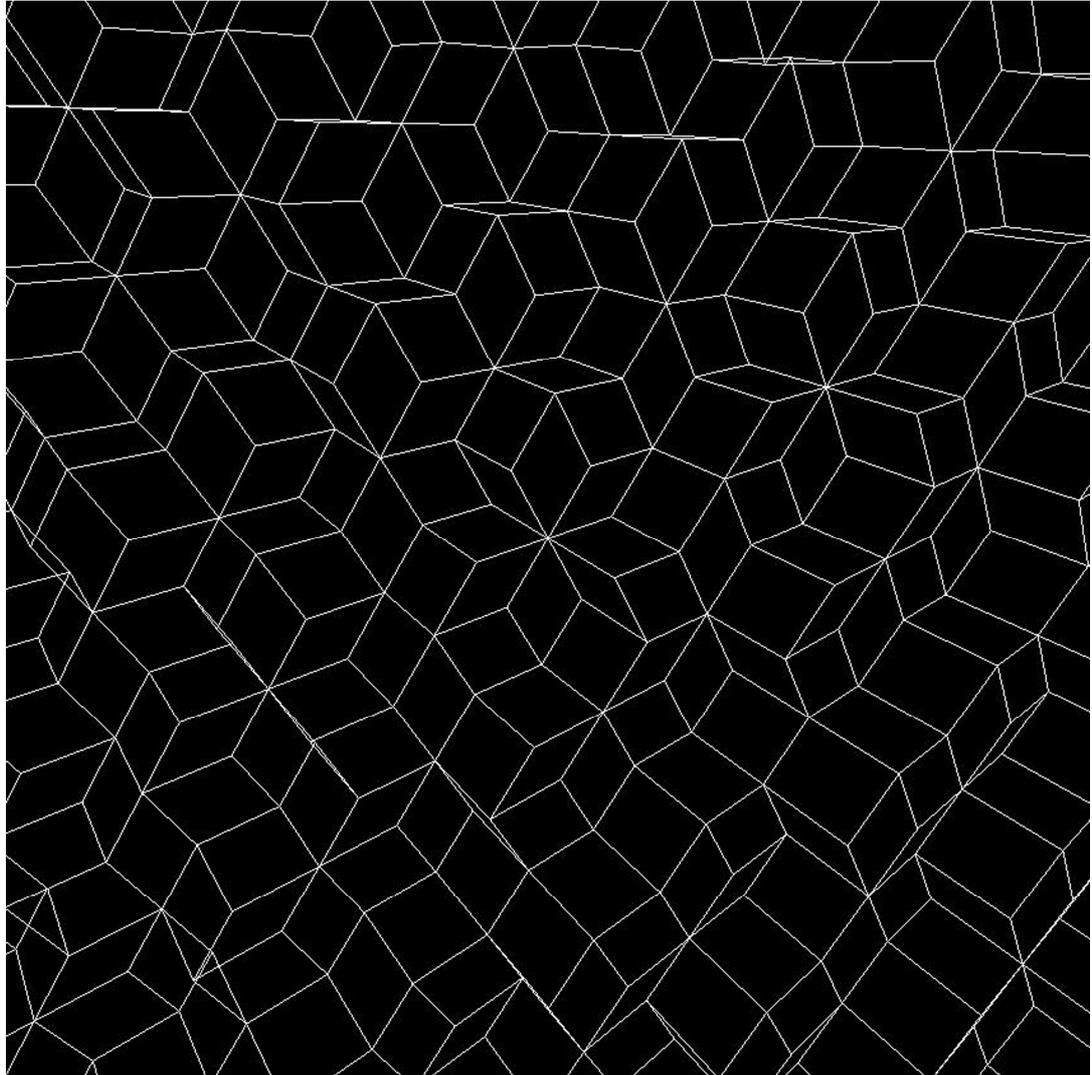
```

D2. Script showing *DPoint 3d/DVector3d* geometry employed for NURBS adaptation

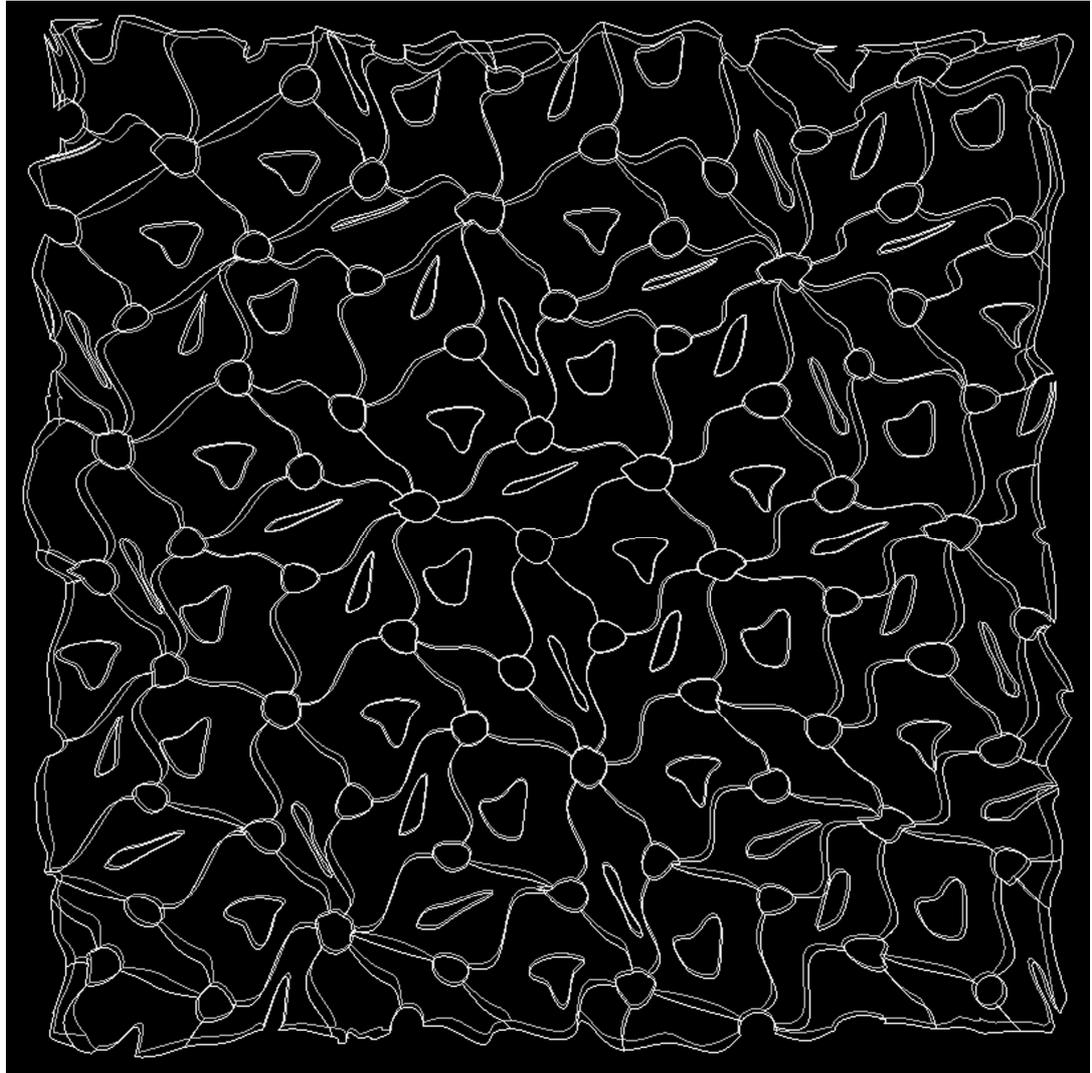
