Responsive spatial growth of the Danzer packing

Charis Charalampous

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

Bartlett School of Graduate Studies
University College London

September 2007

Abstract

The project is an exploration on aperiodic spatial tiling as for its capacity to be used as a form generator. A responsive system is developed and studied, in order to deliver varying densities of tetrahedral tiles that compose the space frame structure produced. The method is a study on the Danzer packing, and provides a process for the adaptive growth of the packing, considering potential implementations in the architectural and the sculptural scale. The growth of the mathematical formula to produce the tiling is constrained and influenced by the introduction of a simulated visual environment that results in the production of emergent forms. The produced tilings are assessed as for their capacity to respond to the stimulus, by producing a gradation of scales of tetrahedral tiles, as a response to environmental or spatial constrains. Issues of perception and articulation of hierarchical emergent space are raised, and are approached through two different connectivity patterns, the tetrahedral tiling and its dual centroid connectivity of adjacent tetrahedral.

Words: 9.856

Contents

Acknowledgments

With many thanks to my supervisors Sean Hanna and Cristian Derix for their invaluable guidance, inspirational discussions and encouragement; Chiron Mottram for his contribution in some nodal points of the development of the program; Alexandros Tsolakis for endless conversations on the project and helping me to channel my ideas; Pavlos Antoniou for helping me to comprehend the logic of computational structures throughout the year.

Most of all I would like to thank my family for their constant and unconstraint support.

List of Illustrations

List of Tables:

1. Introduction

A reference point for the consideration of any manmade structure is nature and the formations it suggests. Natural growth is a result of the interaction of lower order units that cooperate both among themselves and with the environment. A set of genetic rules and the responsive growth to external stimulus sets the framework of natural growth. Artificial growth is a field of research for the production of industrial materials, and is only recently led into larger scales of constructions (Weinstock, 2006). This shift calls for the definition of the rules and the environments that would guide the growth of such structures, as to generate sufficient emerging configurations in response.
The form finding abilities of natural formations have been studied extensively throughout the 20th century. Moving along that path of thought Frei Otto and Buckminster Fuller, were among the pioneers on the interpretation of nature in the architectural scale, through their theoretical and build projects. The driving force for their experimentations on organic form was the benefits deriving from the dynamic equilibrium that is a result of considering these structures in their wholeness, as a continuum.

The interaction of cells as a result of the parallel dynamic action of mechanical and chemical forces, has been studied thoroughly by D' Arcy Thompson (Thomson, 1961), which provides the guiding principles for the processes studied. In his studies natural formations are studied throughout scales beginning from the microscopic chemistry and physics of cells, all the way to the higher order entities within the organism in an interdepended manner. Local rules among cells provide self organising arrangements that adopt to their environment in the best possible manner. A diversity of layered properties is developed, such as optimal stability and stiffness of the organism, and they are revised as the organism grows or the environment of growth is changed. In that sense the formations are emergent depending on the interaction of entities in a systematic and dynamic way. Anisotropy is a product of the responsive growth of cellular materials, such as bones, that in different instances of their life rearrange the densities and scales of their cells as to better respond to the load conditions received (figure 1).

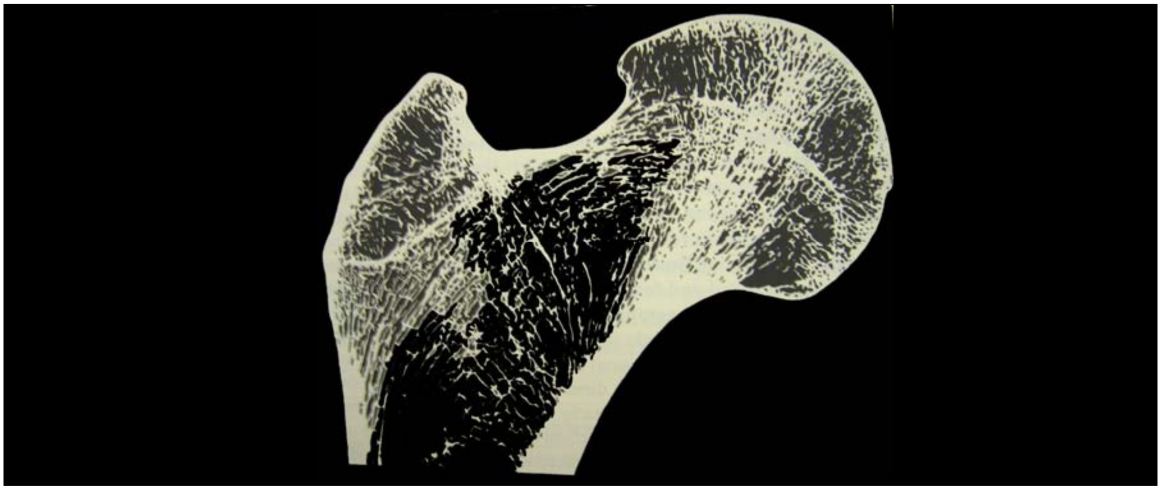


Figure 1. Head of the human femur in section (Thompson, 1961, p. 232)

Whereas the structural consistency of such structures is conceptualised as the quest for minimal energy of the system, the spatial meaningfulness of such structures and their ability to respond to certain environmental constrains is not as apparent and unique. Throughout our research issues of articulation and perception are approached as factors that can help the meaningful use of cellular structures. Spatial tiling is a process of growth and a multilayered event that we foresee to decode as to manage to extract the most out of its inherited properties in all levels of consideration. In the background cases studied diverse aspects of these organic forms are systematically approached in order to identify their affordances to produce the framework for the production of a responsive form generator. The sculptural is studied in parallel to the architectural since the two aspects are interlinked and they inform one another in the case of space frames. The final product of our research is a computational method for delivering aperiodic tilings, deriving from the Danzer packing, as responses to environmental, spatial and aesthetic constrains.

2. Spatial tiling

Two interlinked instances of cellular structures in nature are crystals and quasicrystals, that both have the property of flat faces and precise geometrical shapes, in contrary to other amorphous natural formations. Crystals refer to periodic structures were the composing atoms are arranged in a definite pattern, whereas quasicrystals are aperiodic ordered structures with patterns that do not recur at precisely regular intervals (Brittanica). Such formations can be described in mathematical terms without any physics and chemistry, due to the fact that their physical and chemical background generates rigorous long range periodic or aperiodic structures (Boroczky et al. 2000). Spatial tiling is a most current field of search on structures that are consisted of units that resemble cells or foam bubbles, assembled in a periodic or aperiodic way. The space filling abilities of these structures deliver space structures through a set of rules, which guides their growth as organisms, producing spatial tilings that fill space without gaps or overlappings. Since the process of their growth is rational and can be described mathematically they can become a sufficient alternative to modularity in space frame structures, allowing lattices to emerge as organisms rather than as an assembly of isolated composing members.

2.1 Periodicity through the Weaire Phelan partition

A built example that managed to introduce spatial packing as a self-sufficient framework of expanding a structure is the National Aquatic Stadium in Beijing for the 2008 Olympic Games, by PTW architects. In that design the Weaire Phelan unit cell was used to produce a periodic tiling that created a self supporting structure to shelter the aquarium. The certain partitioning of space is the most sufficient way of tiling space with minimal surface tension and delivers a group of cells with equal sizes. The initial topology is derived from the periodic cubic clathrate structure Na_8Si_{46} (Weaire and Phelan, 1994). The result is a periodically repeated unit of 8 cells of 2 types, 2 of which 12-sided and six 14-sided polyhedra. Those unit cells interlock without gaps when repeated to infinity. The structure for the Aquarium is the result of rotating a 3d array of such units and introducing certain cutting planes (figure 4) that produce the shell of the Watercube. The structure provides a sufficient envelope that both encloses and structurally supports itself without any further structural elements.

Figure 2. The Watercube, or the National Aquatic Stadium in Beijing (Image from http://centripetalnotion.com)

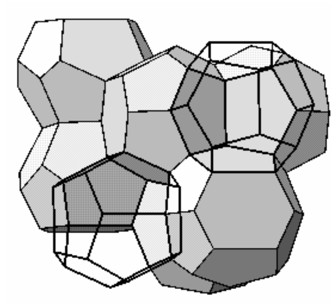
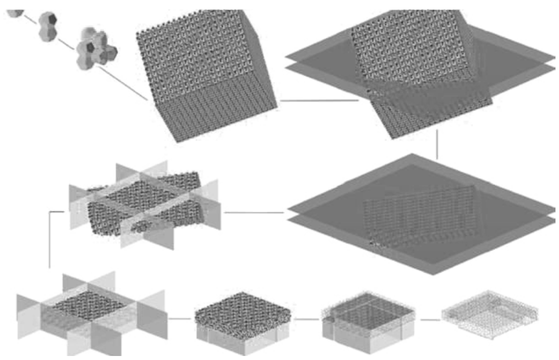


Figure 3. The Waire Phelan unit (Image from http://www.steelpillow.com/)



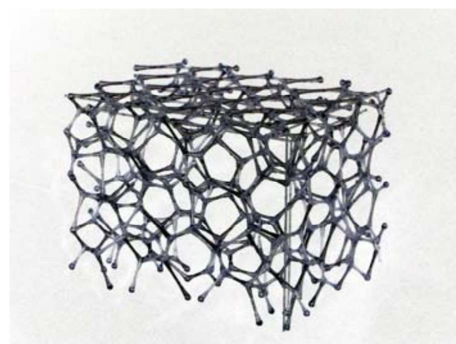


Figure 4. The rotation of the tiling for the Watercube, and the introduction of the cutting planes, followed by a model of the structure (PTW Architects et al., 2007 pp.71,72)

Due to the structural efficiency ensured by the unit of the tiling and the periodicity of the pattern, the geometry of the nodes could not be revised in any of the stages of the design. The unit cell was the final reply, both as topology, connectivity of the nodes, and as geometry, position of the nodes, and didn't afford further manipulation. A much more flexible method of considering spatial tiling is through aperiodicity. In such a tiling there is no transitional symmetry of cells, but rather a composition of prototiles is produced by local rules that govern the overall growth of the pattern. In that manner it produces higher order symmetry, as seen in quasicrystals.

3. The Danzer packing

Though quasicrystals can be obtained by different methods of projections of higher order grids, they can also be described by matching rules of prototiles. The existence of matching rules as to guide growth produces entities that are hard to describe in advance, even though they result from a rational method. The resulting structure is an emergent entity that can only be described and defined through the process of growth.

While considering aperiodic tiling in 2 dimensions, the Penrose tiling is a well known pattern that results from combinations of two golden rhombohedra. The tiling is delivered after the definition of two tiles, the kite and dart that are combined through a set of matching conditions (Grünbaum and Shephard, 1987). The existence of the matching conditions is the creator of aperiodicity, since they restrain connectivity and produce long range order while tiling a plane.

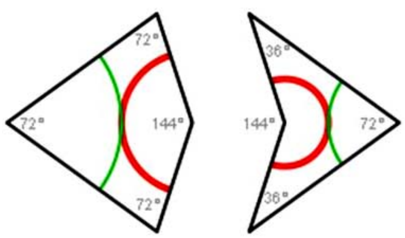


Figure 5. Kite and dart tiles (Image from www.wikipedia.org)

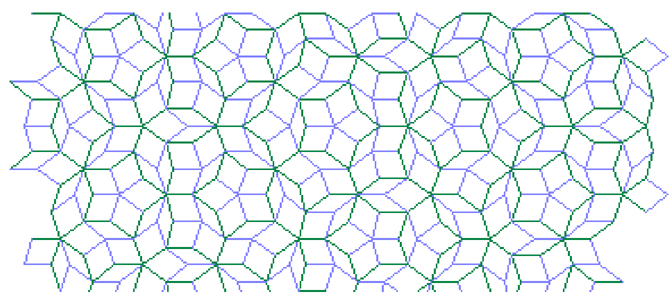


Figure 6. A Penrose tiling (Image from www.wikipedia.org)

Danzer carried out research in defining a set of prototiles that according to some matching conditions give a 3d analogy of the Penrose tiling, admitting quasiperiodic- aperiodic symmetry (Danzer, 1989). His believe for local conditions is a result of the "projection method" developed by N.G. de Bruijn, in which cubic lattices in higher dimensions give nonperiodic tilings with golden rhombohedra.

Figure 7. The 4 Danzer prototiles A, B, C and K (all pictured in the same scale), followed by a face image of tile B subdivided in the 5th level (Shea 2004)

The 4 Danzer prototiles (A,B,C, K) are a subset of the fifteen tetrahedra which are derived from the platonic icosahedron. The matching rules proposed can also be described as subdivision rules of the prototiles, each of which is subdivided in a finite number of prototiles. The process is a result of an iterative method, in which every next generation produces prototiles all of which in a scale τ^-1 (where τ is the golden ratio) in relation to the initial tile to be deflated. Matching conditions for the produced tiles are based on mating identical faces so that the product has no gaps or overlapping.

3.1 Previous computational implementations of the Danzer packing

The iterative nature of the certain packing was considered and analysed through its computational implementation by Dr. Cristina Shea, and ARUP's AGU, who worked with the shape grammar formalism as to implement the tiling (Shea, 2004).
The shape grammar was defined by the 4 prototiles, which were labelled as for their type (A,B,C,K) and their ordered vertices (1,2,3,4), by the matching rules for each prototile, and by the initial prototile (of one of the 4 types) that is selected each time as to initialise the iterative process. The deflation of the Danzer packing might be carried out to the desired number of iterations, thus the desired proportional size of tiles produced in comparison to the initial tile. By deflation we denote the process of decreasing the size of the tiles and then composing them into tiles of the original size (Grünbaum and Shephard, 1987).

For the articulation of the program a solid modelling boundary representation is used, which facilitates the consideration of the prototiles as units that can be further compared and analysed as such. Through this representation only one subdivision rule is needed as to produce additional points on the 6 edges of the tetrahedra, in cutting proportions according to the golden ratio, and the connection of these points produces the deflated tiles. The product, if the iterations are carried out with all rules activated, produces a fractal since it exhibits self similarity in all scales.

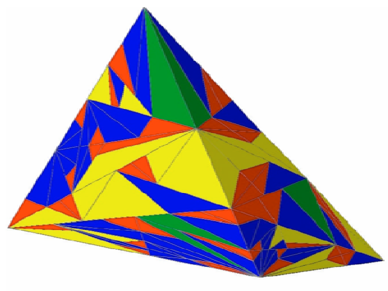


Figure 8. A prototile of type A subdivided by random rule application to level four (Shea, 2004)

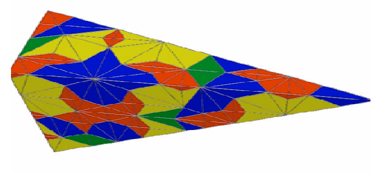


Figure 9. A morphed prototile of type K subdivided to level five (Shea, 2004)

Following the implementation of the tiling, its parametric nature was then considered as to increase its generative power. The existence of rules and the manipulation of the initial prototile to guide the process were considered. The rules for the deflation of certain types of tiles are deactivated or activated after a number of iterations, or in a random manner. In a different case the initial tile was morphed, by changing the geometry of the vertices but maintaining their topology. Another instance was to randomise the proportions for the cuts in consequent iterations as to produce tiles of random sizes and geometries.
The paper provides an initial step in considering the affordances of the aperiodic tiling, and seeks the definition of a process to rationalise the expansion of the tiling as to adopt to some kind of real life scenario, rather than producing variation through randomness.