Appendix E: *Drawings of aperiodic structures*



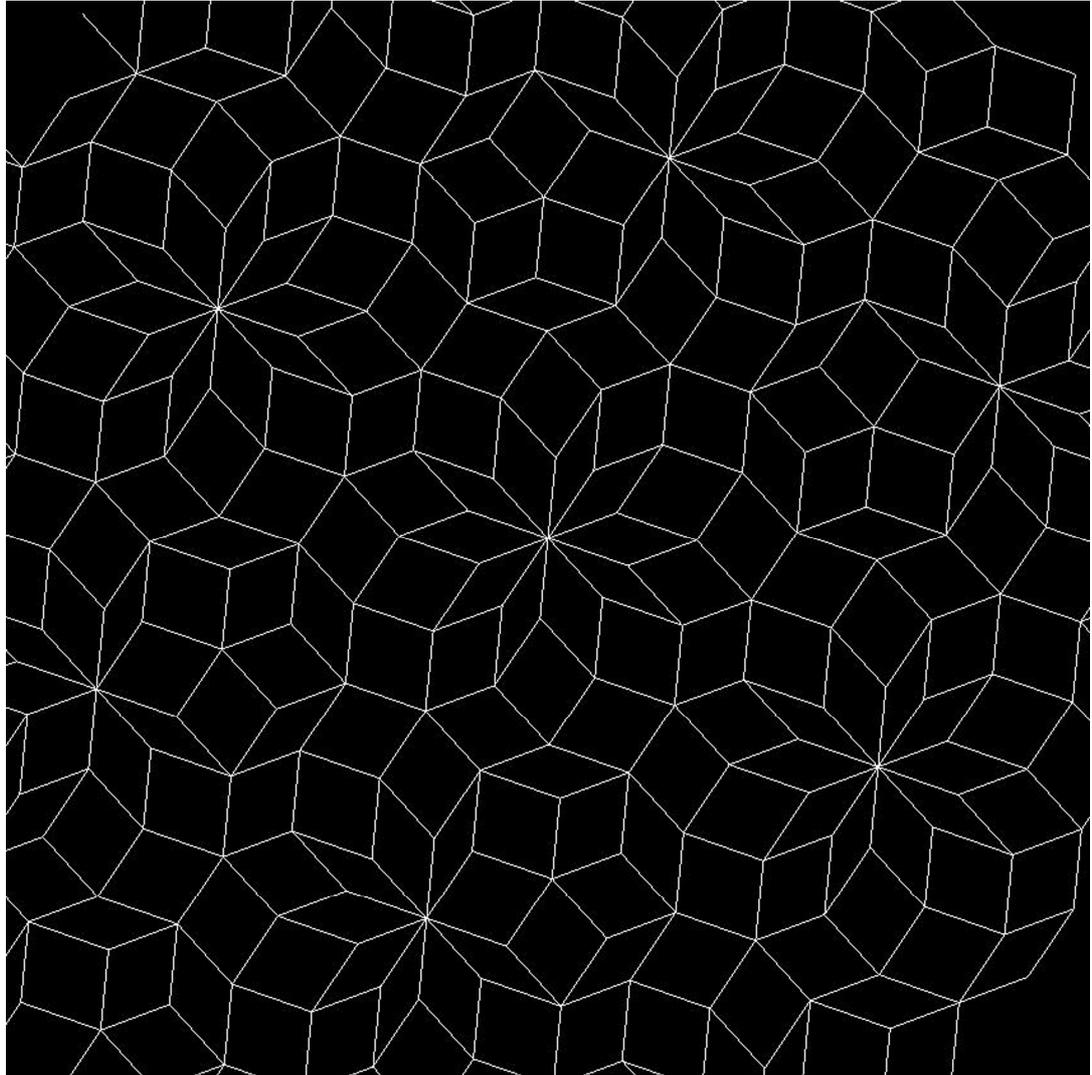
D1. 4d to 2d



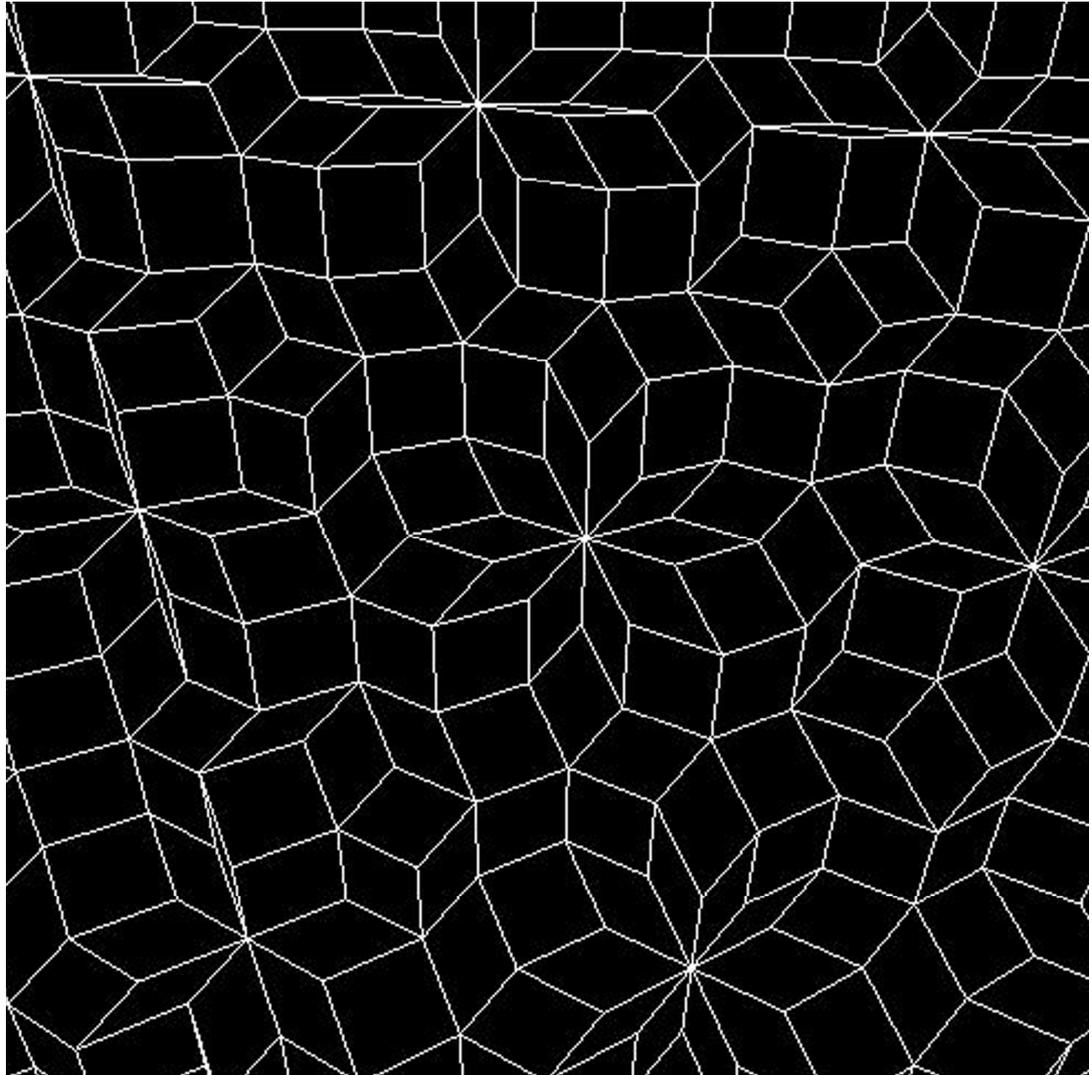
D2. 4d to 3d



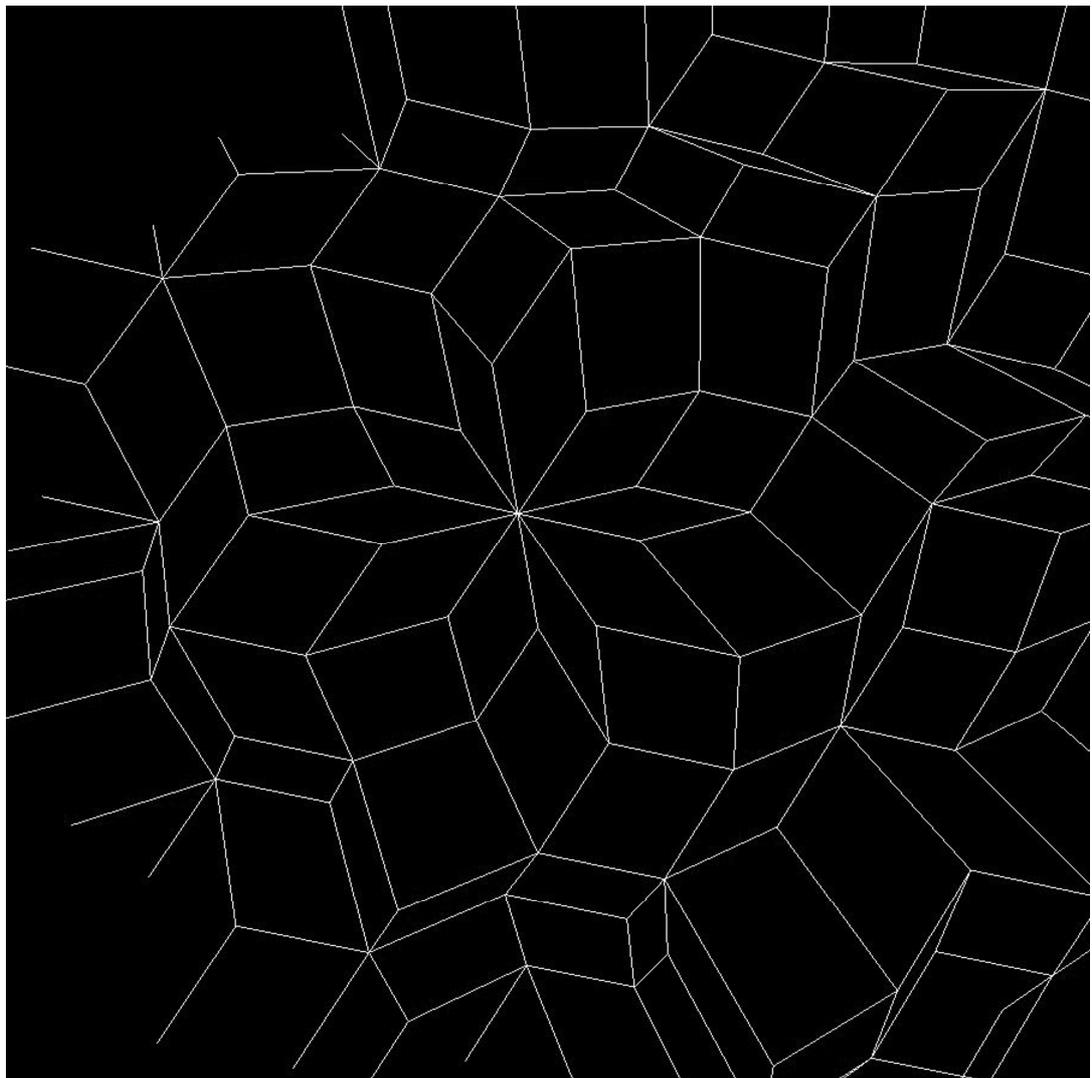