3.2 The Grotto Project

The computational implementation of the Danzer packing by ARUP's AGU, was used for the design of the Grotto project by Aranda and Lasch. The vertices of the tiling in the 7th iteration were used for the creation of a Voronoi pattern, with planes that bisect perpendicularly the edges of the tiles (Aranda/ Lasch and Bosia, 2007). The resulting boulders were found to be of four types, and interlocked forming 3d formations. Space is produced by excavation on the structure of the Voronoi cells produced, and resembles a cave. The result is a wildly ordered three-dimensional pattern that never repeats the same way twice. For the construction of the Grotto, the boulders were to be cut out of foam and assembled together with steel reinforcement between connecting faces.

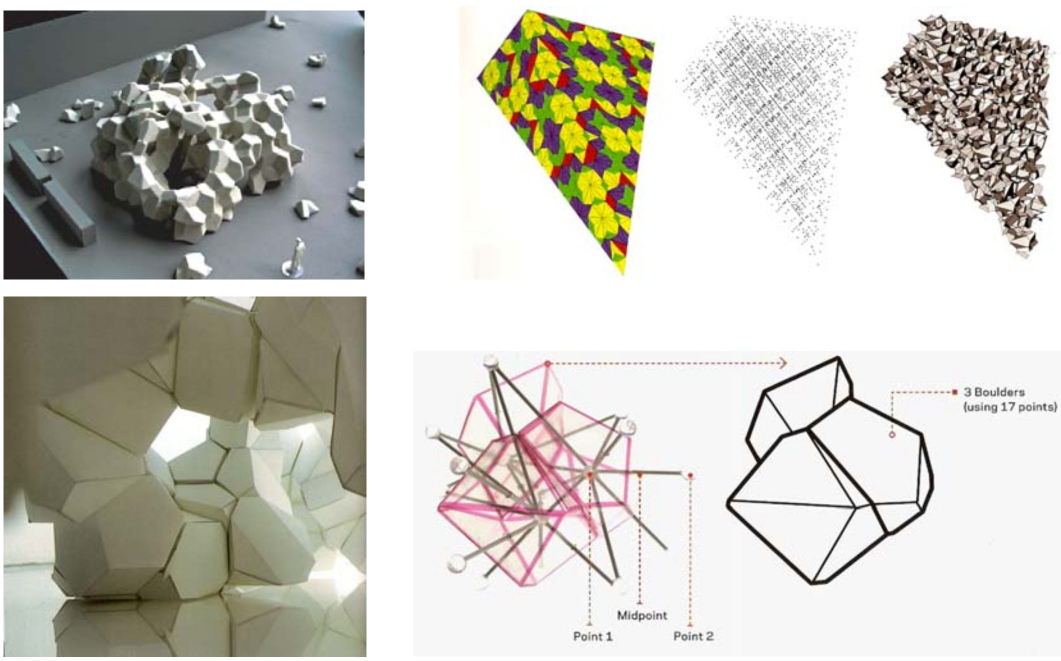


Figure 10. The Grotto project by Aranda and Lasch, deriving from the Danzer packing and the production of the Voronoi cells by considering the vertices of the Danzer tiling (Aranda/ Lasch and Bosia, 2007, pp. 7-14)

The resulting structure was based on defining the higher order formations, as sets of a finite number of boulders, and removing elements that are found in-between these formations. The design task was to excavate the result of the growth rather than guide its formation. All of the boulders were of the same scale and so spaces of varying sizes were a result of the designer's decision on creating an artificial articulation of hierarchical spaces at a later stage, by removing tiles systematically. The boulders are used as building elements, as bricks, rather than as spaces to enter, and so define space as their negative.

4. Towards perceivable and naturally ordered spatial tilings

In order to approach the properties that characterise a rational space frame structure, that responds to a set of constrains, and draw inspiration for the articulation of our program, we investigate former explorations on the use of space frame structures. We attempt this through two different aspects, the first seeking to support scenarios that would result to spatial or habitable entities, and the second foreseeing the production of a perceivable and meaningful space frame. The certain realisations are made in the process of defining the certain vocabulary that guided the structuring of the responsive system we developed.

4.1 The relation of the structure to the architectural scenario

Studies upon the ability of space frames to host human action were held by J. Francois Gabriel. As mentioned in the "Beyond the Cube" (Gabriel, 1997), he derives to his realisations through the consideration of modular spaces of repeated polyhedra which he combines in order to produce rational entities, rather than considering spatial tiling in advance. His realisations are quite relevant and still valid, even though mass customisation makes us rethink the need for modularity (Branco, 2003).
The cornerstone of his research is the definition of habitable space that is described by its capacity to enclose, to provide horizontal planes used as floors and both vertical and horizontal passages as a connectivity system. We chose to consider the results of his study as the basic features that should be met by any emerging form in the architectural scale.

Some basic realisations are the potentials of producing spaces that go beyond the orthocanonical system, and reconsider vertical walls and planar roofs. Horizontal floors are needed in the case of rooms but don't have to keep their horizontality in the cases of corridors or spaces that intent on hosting events. Orientation is a matter of interpretation since space frames afford different positionings in respect to the ground. The analogies of space are thought to be an element that could help the meaningfulness of discrete cells. Narrow spaces can be thought of transitional or spaces for isolation and rest, whereas wide cells after a narrow passage can be the points of concentration or surprise. The third dimension, the height is also a crucial element for the perception of space since it creates the

void on its entirety. The articulation of these spaces can be seen as the way that the story unfolds, having a linear symmetric or a networked expansion, connecting an hierarchy of spaces or composed of a random superimposition of such cells.
Sideways he acknowledges the need to consider different scales of unit cells, when talking about an actual architectural scenario. Anisotropy is also considered to help the structural efficiency, thus proposes the elimination of elements that disturb the articulation of space, when considering the formation as an organic whole. The synergy of elements is highly regarded and proposes endless combinations and emergent self-supportive structures.

Since space frames are the result of a strategic and discreet process, he suggests that human movement within should help the viewer to decode and perceive this underlying logic, and in that sense he highly regards nodal points of tension and density as basic elements of the decoding. Nested scales are last considered to help the synergy of space frames, since they can provide different structural entities in the expansion of the space frame.

4.2 Meaningfulness in the sculptural scale

A space frame structure, as a set of interconnected elements, draws its expressive power from its ability to reflect its connectivity mechanism on the resulting structure. In the case of an organic formation, the unveiling of the synergetic process unifies and bonds the experience of the structure.

An artist that managed to approach the experience of quasicrystals and the fourth dimension through his artworks is Tony Robbin. Through his works he foresees the introduction of modern geometries that could manage to revise the way we design, perceive and thus conceptualise and experience architectural space. As the artist mentions "…artists, architects and engineers owe it to our audience to visualise the four- dimensional, hyberbolic, fractal, and quasicrystalline world that we are really living in" (Robbin, 1997, p. 428). On the cornerstone of his arguments towards this new spatial experience is multiplicity that derives from structures that are not rectilinear and repetitive and produce varying patterns according to the angle of view, thus the position of the human body towards them. His projects refer to the fourth dimension, the space of four mutually perpendicular axes, and the deformation of the projections of the 4d polytopes when rotated in 4d space. Quasicrystal structures are used by the artist since they are the projection of 4d objects to the three dimentional space we inhabit. His Coast sculpture, constructed in 1993 as an installation, is a study on the multiplicity of the projections of quasicrystalline structures through the movement of the viewer and the movement of the sun. The sculpture was positioned in a junction of existing flows in the Center of Art Science and Technology at Denmark's Technical University, as to benefit from the varying directions of movement, thus angles of view and conceptualisation of the structure.

His work is a study on perception, as the experience of a generated quasicrystal figure that unfolds through the movement of the viewer, alongside the space frame produced.

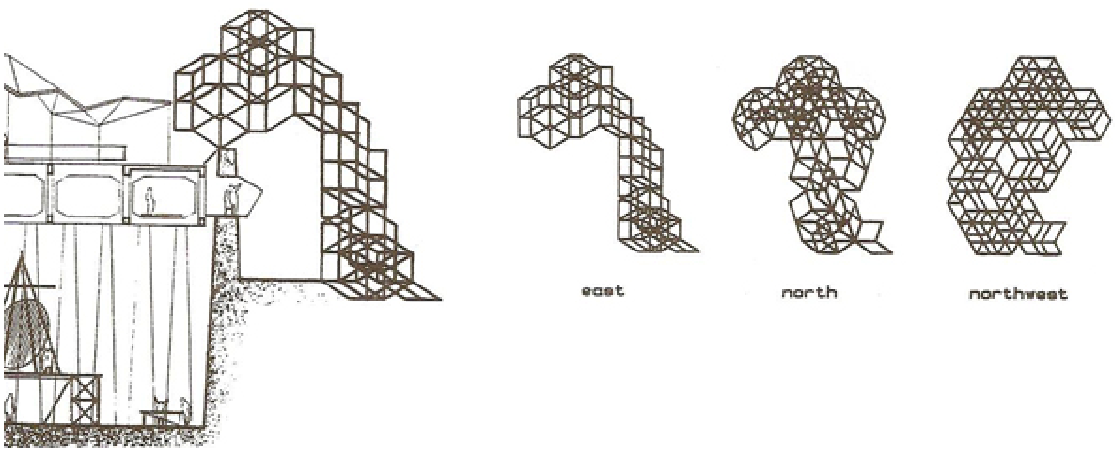


Figure 11. Coast project, by Tony Robbin (Robbin, 1997, p. 438)

A somewhat different approach to the use of lattice structures in the sculptural scale is observed in Antony Gormley' s recent projects, in which space trusses are used in order to describe the human body. His work proposes that an organism, is better understood when represented through a process of organic growth. As the sculptor mentions, all he does is showing the space that the figure was, not representing the body itself (Gormley, 2007). Interconnected elements are used to produce an emerging form through synergy and parallel visual and spatial considerations. His figure is a product of emerging zones within the tiling, and is hard for the viewer to distinguish if the figure adapts to the tiling or the tiling adapts to the figure.

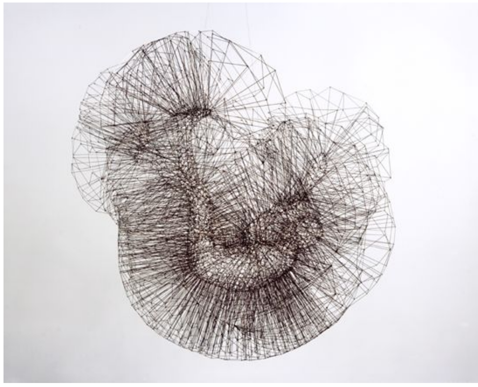


Figure 12. Flare project, by Antony Gormley (Image from www.antonygormley.com)

Figure 13. Body Space Frame, by Antony Gormley (Hanna, 2007)

Features such as varying densities, nodal points and the analogies of cells in his space frame sculptures remind us of the realisations made by J.F Gabriel, as discussed earlier. A main difference is that most of the forms created by the sculptor are based on algorithms of growth, that try to encode these principles within the growing pattern rather than through a manual process of attaching and crafting unit cells. In that sense the former experimentations are closer to the implementation we seek. Algorithms developed by Sean Hanna and Tristan Simmons study the best possible interpretation of body scans, developing methods of placing and articulating points upon those 3d surfaces. In the study for the Body Space Frame varying densities of points are a result of areas of higher curvature of the surface, and so points are generated from these areas and become less and less dense, following a phyllotaxis spiral expansion (Hanna, 2007). At a latter stage points of stress are introduced within the model as to help the decoding of the structure, and define some perceivable areas of tension and higher density within the structure. In the Flare project, the immediate space surrounding the body scan is defined by an offset of the surface and is filled with linear elements that connect the two surfaces. An intermediate stage is the introduction of a Voronoi pattern on the 3d surfaces, which delivers the final connectivity pattern.

In both of Gormley's projects mentioned the basic vocabulary for the interpretation of the figure is formed by the varying densities of elements, the expressiveness of the growing pattern, and the variation of connectivity patterns among areas (figure interior, surface, surrounding space) that manage to produce a perceivable form.

5. Aims and objectives

Based on the general framework presented, a parametric system, that delivers the Danzer packing, is developed as a computational method of responsive growth.

The thesis proposes a method of 3d form generation through the anisotropic growth of a cellular structure. Tropism in natural formations refers to the differential growth within the organism as to better respond to certain stress or environmental scenarios (Thompson, 1961). A similar process is introduced as to allow the growing pattern to adapt to an appropriate input scenario by delivering a form that responds to the input in a sufficient way. The parametric structure should afford different input scenarios and allow the user to interfere with the growing pattern by manipulating a set of thresholds. The interaction to the environment will produce emerging responses whereas the influence by the user will allow the evaluation of the output in real time and suggest the decisions to be made as the iteration proceed.

The aim is to approach the benefits that the generation of a 3d tiling can have in a spatial rather than a structural context. The method should result in varying densities of cells, through differential growth, as a response to environmental constrains, such as energy saving. It should also produce a 3d recognisable figure if used in sculptural art, and respond to architectural scenarios by producing an emerging spatiality of spaces of varying scales and orientations.

5.1 Methodology

In order to reply to the research objectives, a parametric implementation of the Danzer packing was created on the Processing Beta 0118, and was further expanded as a responsive model of growing the tiling. The method is tested on a set of input cases and is assessed for its capacity to adapt to these scenarios in the most sufficient way, with the least number of iterations so that the pattern doesn't get to dense, and in the most recognisable way in relation to our input. Two different connectivity patterns were tested as for their affordances to approximate the figure and respond to our objectives in either the sculptural or the architectural scale.

5.2 Selecting the certain spatial tiling, the Danzer packing

The certain tiling is selected since it is aperiodic and can be composed of tetrahedral of varying densities, or scales, through deflation. It also admits simultaneous 2-fold, 3-fold and 5- fold symmetries (Kramer and Andrle 2004), that means that from different angles it appears to be made out of right angles, other times of triangles and yet others of pentagons (Robbin, 1997). It is also a system that can be inflated and deflated through an iteration process. In parallel, its mathematical description as a set of prototiles that are deflated through some matching conditions described as subdivision rules affords its implementation as a shape grammar (Shea, 2004), a method that is structured on the relation of shapes that can be easily assessed visually. Finally through an appropriate strategy of selective activation, the subdivision rules can be activated or deactivated in certain iterations resulting in differential growth of the tiling.

5.3 Steps taken for the development of the responsive spatial tiling

In order to approach the problem the initial step taken was to introduce the mathematic formulas in the programming environment using the shape grammar formalism. The tetrahedral tiles were delivered from a set of points and the subdivision rules were to act upon labelled tetrahedral, that would be identified as solids by the program. Following the definition of the tiling the environment to which the tiling would respond was introduced as a set of buffer images in three perpendicular planes around the tiling, representing two side views and the plan of the desirable form to which the tiling would converge by further subdividing in the portion of space defined by the images. The buffer images were selected as to deliver 3d figures that could be perceived and assessed easily and in real time and affect the decisions for the appropriate tuning of the program. Consequent alternative methods were tested as to best possible define which tiles were to be selected for further subdivision as to save computational effort and approximate the figure as accurately as possible. Several runs of the program resulted in subsequent changes on the program and are reflected on the final interface of the program (APPENDIX I).

6. Development of the program

6.1 Defining the initial tile

The schema we use is based on the shape grammar formalism acting on a dynamic sequence of data. A basic step in order to build the initial structure was to encode the initial types of the tiles, as derive from Danzer's paper 'Three-dimensional analogs of the planar penrose tilings and quasicrystals' (Danzer, 1989). The four initial tiles are tetrahedral of types A, B, C and K and are defined through the lengths of their ordered edges. All of these lengths are proportional to the golden ratio $\tau = (1+\sqrt{5})/2$, and the mathematical relations for them are given in table 1.

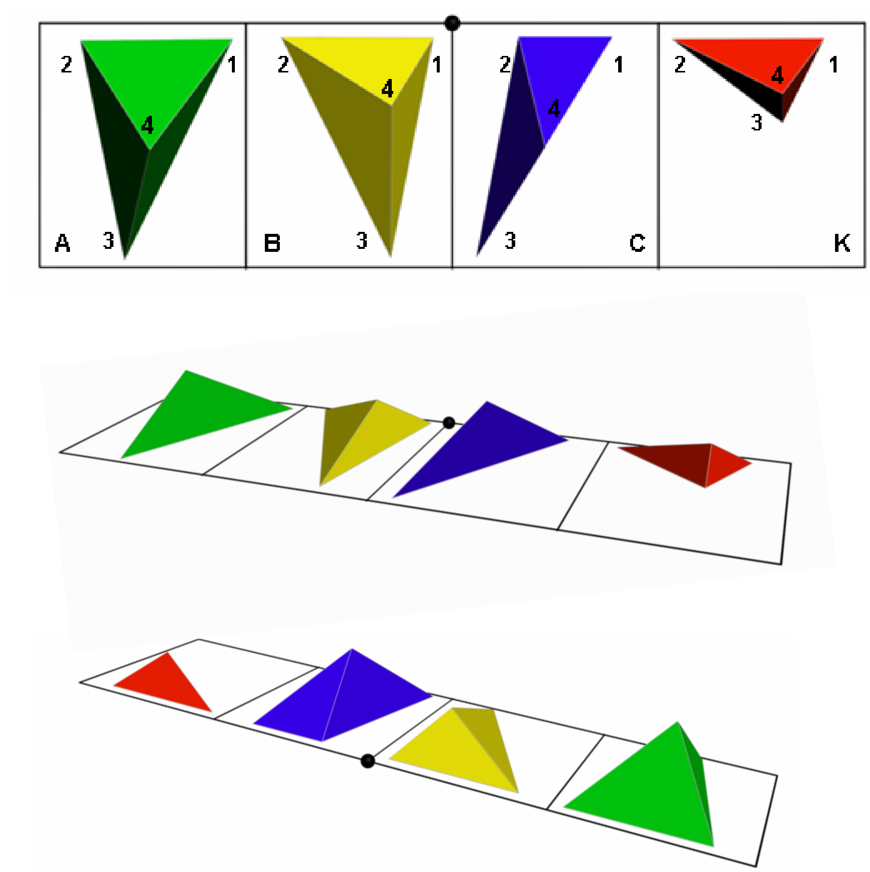


Figure 14. The 4 tetrahedral prototiles of the Danzer packing and their numbered vertices, plan view, SE view, and NW view.

Tetrahedron	Edge					
	1-2	2-3	3-1	2-4	1-4	3-4
A	a	τb	τa	a	1	b
B	a	τa	τb	b	τ^{-1} a	1
C	τ^{-1} a	τb	τ	b	a	a
K	a	b	τ^{-1} a	$\tau/2$	1/2	$\tau^{-1}/2$

Table 1. The lengths of each of the 6 edges of the Danzer tiles, $\tau = (1+\sqrt{5})/2$, $a= (\sqrt{(10+2\sqrt{5})}) /4$, $b= \sqrt{3}/2$ (Danzer, 1989)

The four vertices for each prototile were ordered and numbered correspondingly 1,2,3 and 4 and were calculated by placing the initial vertex on the beginning of our axes, and then points 2 and 3 on the plane z=0 solving the corresponding equations, considering the lengths and the calculated points at each step. The 4th point was found as the peak of the tile and is the only to have a z-coordinate other than zero. The four labelled vertices with their x, y and z coordinates were inserted in an array, which would give the initial topology and geometry for the rules to be applied. An additional element on the array was the type of the tile, 0=A, 1=B, 2=C, 3=K, that would define which set of rules would be applied upon. In that manner the prototiles are described as 3d entities, tetrahedrons that are aware of their topology and type, rather than just a sum of points.

6.2 The subdivision rules

The rules act upon the initial tile by producing proportional cuts on its edges. Each tetrahedral prototile produces a different number of tiles, all of which inscribed within the prototile and all of which in a scale τ^{-1} in relation to the tetrahedral tile that produces them. The rules are to be activated in an iterative process, so at the 5th iteration the tiles produced, if all rules activated will be of scale τ^{-5} (for example in the 5th iteration of the K tile 636 tiles are produced all of which of scale τ^{-5}, 23 of which of type A* τ^{-5}, 182 B* τ^{-5}, 100 C* τ^{-5} and 331 K* τ^{-5}). Following our path of thought, the application of the rule produces an additional number of points on its edges, and the deflated tiles are the product of connecting

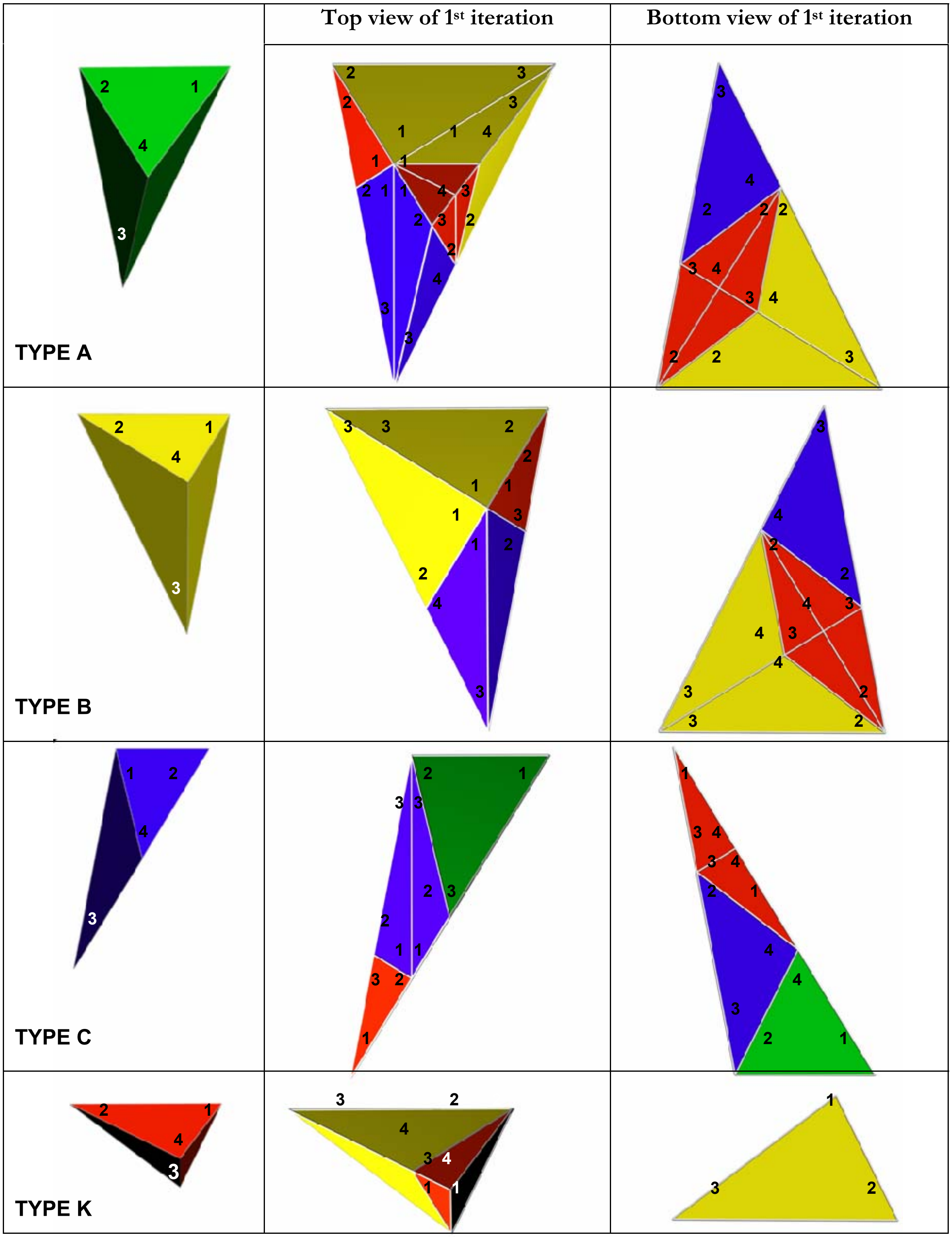


Figure 15. The four tetrahedral in their first iteration, and the reordering of the vertices that their subdivision produces, in order to maintain the topology of the produced tiles

these points. All the cuts are calculated as proportional directed cuts upon the edges, for example point ***V*** would lay between vertices 1 and 2 with proportion ***p*** beginning from vertex 1. The proportions ***p*** for these cuts are one of τ^{-2}, τ^{-1}, $\tau^{-2}/2$, and τ^{-3} in each case. The sum of the resulting points ***V*** i and the initial vertices of the parent tetrahedral are then combined in arrays of four, which describe the deflated tiles. It is crucial to notice that the points for the new tiles must be ordered in correspondence to the numbering of the vertices in the initial tile of that type, so that they keep the correct topology of connectivity and be then subdivided in the desired way (figure 15).

Resulting tetrahedral	Initial tetrahedron			
	A	**B**	**C**	**K**
$A*\tau^{-1}$	0	0	1	0
$B*\tau^{-1}$	3	2	0	1
$C*\tau^{-1}$	2	1	2	0
$K*\tau^{-1}$	6	4	2	1
Total number of Tetrahedral$*\tau^{-1}$	11	7	5	2

Table 2.. The inflation- matrix of the Danzer packing, showing the products of one iteration of each tetrahedron (Danzer, 1989)

Since each tile produces more than one tiles of each type (table 2), each connecting different vertices from the subdivision, we assign the type of each resulting tile as a 3digit number, the first indicating the type of the tile that produced it (one of 0, 1, 2, 3), and the second and third indicating the instance and the type of the resulting tile. Thus, since the tetrahedral tile A, produces 6 tiles of type K in one subdivision, we label the 4th tile of type K as 043 (0=A as the initial tile, 4=the 4th tetrahedral of type 3= K). This helps the draw function decide which vertices of the parent tile should be considered for the creation of each individual child.

6.3 Structuring the iterative process

The data configuration we built is a tree structure beginning with the definition of the type of our initial tetrahedron, thus selecting which of the four arrays of vertices (A, B, C, or K) to iterate. In each iteration the subdivision rules are applied and the produced tiles are described as an array of their vertices.

As mentioned each tile gives a finite number of children, constant according to its type. The first step as to apply the subdivision rules is to consider the existing tiles of the previous iteration, identify them as for their type and apply the respective rule. The Vector class of Java is used as to create a growable array of objects, containing the children of each tile. The Vector contains components that can be accessed using an integer index, and its size can grow or shrink as needed to respond in adding and removing items after the Vector has been created.

According to the type of the children added on the produced Vector, the corresponding draw function in the class Tetra is activated from an *if else* structure, in the form of a *switch*, included in the Geometry class. The new tetrahedron includes the type of the tile to be calculated and the vertices of the parent tile. In the draw function in addition to the four vertices that are selected and reordered, we also calculate the 4 face centres and the centroid of the tile, and also the type of the produced tile (A, B, C and K).

Algorithm for the iterative process (APPENDIX II):

1. In the main: a) define the type of the initial tile

2. In class Node: a) apply the corresponding subdivision rule
b) define the Vector children that includes all the produced tetraherons
c) define which tiles to draw, thus all children that are created in this iteration and are aren't subdivided

3) In class Geometry: a) define which of the vertices produced by the subdivision of the parent tetrahedron should be considered for each child tetrahedron

4) In class Tetra: a) subdivide the parent tetrahedron
b) reorder the vertices produce and define the vertices of each child
c) draw each child tetrahedron

6.4 Defining the environment of growth, and its relation to the tiles

The responsive growth is a product of the selective activation and deactivation of subdivision rules, according to the position of each tetrahedron. Tetrahedrons that are found within a portion of the tiling are deflated, and result in differential growth of the tiling. The portion of the volume of the tiling to be further subdivided is defined as the volumetric representation of a figure, which is composed of three buffer images surrounding the tiling. The images are inserted on the beginning of our axes and are based each in one of the planes x=0, y=0 and z=0. In that sense the resulting area in 3d space is the union of the projection of these three images.

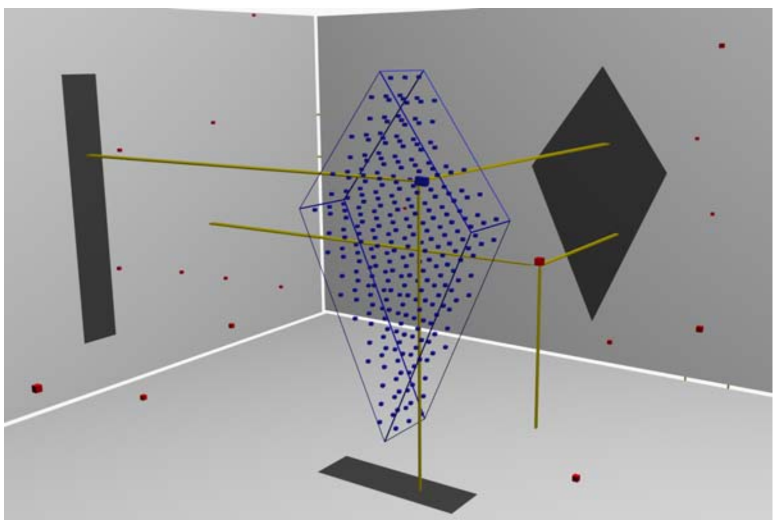


Figure 16. The buffer images that compose the environment of the growth, and the volumetric representation of their projection in space (in blue)

The 3 buffer images are coloured in the same manner, initially in black and white, so the projection of the areas with the same colouring (black in figure 16) defines a portion of space. A predefined number of points fill this area by randomly navigating the space enclosed by the images and gathering in their union (blue in figure 16). Whenever the projection (yellow in figure 16) of a point on the three planes is within the designated area its position is locked and the program proceeds to positioning the next point, of a finite number of points. The selection of which tiles to subdivide is performed by checking the relation of these points to the existing tiles. If one of the points is within a tile then the tile is further subdivided (APPENDIX III).

The certain method of the volumetric representation as a point cloud was selected since it further allows the user to move the point cloud as the iterations proceed. In that manner the method can be used as a real time sketching tool, by allowing the user to visually evaluate the outcome and rearrange the stimulating factor before the next iteration (some results of this part of the method are included in the section 8.1).