D3. 4d to 2d NURBS adaptation



D4. 5d to 2d

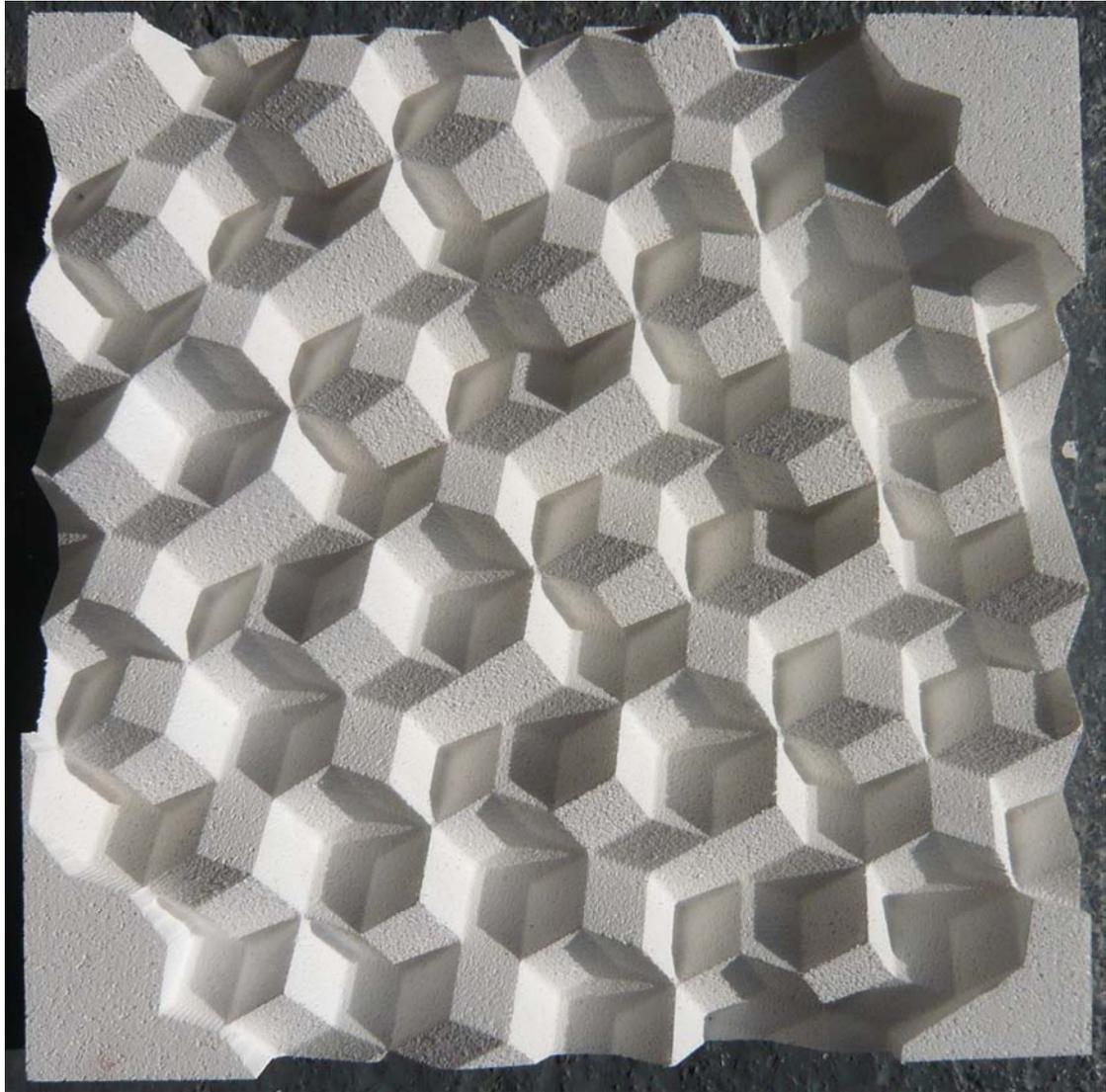


D5. 5d to 3d



D6. 6d to 3d

Appendix F: *Fabrication of tiles' field, formwork and cast*