The mathematical method for deciding if a point of the volumetric representation is within a certain tile is provided by Gary Herron (Herron, 1994).

At the initial stage we consider the four vertices (V:1, 2, 3, 4) of each tetrahedral tile ***T***:

```
V1 = (x1, y1, z1)
V2 = (x2, y2, z2)
V3 = (x3, y3, z3)
V4 = (x4, y4, z4)
```

In order to check the relation of the tetrahedral to the point ***P***:

```
P = (x, y, z)
```

We calculate the five following determinants ***D*** and if all have the same sign then the point ***P*** is within the tetrahedral ***T***. At the same time if any Di=0, then P lies on the boundary of ***T***.

$$D0=\begin{vmatrix} x1 & y1 & z1 & 1 \\ x2 & y2 & z2 & 1 \\ x3 & y3 & z3 & 1 \\ x4 & y4 & z4 & 1 \end{vmatrix} \quad D1=\begin{vmatrix} x & y & z & 1 \\ x2 & y2 & z2 & 1 \\ x3 & y3 & z3 & 1 \\ x4 & y4 & z4 & 1 \end{vmatrix} \quad D2=\begin{vmatrix} x1 & y1 & z1 & 1 \\ x & y & z & 1 \\ x3 & y3 & z3 & 1 \\ x4 & y4 & z4 & 1 \end{vmatrix}$$

$$D3=\begin{vmatrix} x1 & y1 & z1 & 1 \\ x2 & y2 & z2 & 1 \\ x & y & z & 1 \\ x4 & y4 & z4 & 1 \end{vmatrix} \quad D4=\begin{vmatrix} x1 & y1 & z1 & 1 \\ x2 & y2 & z2 & 1 \\ x3 & y3 & z3 & 1 \\ x & y & z & 1 \end{vmatrix}$$

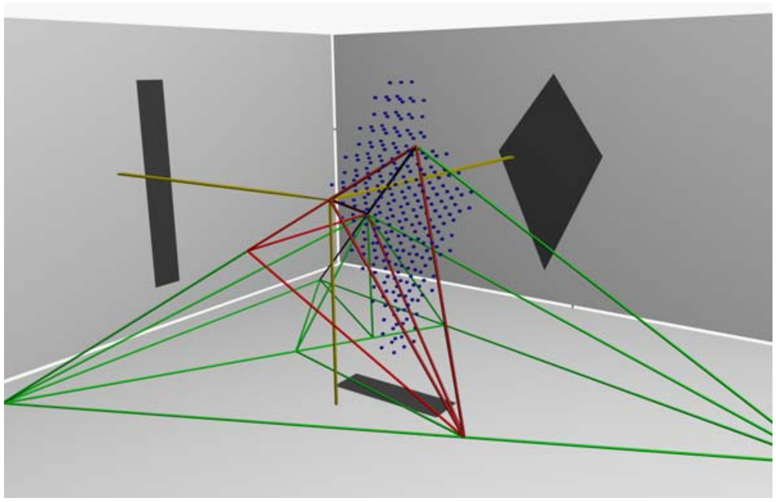


Figure 17. The volumetric representation as a point cloud is the stimulus for the adaptive growth of the tiling

A different method of defining whether a tile should be further subdivided can be used after a certain amount of iterations, when the deflated tiles are of a certain small size. At that stage, since the size of the produced tiles decreases, we can check if a subset of the four vertices of each tile is within the volumetric representation, by projecting (yellow in figure 17) each vertex on the three buffer images. If all three projections of 1, 2, 3 or 4 of the vertices are within the designated area then the tetrahedron is iterated (the check to the point cloud is not used for this step). This method is used in consequent steps, beginning with checking 1 to 4 vertices, as to gradually reduce the portion of the tiling that is iterated each time, and finally consider the tetrahedrons found to be entirely within the volumetric representation.

6.5 The comparison of the data produced

Each iteration produces a new amount of tiles following the data tree of the shape grammar. We only consider and draw the tiles that have produced no children in each iteration, thus the final product of the current iteration, and we store these in a new Vector. These data are then accessed and compared in our main, through a double for loop. The main check is for adjacencies.

In the initial stage the four face centres of each tile are checked against all tiles, and if within the boundary of another tile then the two are considered adjacent. The check is on the basis of the mathematic formula for the check of the relation of a point P to a tetrahedral T, as referenced earlier. The certain method was used, to identify relations among tiles of different scales, since the tiles in this way don't have to have a face of same size in common. If desired following this test we connect the centroids of the adjacent tiles, producing another Vector that includes all these connecting lines.

A second test is conducted within the produced centroid connectivity, and adjacent lines give triangular planes. Any two lines that have a point in common can define a triangle that has for vertices, the common point and the two free ends of the two lines. For each plane we calculate its normal ***N*** as the cross product of the two directed vectors AB= (v1,v2,v3) and AC= (v4,v5,v6), our two adjacent lines (figure 18).

`N= crossProduct ((v1,v2,v3)(v4,v5,v6))= ((v4-v1), (v5-v2),(v6-v3))` (1)

As mentioned earlier the interpretation and the orientation of a space structure is a decision of the designer (Gabriel, 1997). In order to decide on the orientation of the structure we introduce a reference plane ***L*** that will define the horizontal plane for the orientation.

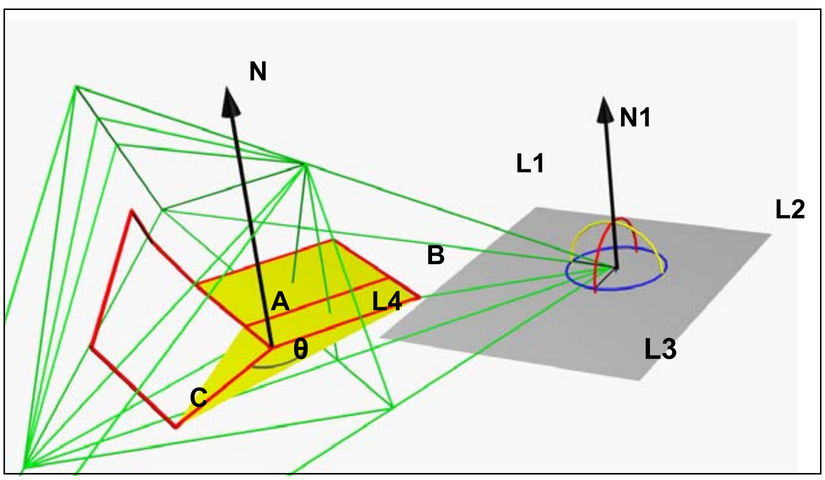


Figure 18. The reference plane L (in grey), that suggests which planes (in yellow) should be drawn

Our reference plane ***L*** is a plane that rotates around the X, Y and Z axes. The plane is a quadrilateral, with vertices (L 1,2,3,4) of fixed y and z coordinates and an x coordinate that is calculated through the calculation of the normal of the plane.

We consider:

i) The normal of the reference plane is the vector N1= (a, b, c) .

ii) The angle θ of two unit vectors AB, AC is the invCosine of their dot product.

iii) The reference plane is directed through its rotation around the x axis by U, around the y axes by V and around the z axes by W.

From i, ii and iii we conclude that:

```
cosU = dotProduct ((a, b, c)(1,0,0)) = a
cosV = dotProduct ((a, b, c)(0,1,0)) = b
cosW = dotProduct ((a, b, c)(0,0,1)) = c
```

So the normal N1 of the reference plane is:

N1= cos(U), cos(V), cos(W) (2)

We can now calculate the angle between our reference plane and each plane, by taking in consideration the two plane normals, ***N*** and ***N1*** given from equations (1) and (2), convert them to unit vectors, and consider their angle ***θ1*** as described in (ii).

The method is based on the selection of which planes to draw in the tiling, through the definition of the thresholds of the angle that they should form with the plane of reference (upper and lower limits).

At the same time the reference plane ***L*** is drawn on screen, by calculation of the x coordinate of its four vertices, according to the plane normal ***N1*** each time, since the dot product of any point ***Li*** of the plane ***L*** with the normal ***N1*** of the plane is zero. The plane of reference is also included in the dxf file, and suggests the orientation of the horizontal plane for the best possible orientation of the structure as to obtain as many horizontal planes as possible from the tiling (APPENDIX IV).

7. Testing and tuning the algorithm

Following the definition of the algorithm to deliver the Danzer packing and the structuring of the method through which the selective activation of subdivision rules would take place, we test our algorithm in a sequence of scenarios. By using a shape grammar the functions refer to relations of shapes that are easily assessed through vision and perception (Stiny, 1980). Throughout these sequential approaches to the algorithm we test the ability of the two connectivities (tetrahedral tiling and centroid of adjacent tetrahedrons) to approximate the input volumetric representation, using the most economic, computationally, process. We also try to develop and test some heuristics that produce a more accurate or delicate result and comment on the output delivered. We further try to approach the centoid connectivity pattern as for the emerging spatiality it produces, and the way it could be rationalised or further used in the architectural scale through the selective introduction of floor planes.

A certain instance of the method might be more appropriate for the architectural or the sculptural, or any other scale. The purpose of this thesis is to develop sufficiently the method proposed, and propose possible implementations that will make use of the spatial affordances and properties of the Danzer packing that are augmented through its differential growth.

Through these experiments we obtain some further suggestions for the proper tuning of the algorithm according to each implementation. The results both informed and refined our algorithm, and revealed some further features of the tiling, which will be further analysed in section 8.

7.1 The case of the cross

For the first step of our experimentation, the centroid connectivity was used. The guiding principle for this decision was that this connectivity pattern is less obvious as a structure, thus less apparent and predictable to the viewer than the tetrahedral connectivity.

The initial experimentations were held with simple figures composed of linear parts. One of the first experiments that indicated certain heuristics that would result to a better approximations of a linear form was the case of the cross figure (table 3). The certain figure was chosen since it creates areas of density and intermediate spaces of gradation between its legs.

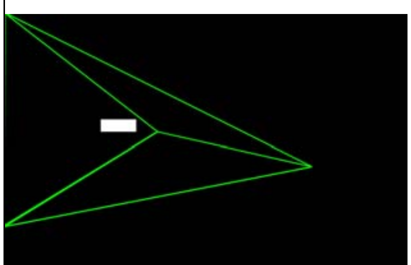
<table>
<tr><th rowspan="2">TYPE</th><th rowspan="2">SIDE VIEW</th><th>FRONT VIEW</th><th colspan="2" rowspan="2">DATA</th></tr>
<tr><th>PLAN VIEW</th></tr>
<tr><td rowspan="5">A</td><td rowspan="5"></td><td rowspan="5">
</td><td>Volume of buffer figure / Volume of the initial tile</td><td>0,6%</td></tr>
<tr><td>Number of points to fill the volumetric representation</td><td>1000</td></tr>
<tr><td>Number of iterations</td><td>9</td></tr>
<tr><td>Number of tetrahedral in the responsive tiling</td><td>7273</td></tr>
<tr><td>Number of lines in the centroid connectivity of adjacent cells</td><td>21103</td></tr>
</table>

Table 3. The data for the definition of the cross

The desirable degradation of densities within the tiling suggests that we should carry the run for as many iterations as possible. In each iteration we have an increase in the number of tetrahedral produced, and a big increase in the number of lines connecting the centres of adjacent tetrahedral. Adjacent cells are identified through a double for loop of all tetrahedral of the same iteration, and so if the number of tiles is 5000, the resulting array of all cases checked is 5000*5000, thus 25 000 000.

In order to proceed to further iterations we kept the number of tetrahedral produced and needed for the approximation as low as possible. The ratio of the volume of the buffer figure to the volume of the initial is at 0.6% in the case of the cross. In parallel the iterations were carried out up to the 5th level through considering the relation of the tiling to the point cloud of the volumetric representation, and then the 6th 7th, 8th and 9th were subdividing all tetrahedral that when projected on the buffer images were found to have 1, 2, 3 and 4 points respectively within our figure. In that way, at the last iteration only the tetrahedral that were totally within the figure were further subdivided.

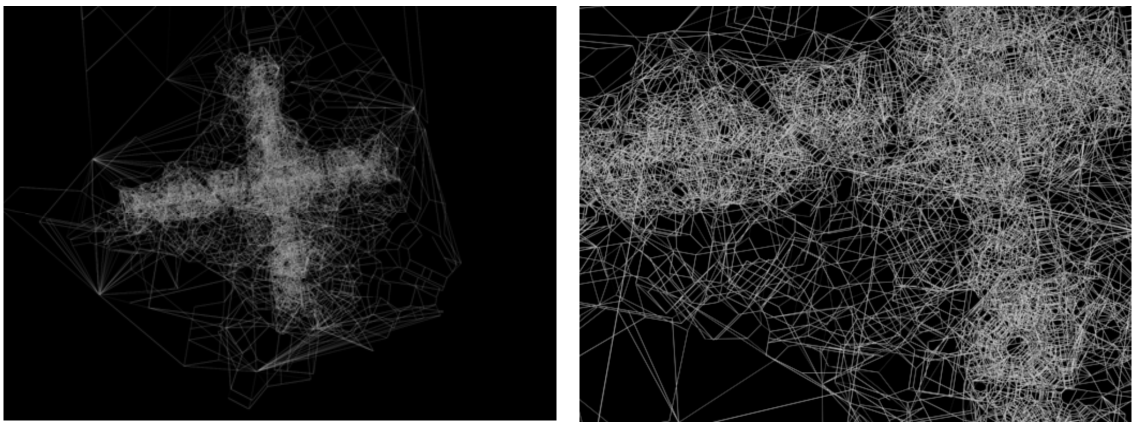


Figure 19. The centroid connectivity of the cross in prototile A

The resulting figure is identifiable and provides the desired degradation of scales (figure 19). The consistency of the tiling is maintained even though the checking principle is changed in the final iterations, and affords and can be carried out for a sufficient number of iterations.

7.2 The human figure

At the next stage of the experimentations the human figure was approached. The buffer images used for this stage derived from Da Vinci's Vitruvian man, as an identifiable figure, and since it is a figure with open ends that would allow the distinction of the areas of density following the example of the cross.

The tetrahedral connectivity of the Danzer packing is maintained here, as to perform further iteration of the tiling, since the adjacency check is not used for this tiling.

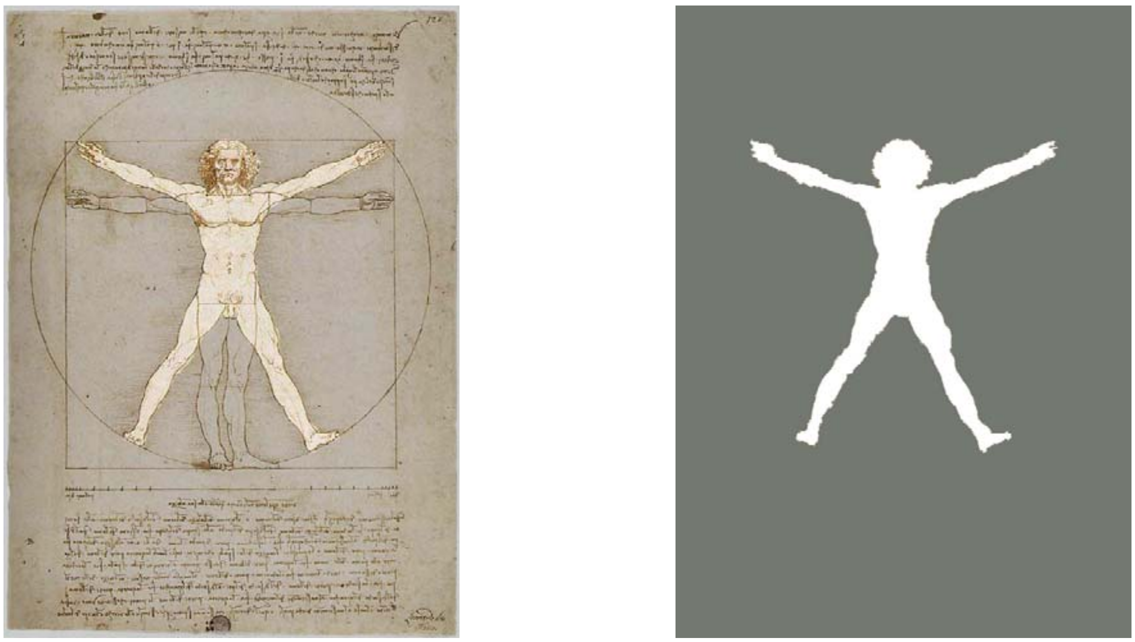


Figure 20. Extraction of a buffer image from Da Vinci's Vitruvian man (Image of the drawing from www.drawingsofleonardo.org/)

In order to decide which tile should be used as the initial tetrahedral for the human figure we carried out a run of 12 iterations four times having as initial tile one of type A,B,C and K each time. The ratio of the volume of the buffer figure to the initial tile was calculated, and we kept track of the number of tiles produced in the 12th iteration (table 6).

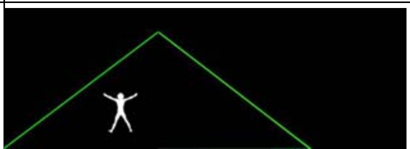
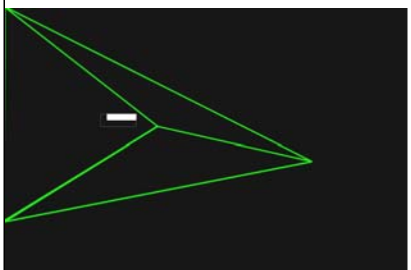
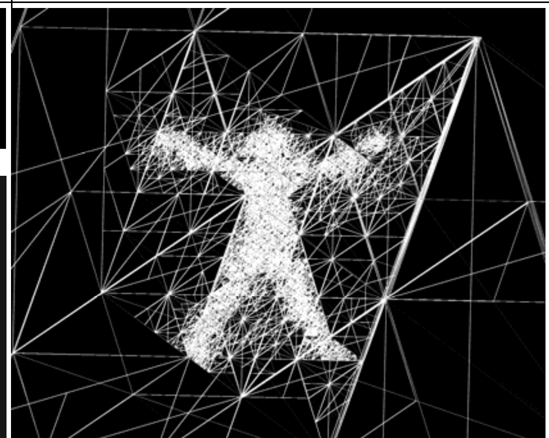
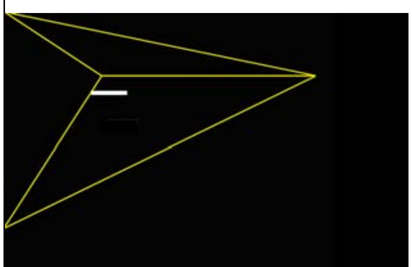
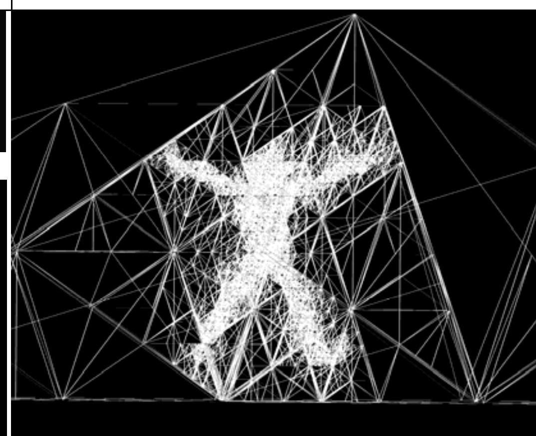
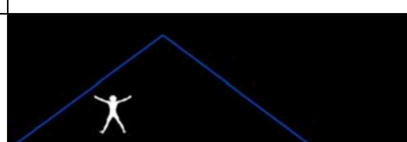
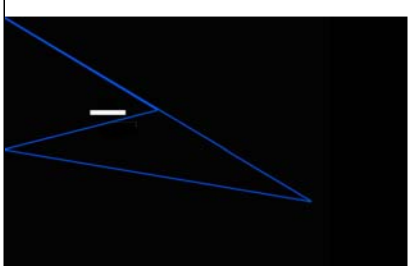
TYPE	SIDE VIEW	FRONT VIEW PLAN VIEW	TILING PRODUCED
A		 	
B		 	
C		 	

Table 4. The tetrahedral tiling of the human figure in prototiles A, B and C

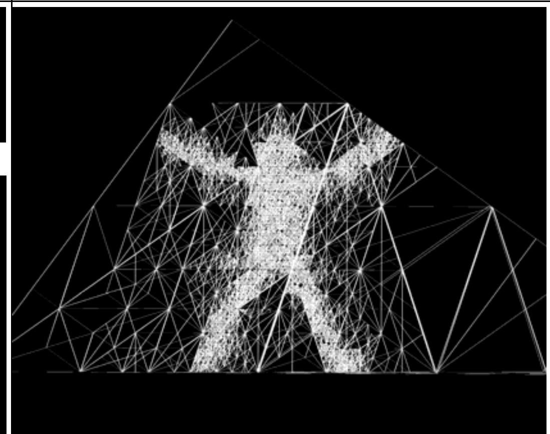
TYPE		FRONT VIEW	TILING PRODUCED
	SIDE VIEW	PLAN VIEW	
K		 	

Table 5. The tetrahedral tiling of the human figure in prototile K

TYPE	Volume of buffer figure / Volume of the initial tile	Number of tetrahedral in the responsive tiling in the 12^{th} iteration
A	0.25%	31656
B	0.32%	31975
C	0.32%	32040
K	1.08%	33411

Table 6. The data from the responsive growth of the human figures in the 4 prototiles

By considering the produced tiling (table 4 and 5) it was observed that the tile of bigger volume gave a better approximation of the form, producing a more gradual degradation of scales. It was also observed that as the ratio of the volume of the figure to the tile increased, the number of produced tiles also increased and the expansion of the tiling was restricted by the faces of the initial tile.

For these reasons we carried on the next steps of the experimentations with the human figure by considering a tetrahedral of type A as our initial tile.

By closer inspection of the produced tilings to deliver the human figure it was observed that there existed areas of great density within the figure. In order to overcome the possible implications of such a density, in the case that such a tiling was to be constructed, we introduce a method to reduce the number of tiles created, but maintain the identifiable figure. The method is hinted through previous implementations on the human figure, found in the work of Antony Gormley as mentioned earlier, which negotiate the existence and representation of a boundary for the figure. The description of a 3d surface gives the boundary, an interior and an exterior space to the figure (figure 12), that receive a different spatial tiling, as to reinforce the recognisability of the figure.

TYPE	FRONT BUFFER VIEW	DATA	
A		Volume of buffer figure / Volume of the initial tile	0.25%
		Number of points to fill the volumetric representation	1050
		Number of iterations	13
		Number of tetrahedral in the responsive tiling	48952
		Number of tetrahedral by removing tiles found in the interior of the figure	40113

Table 7. The data for the definition of the human figure

This step of the method is implemented through the removal of all tiles that when projected on the buffer images are found to be within the figure. What is actually done is that these tiles are not drawn, rather than removed. The iterations still have to be carried out to a certain degree, since gradually, tetrahedral of a smaller scale surround the figure, and so approximate sufficiently its 3d boundary.

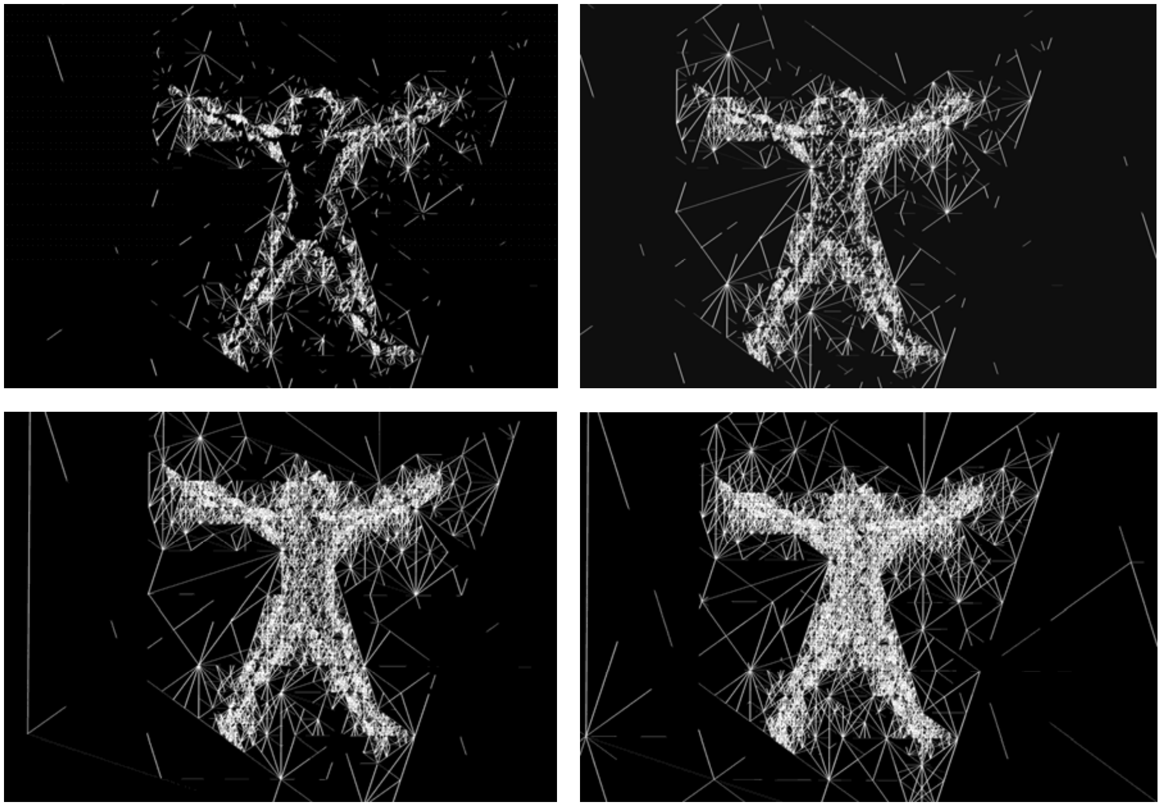


Figure 21. Consequent sections of the tetrahedral tiling of the hollow human figure

The resulting hollow tiling is composed of a reduced number of tiles by 18%, in the 13th iteration of tile A (table 7). The product is closely inspected through consequent renderings that prove that the consistency of the figure is maintained. In parallel we should bare in mind that the tiles of the interior are removed as entities, so they will not disturb the consistency of the tiling, which will still interlock around the figure without gaps.

Another way to approximate a figure, through the definition of degraded images is proposed here. This method further reduces the number of produced tiles, and produces areas of grater density, that reinforce the underlying structure of the form.

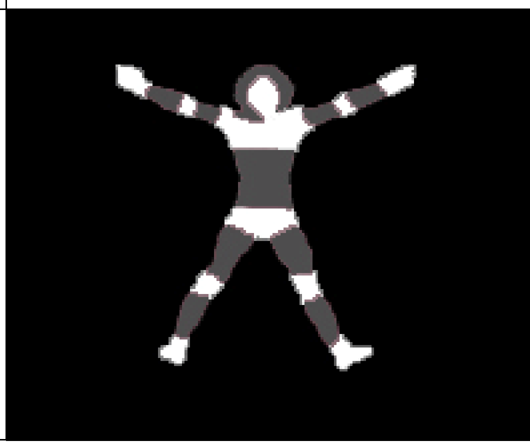
TYPE	FRONT BUFFER VIEW	DATA	
A		Volume of buffer figure / Volume of the initial tile	0.11%
		Number of points to fill the volumetric representation	750
		Number of iterations	13
		Number of tetrahedral when the buffer image tests both the grey and white area	48952
		Number of tetrahedral by considering the nodal points in white	22715

Table 8. The data for the definition of the nodal points of the human figure

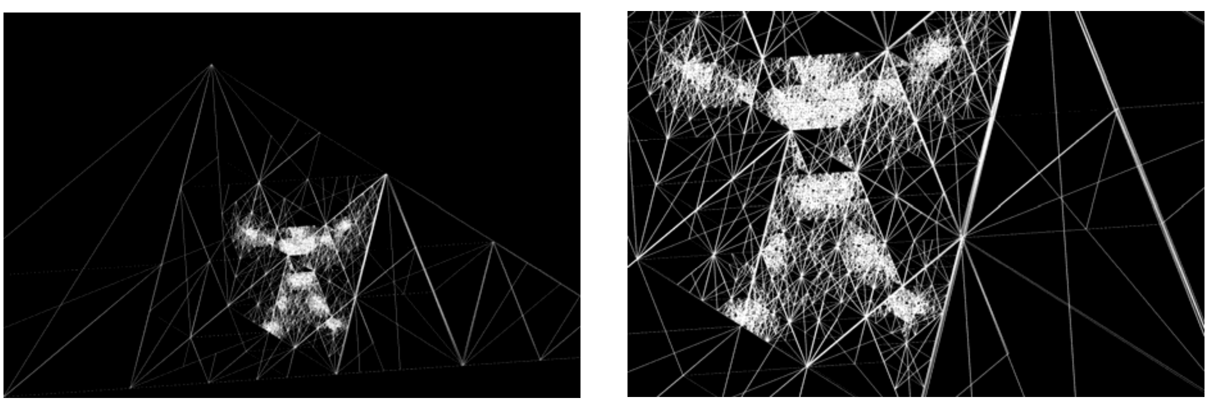


Figure 22. Tetrahedral tiling from the approximation of the nodal points of the human figure

The produced tiling is a result of the 13th iteration and gives a sufficient approximation of the human figure, through a reduction of 53% (table 8) to the number of tiles produced in comparison to considering the whole figure. The tiling appears to represent the internal organization of the figure (figure 22) and articulates the form through synergy of these areas of density.

7.3 Introducing planes in the centroid connectivity

As mentioned earlier the centroid connectivity of adjacent tetrahedral delivers a more diverse pattern. The restrictions of the tetrahedral tiling of the Danzer packing when considered for the production of inhabited space was the principle underlying the creation of the Voronoi pattern the Grotto project (Aranda, Lasch and Bosia, 2007).

As a next step for the experimentations on the affordances of the produced tiling we draw inspiration by J. F Gabriel's realisation that a crucial decision to be made is the orientation of the space frame structure in respect to the ground, and that a basic restriction is the need for horizontal planes in the space frame structure (Gabriel, 1997). We thus consider a method that checks the tiling produced by the centroid connectivity as for its ability to suggest its orientation if considered as a sketch of spaces to inhabit.