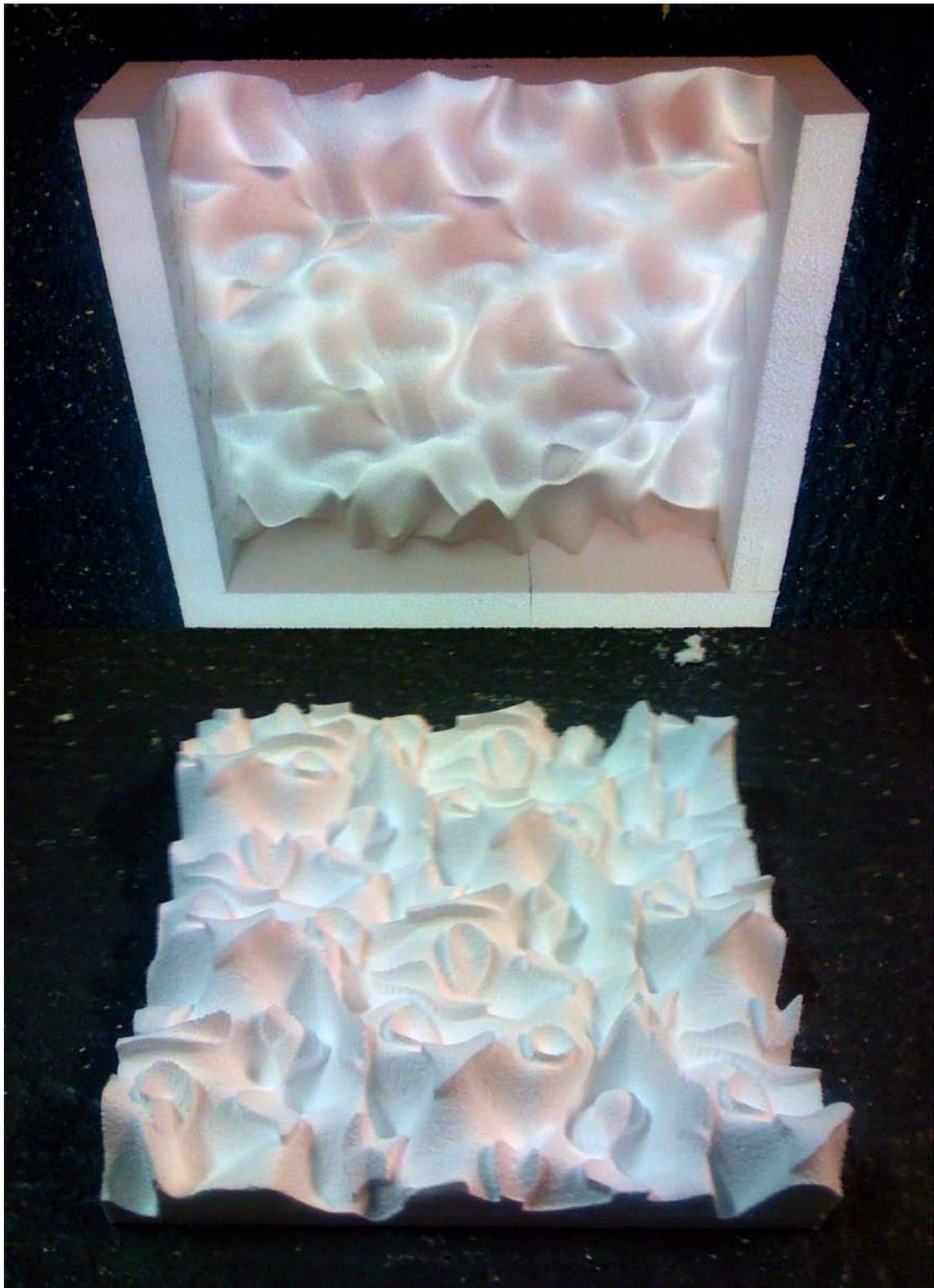
F1. 3d projection of a 4d tiling
Dimension: 500x500x90



F2. NURBS field from 2d projection of 4d tiling (z parameters only)
Dimension: 500x500x90



F3. NURBS field from 2d projection of 4d tiling (x, y, z parameters)
Dimension: 500x500x90



F4. Two-part mould (negative) for a 4d NURBS porous field
Dimension: 2x (500x500x90)



F5. Three cast tiles assembly from 4d tiling with working tangencies
Dimension: 1100x400x300



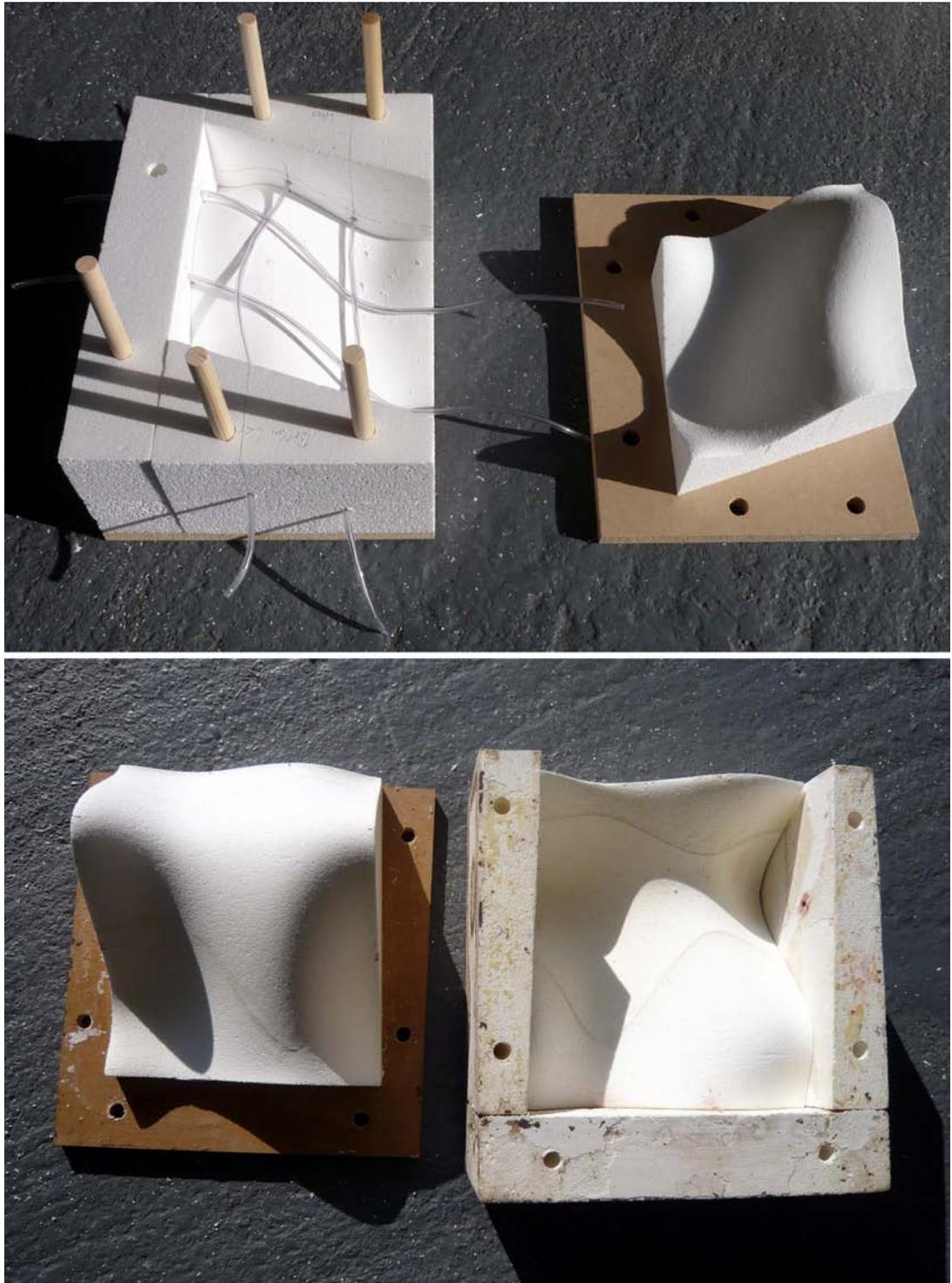
F6. Six cast tiles assembly from a 4d tiling