SIDE VIEW	FRONT VIEW PLAN VIEW	DATA	
	 	Number of iterations	4
		Number of points to fill the volumetric representation	250

Table 9. The data to which the orientation of the centroid tiling responds

The experiment aims to identify if there exists a certain plane that could be used as the horizontal plane for the orientation of the tiling. The growing pattern is iterated to the 4th level and then we proceed to an approximation of this reference plane for each of the four tilings, by considering the amount of produced tiles that are within a threshold of gradient to the referenced plane, as to be used as floors in the space frame.

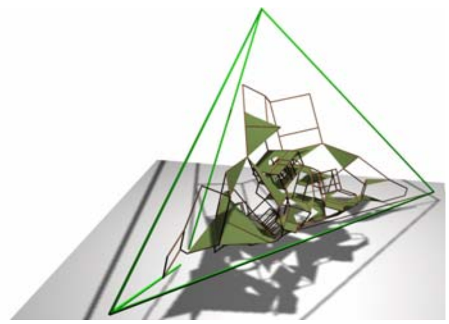
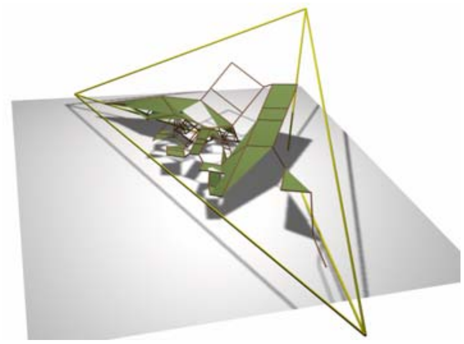
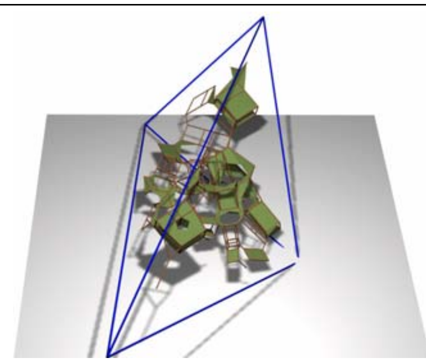
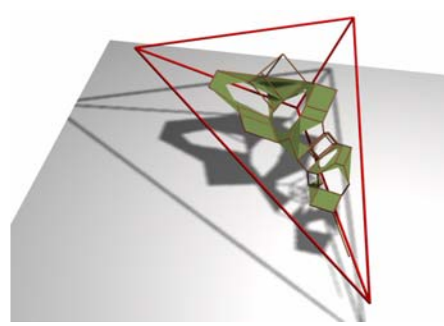
TYPE	**a.**)Total number of planes that fill the tiling	**b.**)Max. no of planes with gradient $< 30^{o}$	Ratio **b/a**	(Number of respective planes with gradient 80-90%)/**a**	Ref. plane rotation X axis	Ref. plane rotation Y axis	Ref. plane rotation Z axis	TILING WITH PLANES OF GRADIENT$<30^{o}$ & PLANE OF REFERENCE
A	5500	458	8%	65%	102	0	178	
B	2282	250	10%	63%	78	64	0	
C	5318	538	10%	59%	112	28	6	
K	1444	188	13%	63%	96	82	54	

Table 10. The data from the orientation of the centroid tiling

The method is carried out by a systematic approximation of the highest ratio of planes that are within a gradient of 30 degrees to the referenced plane. All planes within the threshold are drawn, and the ratio to the total number of planes of all orientations is printed on the screen. The approximation was carried out by rotation of the referenced plane. At the initial stage rotation around one axis was considered, and then the plane was oriented as a combination of the three resulting orientations of the initial step. At the next step we would change slightly one by one the angles of rotation produced, following small steps and lock the angle whenever we would get a better ratio. For each tiling a set of 15- 20 steps with an increase on the ratio of tiles within threshold was recorded.

The approximation was carried out manually, but a certain algorithm for the approximation of the optimal orientation of a potential horizontal plane could be implemented following the same logic of subsequent approximations, combination of the best results, and then browsing the space around these values.

The results of the experimentation are shown in table 10. The ratio of planes to be used as floors for planar rooms or event spaces is kept in the area of 8% for tile A to 13% for tile K. For the same rotations we also considered the planes that could be used as walls, by being approximately vertical to the reference plane, with a gradient of 80 to 90 degrees. The ratio of these planes to the total number was up to 65% for tile A. The combination of these two ratios suggests that the tiling could be used as a generator of emergent spatiality in the architectural scale, if appropriately oriented.

The tiling and the referenced plane are exported as a dxf file to a CAD program and are rendered after the reorientation of the tiling, by considering the referenced plane as the horizontal (table 10).

8. Estimation of the method and some thoughts for further development

The experimentation with the algorithm was a means to approach the morphology of the tiling deriving from the responsive growth of the Danzer packing and make some further thoughts on possible implementations of the method. The buffer images used were kept simple in order to draw comparisons within the instances, by altering a small set of constrains at each consequent step, and compare the results. Sideways the development of the method, the structures were interpreted and evaluated for the spatial relations they propose and how these could further be developed.
A set of realisations is made at this section, and is combined with a set of sketches through the use of the algorithm that explain these realisations and propose further investigation on the method.

8.1 Considering the sculptural result

Through the products of the approximation of the human figure, certain properties of the quasiperiodic structure became apparent and enhance the multiplicity of the emerging figure. The existence of 2-fold, 3-fold and 5 fold symmetry within the tiling (Kramer and Andrle, 2004), produces the realisation of a different spatial grid if seen from different angles. The patterns produced are appeared rational if seen from certain angles, since the cutting planes introduced by the initial iterations are maintained throughout the process, and provide a visual effect that resembles a fractal (picture 1, figure 23). It is manageable through these views to decode the principle logic behind the growing pattern and follow the subdivision process while focusing on the main figure. On the contrary from most angles,

Figure 23. Tetrahedral tiling from the approximation of the human figure

the body is conceived as the denser area of a triangular or orthogonal grid, (pictures 2 and 3, figure 23). The sequence of these aspects makes the result quite vivid since it combines a rational aspect, that allows perceiving the underlying logic, and a randomised aspect that facilitates its consideration as an entity flowing in the tiling.

The multiplicity of the visual outcome and the coexistence of these views bear relation to the cubists and the futurists, which in the beginning of the 20th century tried to capture motion in their paintings through the superimposition of simultaneous views of the figure. In a painting by Marcel Duchamp, *Nude Descending a Staircase,* the trajectory of movement of a human figure moving on a staircase introduced the factor of motion, in the painting. The painting is actually a result of superimposing the consequent captures of Eadweard Muybridge's photo with the same name.

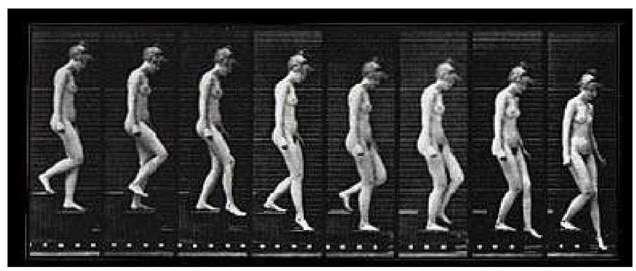



Figure 24. Painting: *Nude Descending a Staircase* by Duchamp in 1913, and Photo: *Nude Descending a Staircase* by Eadweard Muybridge in 1887 (www.snap-dragon.com/duchamp.html)

Such images inspired the experimentation on our computational model, in order to take this concept in the sculptural scale of space frame structures. The structuring of our program through the creation of a volumetric representation as a point cloud affords the movement of this point cloud as an entity throughout the running of the program. In the following experimentation, we initially consider the representation of a sphere as a point cloud (in red in figure 25), and subsequently move it throughout the tiling, and pause at selected areas, where the tiling is iterated producing denser areas. The resulting structure gives a result similar to the consideration of the nodal points of the buffer figure, as mentioned earlier, when the human figure was described as the sum of its joints and ends (section 7.2).

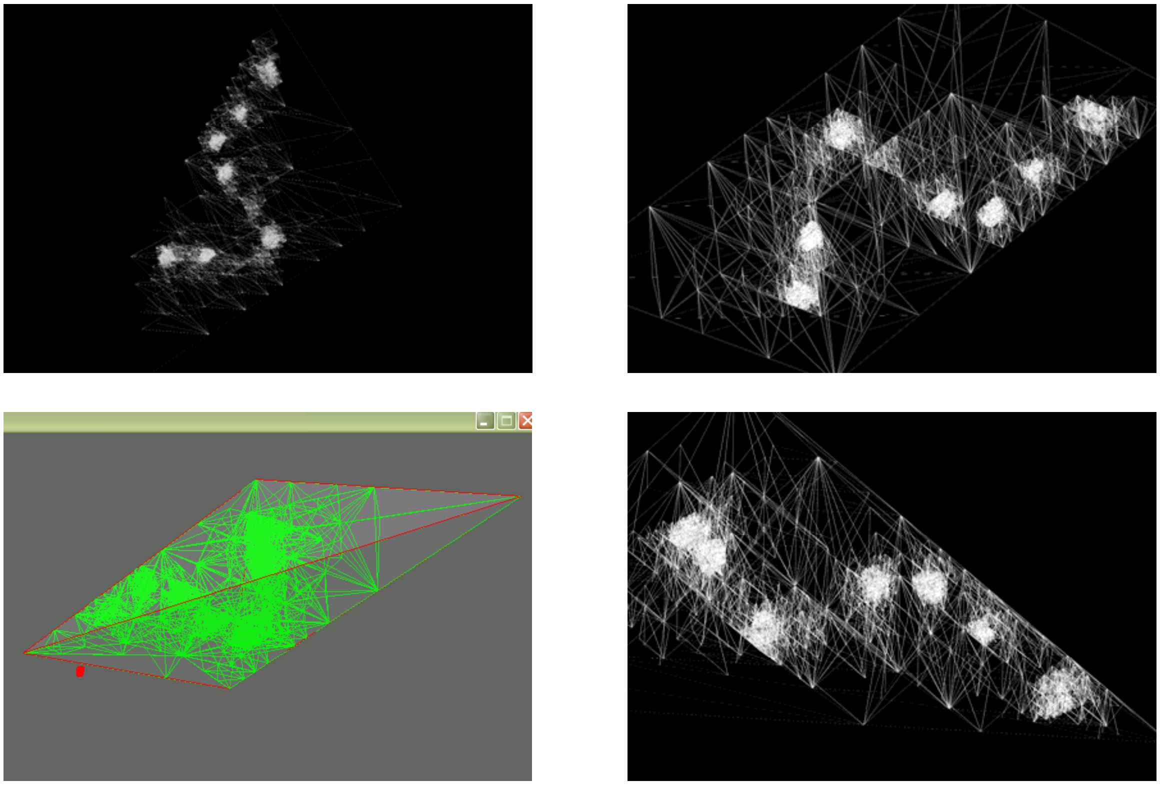


Figure 25. Tetrahedral tiling from the movement of a sphere throughout the tiling

The method is implemented by affecting the vector position of the point cloud, and manually moving the volumetric representation by affecting its x, y and z coordinates each time. Through this method the responsive system becomes a real time drawing tool, since the user can interfere in the process by moving the buffer figure and further subdivide certain areas according to his criteria.

The idea of movement can be carried further as to enhance the expressive qualities arising from the movement of a figure in time. The method can be extended by altering the buffer images throughout the run of the program, and so approximate subsequent alterings in the positioning of the figure, within the same space frame structure.

8.2 Considering possible implementations in the architectural scale

The second aspect approached is for the further use of the packing in architecture. In the Danzer packing the tiles considered are tetrahedrons that are not a very good host of habitable space. On the contrary the triangulations that the tetrahedrons produce are structurally sufficient and can produce self supportive entities.

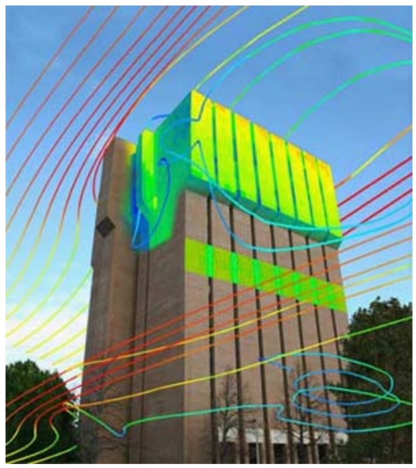


Figure 26. Architectural thermal analysis (Image from www.thermoanalytics.com)

The method could be used for the design of façade walls that would adapt to certain energy saving scenarios. A certain description of the environment to host the structure, through energy saving diagrams, or thermal analysis images that suggest areas of the building that should be shaded accordingly throughout the day, could be inserted as to guide the growth. Degraded images could also suggest the desirable orientation and placement of the openings, which could be regarded as the less dense areas of the structure.

We proceed with a certain basic scenario for the production of a part of the façade of a building, by just defining some openings on the wall. The method is checked as for its affordances towards the production of the surrounding shell of a building. The images used were to define areas of solid wall and openings, alongside the shape of the plan of the wall (table 11). The resulting tiling was handled as in the case of the Watercube project, as mentioned earlier, by manually introducing cutting planes that removed the surrounding tiles of a bigger scale.

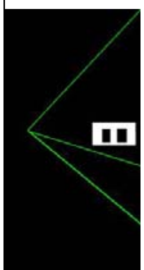
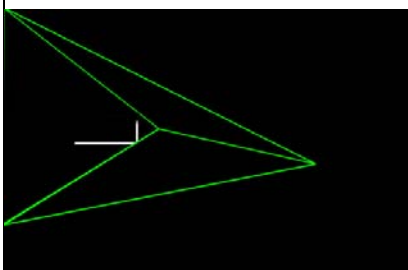
TYPE	SIDE VIEW	FRONT VIEW / PLAN VIEW	DATA	
A			Volume of buffer figure / Volume of the initial tile	1,1%
			Number of points to fill the volumetric representation	1050
			Number of iterations	10
			Number of tetrahedral in the responsive tiling	20437

Table 11. The data for the description of a façade wall

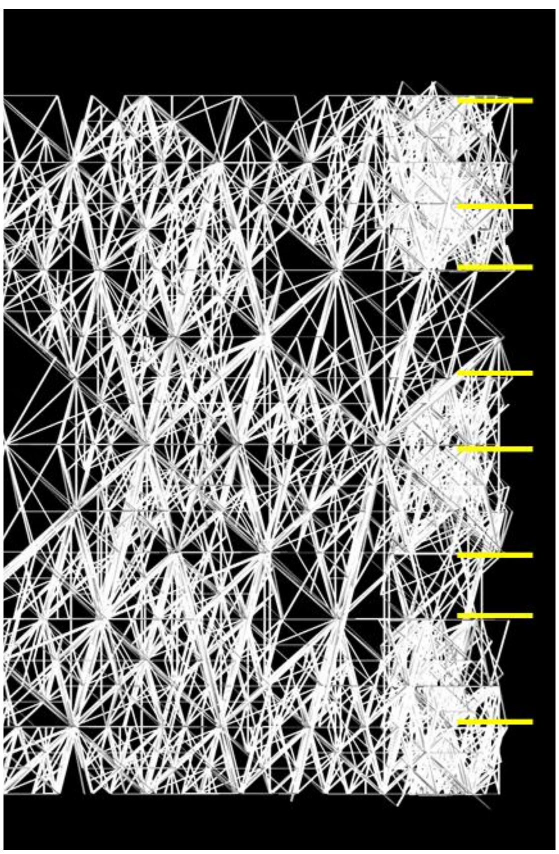


Figure 27. Tetrahedral tiling producing the façade wall

It is observed that the variety in scale, can reflect a comprehensive articulation of a façade. The method could be used for the study of shading and receive a feedback from the evaluation of the resulting shading patterns.

Alongside this responsive degradation of scales we can notice that certain planes that are produced by the tiling can support horizontal floors (yellow bars in figure 27). The appearance of these planes is also noted by Dr. Shea (Shea 2004), and we can observe that they maintain their consistency in the differential growth of the tiling.

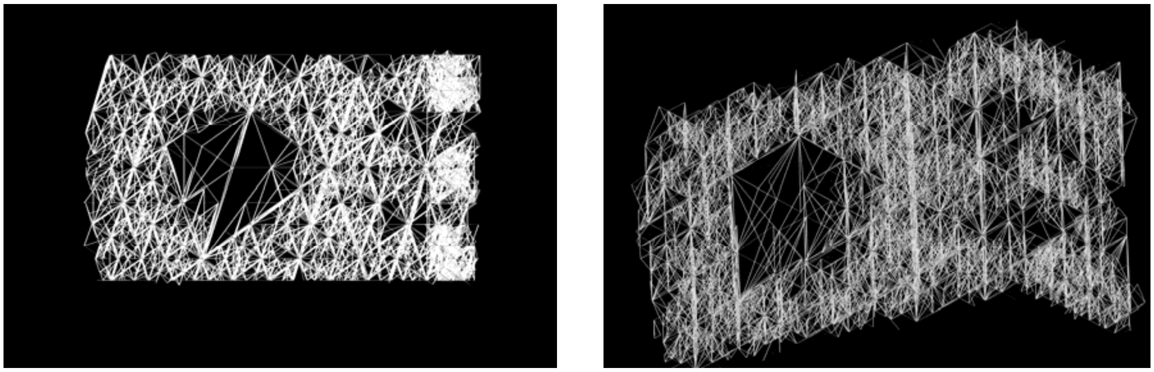


Figure 28. Views of the space frame façade wall

As a further step, a simple method of introducing cutting planes could be implemented on the algorithm in order to produce the planar tiling of the surface. Since our shape grammar is based on tetrahedral tiles as units, whenever such a cutting plane would bisect a set of the six edges of the tetrahedral, it could result in the definition of the cutting points and connect them. By connecting these points of each tetrahedral, one could obtain a planar tiling of triangles and quadrilaterals on the faces of each side of the façade wall.

At a different scenario the subdivision process could be a response to the structural optimisation of the structure. The computational model constructed could be combined with a program of finite element analysis, and define the areas that should become denser in order to obtain the structural efficiency of the tiling. Functionality is implied by the grammar but the external finite element analysis will provide performance evaluation through a simulated annealing algorithm (Cagan, 2001). Such a method could require that one keeps track of the tiling of the previous iteration and return to that if the result of the evaluation of the latter subdivision doesn't result in a better structural response, and so reconsider a previous subdivision.

8.3 Emergent spatiality

As it was observed the dual connectivity of adjacent cells produces an emerging space frame composed of triangles, squares, hexagons and octagons, combined in a variety of ways. These shapes produce patterns and result in the creation of units such as cubes or certain polyhedral that are articulated around certain axes of symmetry as seen in figure 29.

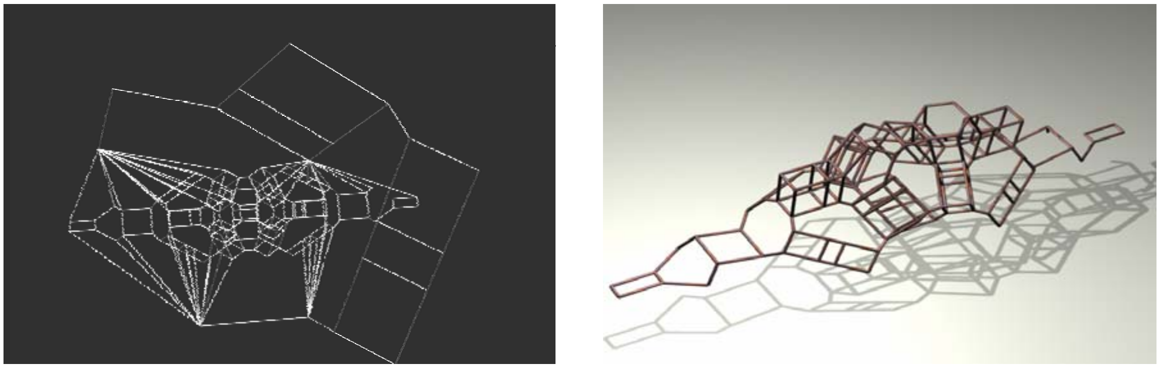


Figure 29. Views of a connectivity pattern through responsive growth to a moving sphere

A realisation made by considering these entities is that these axes of symmetry are actually one or more planes of symmetry. These planes are the ones produced in the first iteration of the initial tetrahedral, and they are combined as to produce planes that cut the initial tetrahedral throughout, and are maintained as iterations proceed. Thus any consideration of the tiling should consider the restrictions inherited from the geometry of the initial tile and structure a scenario to make use of these axes, or judging from the result of the iterations should go back and appropriately rearrange the position of the nodes of the initial tile, that would result in the reorientation of the resulting planes of symmetry.

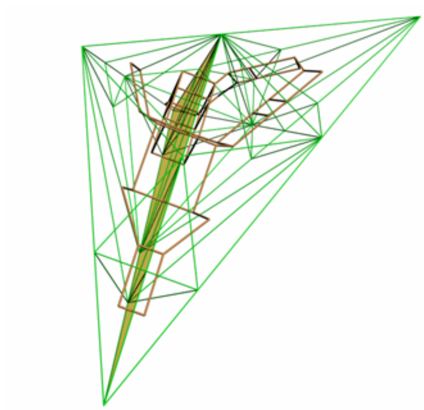


Figure 30. Appearance of a plane of symmetry (in yellow) for the centroid connectivity (in brown)

The existence of these axes gives a first element of articulation in the pattern produced, by introducing a connectivity diagram that connects to a set of secondary units adjacent to the corridor provided by the axis.

In parallel to considering the connectivity axes, we can also consider the placement of planes according to the constrains set in section 7.3. In that sense the resulting tiling can be oriented and filled with planes that could propose the positioning of floors, roofs and walls, alongside corridors that would connect these floors vertically and horizontally.

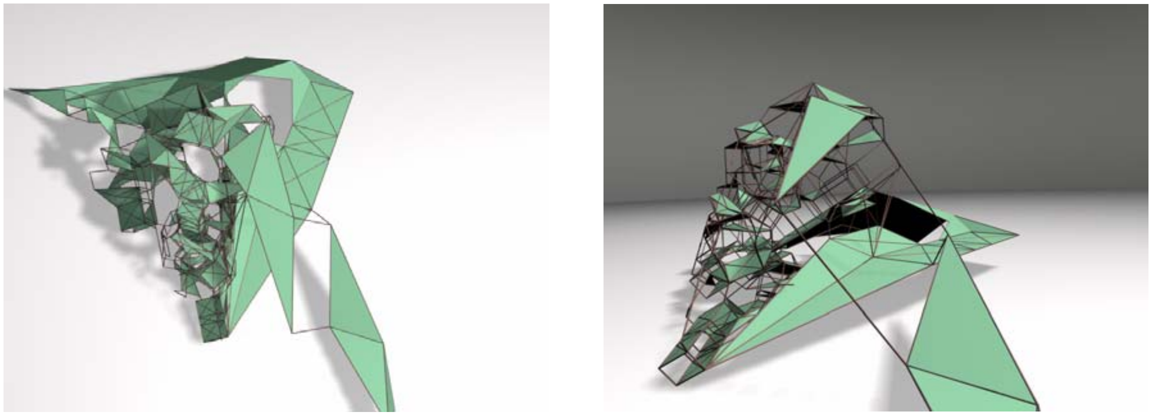


Figure 31. Centroid connectivity resulting from using the method as a real time design tool

At this stage we experiment with a tiling that is a response to consequent positions of the volumetric representation of a sphere, as mentioned earlier in section 8.1, using the method as a real time design tool. Planes are introduced with a threshold of 20 degrees to the horizontal (figure 31) and can be used as a sketch for further development in a CAD program. The introduction of planes also results in further triangulations within the structure that are considered to improve the structural behaviour of the structure.

Following this path of thought these self- supportive, emerging forms can represent the sketch for public buildings and be further extensively studied. The certain thresholds of gradients can result in the placement of floors according to the use of each cell, and can also help the acoustic behaviour in the case of an auditorium if considered as the roof of a room. Some parts of the generated form can also be interpreted as the landscaping of the plot to receive the structure, and so result in a unified design strategy that would be climaxed by the structure. The continuum of the interior and the exterior of the building is a well known strategy, which can be facilitated through this process of growing a structure.

9. Conclusion

Through the development of the method it was realised that an organic system of growth based on some simple and rational principles can respond in several ways to the same stimulating scenario. The diversity of an aperiodic formation, such as the Danzer packing, goes beyond the restrictions of its mathematic description, as soon as it manages to make use of its inherited affordances. The thesis first considers the inherited properties of the tiling and then reconsiders the way they could be could be used as to inform a meaningful scenario.

The approach taken, through the consideration of an ubiquitous factor such as the spatial properties managed to propose a set of possible implementations, each making use of particular instances of these properties. The method was at the same time an analytical tool and a form generator since the results taken from simple experimentations reinforced the algorithm through a feed forward loop. Each of the runs presented incorporates a certain problematic that underlies the former explorations on space frame structures mentioned, as to indicate how these could be answered through the use of the differentiated growth of an organic formation.

The description of the stimulating factor through a set of buffer images suggests the areas of density within the tiling but the generated form is always emergent as a combination of the inherited properties of the tiling and the adaptation to the buffer figure. At this stage the performance criteria can be included in the buffer images, as in the case of the thermal images, and each next iteration can be guided through the manipulation of the threshold values in the interface of the program according to the evaluation by the user. Through a different approach for further development, the computational model could communicate with other programs that would make decisions on the threshold values and the placement of the position of the triggering factor, by evaluation of its structural or environmental performance.

On the whole the thesis provides an open end method that affords further experimentation through the multiplicity it introduces, following the multiplicity of the aperiodic tiling it considers.

References

Aranda/Lasch and Bosia, D., 2006, Grotto, In I. Hwang et al., ed. *Verb Natures*, Actar, 2007

Boroczky K. Jr., Schnell U., and Willis M. J., 2000, Quasicrystals, Parametric Density and Wulff- Shape, in Baake M. and Moody R. V., ed., *Directions in Mathematical Quasicrystals*, American Mathematical Society, CRM Monograph Series, Vol 13, pp. 259- 276

Branco K., 2003, 'Digital Morphogenesis', in *Architecture in the Digital Age: Design & Manufacturing*, London:Taylor & Francis

Britannica Encyclopaedia, [online], [Accessed August 2007] Available from World Wide Web: http://www.britannica.com/

Cagan J., 2001, Engineering Shape Grammars, in Antonsson E. K, and Cagan J., ed. *Formal Engineering Design Synthesis*, Cambridge University Press, pp. 65-92

Danzer L., 1989, Three-dimensional analogs of the planar penrose tilings and quasicrystals, *Discrete Mathematics,* Vol. 76, pp. 1-7

Gabriel J. F., 1997, Are space frames habitable?. In J. F. Gabriel ed. *Beyond the Cube: The Architecture of Space Frames and Polyhedra,* John Wiley & Sons, 1997

Gormley A., 2007, *Blind Light* (exhibition guide), Southbank Center

Grünbaum B., and Shephard G. C., 1989, *Tilings and patterns*, Freeman New York

Hanna S. 2007, Body/Space/Frame, [online] [Accessed 15th July 2007], Available from World Wide Web: < http://www.sean.hanna.net/bodyspaceframe.htm >

Herron G., 1994, Re: Point within a tetrahedron, [online] [Accessed 7th July 2007], Available from World Wide Web: < http://steve.hollasch.net/cgindex/geometry/ptintet.html>

Kramer P., and Andrle M., 2004, Inflation and Wavelets of the icosahedral Danzer tiling, in Journal of Physics A: Mathematical and General, Volume 37, Issue 10, pp. 3443-3457

PTW Arcitects, ARUP and CSCEC, 2006, Watercube, In I. Hwang et al., ed. *Verb Natures*, Actar, 2007

Robbin T., 1997, Quaicrystal Architecture: The Space of Experience, In J. F. Gabriel ed. *Beyond the Cube: The Architecture of Space Frames and Polyhedra,* John Wiley & Sons

Shea K., 2004, Explorations In Using An Aperiodic Spatial Tiling as a Design Generator, First International Conference on Design Computing and Cognition 2004, 19-21 July, MIT, Cambridge, USA, pp. 137-156

Stiny, G: 1980, Introduction to shape and shape grammars, *Environment and Planning B*, Vol 7, pp.343-351

Thompson D'Arcy, 1961, *On Growth and Form (Abridged)*. Cambridge University Press, Cambridge

Weaire D. and Phelan R., 1994, A counter-example to Kelvin's conjecture on minimal surfaces, *Phil. Mag. Lett.*, Vol 69, no. 2, pp. 107–110

Weinstock M., 2006, Self Organisation and Material Constructions, *Architectural Design*, Vol 76, Issue 2, pp. 34-41

APPENDIX

APPENDIX I

The interface

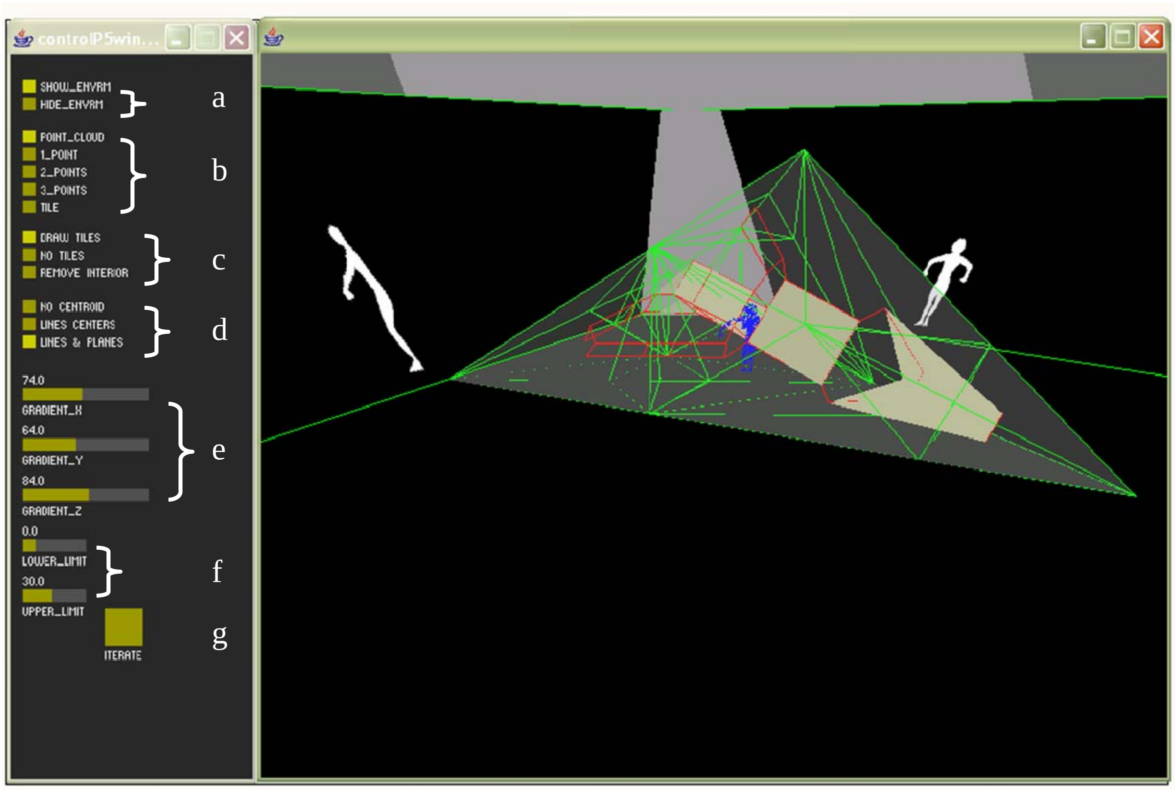


Figure 32. The Interface of the program, in Processing Beta 0118

The connectivity pattern of the Danzer packing is seen in green and is the result of the 3rd iteration of tile A. The tiling is surrounded by the buffer images in black and white (on the xy, xz and yz plane), which define the volumetric representation of the figure, seen in blue as a point cloud. Red lines represent the centroid connectivity of adjacent cells. We can also see the appearance of planes on the centroid connectivity within the threshold of gradient to the reference plane, seen in light grey between the connection of the three buffer images.