with working tangencies and cables through
Dimension: 1400x700x300



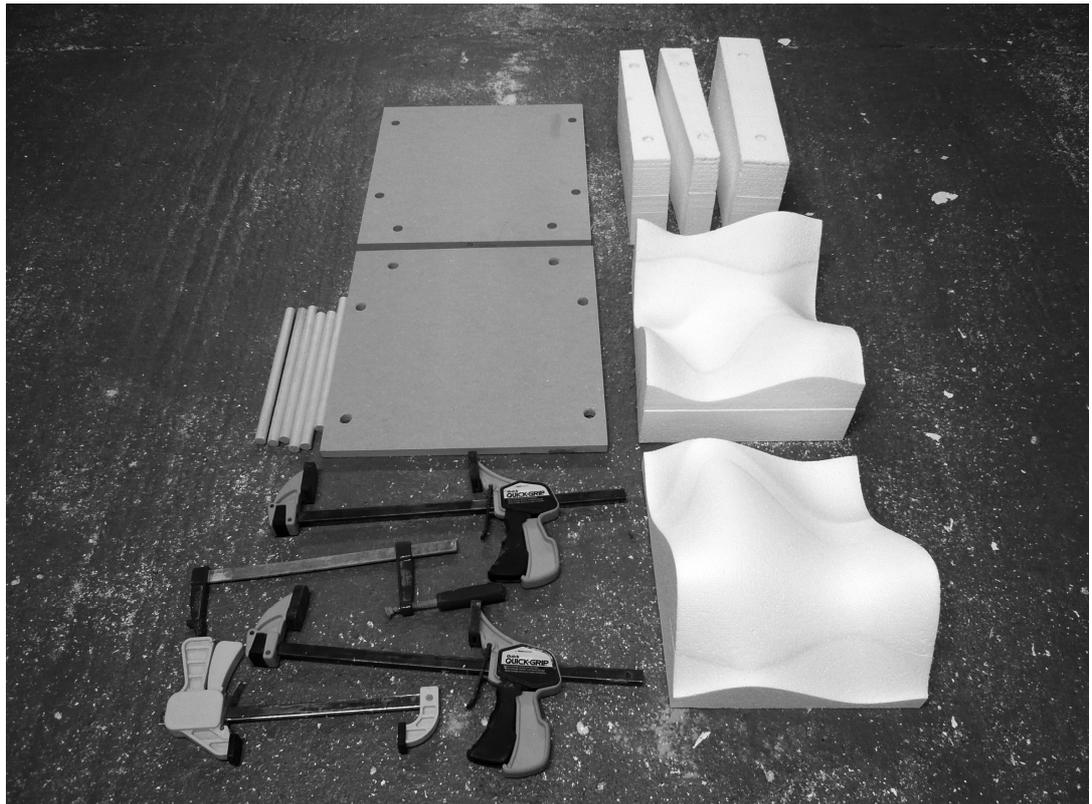
F7. Example of another tile's formwork



F8. NURBS tile from quasi cylinder; one bed and positive tile for fibreglass mould



F9. Examples of formworks for other shapes of tiles



F10. Formwork before and after casting



F11. Ready formwork for casting