On the control panel it can be selected through each set of buttons and cursors as seen on figure 30:

a) whether to appear or remove the buffer images from the screen,

b) how to conduct the check in order to subdivide the tiles, check to the point cloud, or project one to four Vertices of each tetrahedron on the images,

c) whether to draw the full tetrahedral connectivity in green, or whether to create a hollow figure,

d) whether to create the centroid connectivity and create planes to fill it,

e) define the rotation of the reference plane

f) define the limits for the gradient of planes to be drawn in relation to the reference plane,

g) conduct the next iteration

Through the keyboard:

x) move the volumetric representation forward on x axis

y) move the volumetric representation forward on y axis

z) move the volumetric representation forward on z axis

u) move the volumetric representation backward on x axis

v) move the volumetric representation backward on y axis

w) move the volumetric representation backward on z axis

r) write and save a dxf file

APPENDIX II

In the main

//selection of the type made at the beginning of the run

//tile A
float [] init_tile=
{0.0,0.0,0.0,0.0,0.9510565,0.0,1.3763819,0.68819094,0.0,0.68819094,0.52573115,0.5, 0};

//tile B
float [] init_tile=
{0.0,0.0,0.0,0.0,0.9510565,0.0,1.3763819,0.26286548,0.0,0.42532536,0.2628655,0.3090171, 1};

//tile C
float [] init_tile=
{ 0.0,0.0,0.0, 0.0,0.58778524,0.0,1.3763818,0.85065114,0.0,0.688191,0.42532533,0.4999999, 2};

//tile K
float [] init_tile=
{0.0,0.0,0.0, 0.0,0.9510565, 0.0,0.52573115,0.2628655,0.0,0.34409553, 0.2628655, 0.24999997, 3};

Node root = new Node(init_tile);

void draw()
{ ....

//Vector containing all current tetrahedral

all_drawn= new Vector();

//Vector containing all centroid connectivity lines

all_lines= new Vector();

```
...
if (iterate==true)
{
nrOfiteRations++;
root.applyRule();
iterate=false;
}
...
all_drawn.clear();
all_lines.clear();
...
root.reset();
root.draw();
...}
```

In Class Node

```
class Node
{
  Vector children;
   Node(float[] s)
 {...
  children = new Vector();
  ...
  void reset()
 {
   COUNT =0;
     for (int i = 0; i < children.size(); i++)
   {
     ((Node) children.elementAt(i)).reset();
   }
 }

  void draw()
 {
   ...
   if (geometry != null)
   {
     if (children.size()<1)
     {
     all_drawn.addElement(this);
     geometry.draw();
     }
   }
   ...
   void applyRule()
 {  ...
```

// if the tile is any of these instances of type C (2,012,022,112,212,222, all ending with 2 that //stands for type C

```
if (geometry.type == 2 || geometry.type == 012 || geometry.type == 022 || geometry.type == 112 ||
geometry.type == 212 || geometry.type == 222 )
      {
        int r = 0;
        switch (r)
        {
        case 0:
          Node a = new Node(geometry.myshape.new_tile);
          a.geometry = new Geometry(210,geometry.myshape.new_tile);
          children.addElement(a);
          Node b = new Node(geometry.myshape.new_tile);
          b.geometry = new Geometry(212,geometry.myshape.new_tile);
          children.addElement(b);
          Node c = new Node(geometry.myshape.new_tile);
          c.geometry = new Geometry(222,geometry.myshape.new_tile);
          children.addElement(c);
          Node d = new Node(geometry.myshape.new_tile);
```

```
        d.geometry = new Geometry(213,geometry.myshape.new_tile);
        children.addElement(d);

        Node e = new Node(geometry.myshape.new_tile);
        e.geometry = new Geometry(223,geometry.myshape.new_tile);
        children.addElement(e);
        break;
      }
    }
    ...

    else {
    for (int i = 0; i < children.size(); i++)
    {

  if ( nrOfiteRations <1)
   {
      ((Node) children.elementAt(i)).applyRule();
  }
  ...
```

//for the next iterations checks in relation to the volumetric representation condacted here decide
//whether to apply rule or not

```
}
```

In Class Geometry

```
class Geometry
{
    Geometry(int t, float [] a)
  {
    parentspos=a;
    type = t;
  }


   void draw()
  {

  switch (type)
    { ...
```

//the different instances of tile C activate the appropriate draw function in class Tetra
//example (myshape.draw---(---,parentspos))

```
    case 2:// if type 0 then it doesnt execute, if 1 then it executes
      myshape=new Tetra(2,parentspos);/// parentspos= my_pos
      myshape.draw(2,parentspos);
      break;
    case 012:// if type 0 then it doesnt execute, if 1 then it executes
```

```
    myshape=new Tetra(012,parentspos);/// parentspos= my_pos
    myshape.draw_div_A(012,parentspos);
    break;
   case 022:// if type 0 then it doesnt execute, if 1 then it executes
    myshape=new Tetra(022,parentspos);/// parentspos= my_pos
    myshape.draw_div_A(022,parentspos);
    break;
   case 112:// if type 0 then it doesnt execute, if 1 then it executes
    myshape=new Tetra(112,parentspos);/// parentspos= my_pos
    myshape.draw_div_B(112,parentspos);
    break;
   case 212:// if type 0 then it doesnt execute, if 1 then it executes
    myshape=new Tetra(212,parentspos);/// parentspos= my_pos
    myshape.draw_div_C(212,parentspos);
    break;

    case 222:// if type 0 then it doesnt execute, if 1 then it executes
    myshape=new Tetra(222,parentspos);/// parentspos= my_pos
    myshape.draw_div_C(222,parentspos);
    break;
  ...
   }
  }
}
```

In class Tetra

```
class Tetra
{

  Tetra (int n, float[] aa)
  {
   parent = new float[29];
   n= type_tile;
  }

//in the beginning we just draw the initial tile

   void draw(int n, float[] aa)
  {
   new_tile = new float[29];
   new_tile = aa;
   draw_tile(new_tile);
  }

//whenever it is iterated though we subdivide and reorder

   void draw_div_C( int n, float[] aa)
  {
 //the initial vertices of type C
   x1=aa[0];
   y1=aa[1];
   z1= aa[2];
```

```
   x2=aa[3];
   y2=aa[4];
   z2=aa[5];
   x3=aa[6];
   y3=aa[7];
   z3=aa[8];

   x4=aa[9];
   y4=aa[10];
   z4=aa[11];

//the additional points on the edges delivered from the subdivision

   x5= aa[6]*(1-(1/(t)))+aa[9]*(1/(t));
   y5= aa[7]*(1-(1/(t)))+aa[10]*(1/(t));
   z5= aa[8]*(1-(1/(t)))+aa[11]*(1/(t));

   x6= aa[0]*(1-(1/(t*t)))+aa[6]*(1/(t*t));
   y6= aa[1]*(1-(1/(t*t)))+aa[7]*(1/(t*t));
   z6= aa[2]*(1-(1/(t*t)))+aa[8]*(1/(t*t));

   x7= aa[6]*(1-(1/(2*t)))+aa[0]*(1/(2*t));
   y7= aa[7]*(1-(1/(2*t)))+aa[1]*(1/(2*t));
   z7= aa[8]*(1-(1/(2*t)))+aa[2]*(1/(2*t));

   x8= aa[3]*(1-(1/(t)))+aa[6]*(1/(t));
   y8= aa[4]*(1-(1/(t)))+aa[7]*(1/(t));
   z8= aa[5]*(1-(1/(t)))+aa[8]*(1/(t));

//according to the type of produced tile A, in its instance 210- thus child of type 2=C, the 1st of
//its type 0= A
//we chose which vertices to use and rearrange for its creation

if (n==210)
   {
     //C1/first tile A............points on initial 1,2,4,6
     new_tile = new float[29];
     new_tile[0]=x1;
     new_tile[1]=y1;
     new_tile[2]=z1;
     new_tile[3]=x2;
     new_tile[4]=y2;
     new_tile[5]=z2;
     new_tile[6]=x4;
     new_tile[7]=y4;
     new_tile[8]=z4;
     new_tile[9]=x6;
     new_tile[10]=y6;
     new_tile[11]=z6;

  //face centers

  new_tile[12]= (new_tile[0]+ new_tile[3]+ new_tile[6])/3;
  ...
```

```
  //tile center

  new_tile[24]= (new_tile[12]+new_tile[15]+new_tile[18] +new_tile[21] )/4;
  ...

  // the type of the produced tetrahedron A
    new_tile[27]= 0;
    new_tile[28]=nrOfK;
    draw_tile(new_tile);
   }

   void draw_tile(float[] aa)
 {

  //and draw the tetrahedral produced

 }

}
```

APPENDIX III

In Class Node

```
void applyRule()
  {
…
if (nrOfiteRations >=2&& hologram == true )
{
for (int k=0; k< COUNTB; k++)
{

//the point to be checked

p1= new Vec(boids[k].m_pos.m_vec[0],boids[k].m_pos.m_vec[1] ,boids[k].m_pos.m_vec[2]);

//the vertices of the tetrahedron

t1= new Vec(((Node) children.elementAt(i)).geometry.myshape.new_tile[0], ((Node)
children.elementAt(i)).geometry.myshape.new_tile[1], ((Node)
children.elementAt(i)).geometry.myshape.new_tile[2]);

t2= new Vec(((Node) children.elementAt(i)).geometry.myshape.new_tile[3], ((Node)
children.elementAt(i)).geometry.myshape.new_tile[4], ((Node)
children.elementAt(i)).geometry.myshape.new_tile[5]);

t3= new Vec(((Node) children.elementAt(i)).geometry.myshape.new_tile[6], ((Node)
children.elementAt(i)).geometry.myshape.new_tile[7], ((Node)
children.elementAt(i)).geometry.myshape.new_tile[8]);

t4= new Vec(((Node) children.elementAt(i)).geometry.myshape.new_tile[9], ((Node)
children.elementAt(i)).geometry.myshape.new_tile[10], ((Node)
children.elementAt(i)).geometry.myshape.new_tile[11]);

//the Determinants

 float D0_p1= det_4x4(t1,1,t2,1,t3,1,t4,1);
 float D1_p1= det_4x4(p1,1,t2,1,t3,1,t4,1);
 float D2_p1= det_4x4(t1,1,p1,1,t3,1,t4,1);
 float D3_p1= det_4x4(t1,1,t2,1,p1,1,t4,1);
 float D4_p1= det_4x4(t1,1,t2,1,t3,1,p1,1);

//check if the point within the tetrahedron

 if ((D0_p1>=0 && D1_p1>=0 && D2_p1>=0 &&  D3_p1>=0 &&  D4_p1 >=0) || (D0_p1<=0 &&
D1_p1<=0 && D2_p1<=0 &&  D3_p1<=0 &&  D4_p1 <=0))
  {
    ((Node) children.elementAt(i)).applyRule();
  }

}
```

APPENDIX IV

In the main

```
 for (int q = 0; q < all_lines.size(); q++)
  {
   stroke(255,0,0);
   float [] a1= ((Line)all_lines.elementAt(q)).m_line;
   line (a1[0], a1[1], a1[2], a1[3], a1[4], a1[5]);
   if (nrOfiteRations>1 && CentrPlanes== true)
   {
     float thr1=gradient_X;
     float thr2=gradient_Y;
     float thr3=gradient_Z;
     float thr5=upper_limit;
     float thr6=lower_limit;
     float thr_upper=cos(radians(thr6));
     float thr_lower=cos(radians(thr5));
     Vec hor= new Vec(cos (radians(thr1)),cos (radians(thr2)), cos (radians(thr3)));
     hor.normalize();

     Vec dg1= new Vec(a1[0], a1[1], a1[2]);
     Vec dg2= new Vec(a1[3], a1[4], a1[5]);
     Vec ln12= sub(dg2,dg1);
     Vec ln21= sub(dg1,dg2);
     ln12.normalize();
     ln21.normalize();

     for (int r = 0; r < all_lines.size(); r++)
     {
       float [] a2= ((Line)all_lines.elementAt(r)).m_line;
       Vec dg3= new Vec(a2[0], a2[1], a2[2]);
       Vec dg4= new Vec(a2[3], a2[4], a2[5]);
       Vec ln34= sub(dg4,dg3);
       Vec ln43= sub(dg3,dg4);
       ln34.normalize();
       ln43.normalize();

//go through all lines for adjacencies

       if (q!=r && a1[0]== a2[0] && a1[1]== a2[1] && a1[2]== a2[2])
       {
         planes_all++;
         float a10= abs(dot(hor,ln12));
         float a20= abs(dot(hor,ln34));
         fill (248,255,198);
         Vec norm = cross( ln12,ln34);
         norm.normalize();
         float a30= abs(dot(hor,norm));

//check gradient of plane

         if ( a30>=thr_lower && a30<=thr_upper)
         {
```

```
        planes_drawn++;
        beginShape(TRIANGLES); //and draw a QUAD!
        vertex(a1[0],a1[1],a1[2]);
        vertex(a1[3],a1[4],a1[5]);
        vertex(a2[3],a2[4],a2[5]);
        endShape();
       }
      }
…
```

// print on screen the persentages of the planes drawn

```
println("planes_all");
float a=  100*planes_drawn/planes_all;
println( planes_all);

println("planes_drawn");
println( planes_drawn);
println( a);
println('%');
```