Cell-Based Venation Systems

Christoph Klemmt¹, Klaus Bollinger²
¹University of Applied Arts Vienna, Zaha Hadid Architects ²University of Applied Arts Vienna, B+G Ingenieure Bollinger und Grohmann GmbH
¹christoph@orproject.com ²office@bollinger-grohmann.de

Venation structures in leaves fulfil both circulatory as well as structural functions within the organism they belong to. A possible digital simulation algorithm for the growth of venation patterns based on the leaf surface has been described by the Department of Computer Science at the University of Calgary. Cell-based growth algorithms to generate surface meshes have been developed by biological and medical scientists as well as artists, in order to gain an understanding of developmental biology or to generate artistic form. This paper suggests the combination of the two algorithms in order to generate the morphologies of leaves and other structures while at the same time generating the corresponding venation system. The resulting algorithm develops large non-manifold mesh structures based on local rules of division of the individual cells. The venation system develops in parallel based on the flow of the plant hormone auxin from those cells towards the start point or petiole of the leaf. Different local behaviours of the cells towards their adjacent neighbours, towards their rules of division and towards the rules of developing veins have been investigated. The eventual aim of the algorithms is their application as tools to develop architectural and structural morphologies.

Keywords: Venation, Structure, Cell division, Developmental biology, Growth

INTRODUCTION

The analysis and abstraction of biological precedents for architectural and structural applications has found interests within the realm of a biomimetic design (El Ahmar 2011, Panchuk 2006). Organisms in the natural world have evolved over tens of thousands of years to be well adjusted to their environments. Nature is used as a model in design, but also as a measure to evaluate the performance of the design, and as a mentor from which we can extract as well as learn (Benyus 1997).

In architecture, biomimetic ideas are applied to a wide range of aspects, ranging from sustainability (Pawlyn 2011, Volstad and Boks 2012) to creating adaptive environments and material systems (Hensel 2006, Hensel and Menges 2007, Hensel et al. 2010, Weinstock 2010).

The aim of the paper is to define a cell-based algorithm which can be used to generate or grow morphologies for applications in architecture. A cell-
based growth algorithm is combined with a venation algorithm to enclose spaces while at the same time developing a possible support system for the structure. (figures 1, 2)

**VENATION SYSTEMS**

The vein networks found in leaves fulfil two functions: They form the circulatory system as well as the structural support of the leaf (Roth-Nebelsick et al 2001). Starting from the petiole, the network covers the surface area of the leaf and reaches the proximity of each cell. Angiospermae show the largest variety of ramifying patterns, which can express both dendritic (open) or reticulate (closed) topologies (Sack and Scoffoni 2013).

The development of the leaf and its venation system happens during two distinct phases: An initial phase of cell proliferation and a second phase of cell expansion. The higher order veins develop during the initial phase. Their development is influenced by sources of the plant hormone auxin in the leaf (Sack and Scoffoni 2013).

The lower order veins develop during the second stage of the leaf growth (Sack and Scoffoni 2013). Sachs formulated a canalisation hypothesis for the development of the veins (Sachs 1991), however stress in the leaf surface may be a driving factor which can explain especially the abundance of closed loops in reticulate venation patterns (Laguna et al 2008).

**Algorithmic Simulation of Venation Systems**

Different algorithmic models for the simulation of leaf venation have been developed (Prusinkiewicz and Runions 2012). Aristid Lindenmayer developed L-Systems in order to describe branching networks (Prusinkiewicz and Lindenmayer 1990). A canalisation model based on auxin flux was described by Rolland-Lagan and Prusinkiewicz (Rolland-Lagan and Prusinkiewicz 2005). A model for the simulation of lower order anastomoting veins based on stress mitigation between the epidermis and mesophyll was proposed by Laguna et al. (Laguna et al. 2008).

One algorithm to describe the growth of leaf venation has been described by the University of Calgary (Runions et al. 2005, Runions et al. 2007, Runions 2008). Variants of the algorithm have been used to create geometries within the fields of ar-
Architecture and design (METAfolly by ecoLogicStudio, Xylem and Hyphae series by Nervous System).

The algorithm is based on a seed point and a shape which gets filled with target points. In the case of a leaf venation simulation this shape will be the surface of the leaf (Runions et al. 2005).

In each iteration, the outline shape grows and gets filled with further target points using a dart-throwing algorithm. Starting from the seed points, the vein network grows a step per iteration towards the average direction of the target points. If some target points are in closer proximity to a vein node than others, this vein node buds and the network branches out (Runions et al. 2005).

DEVELOPMENTAL BIOLOGY

Developmental biology is the study of the growth of plants and animals at the level of cells. The main factors which are studied and which influence growth are cell proliferation, cell differentiation and morphogenesis (Wolpert et al. 2011).

Developmental Systems Biology

Algorithmic simulations are used as a methodology to understand the intercellular processes and behaviours. The development of multicellular systems are programmed to understand growth processes of plants and animals (Kaandorp et al. 2005, Kaandorp and Kübler 2001) or to understand human medical conditions (Shirinifard et al. 2009).

Due to the morphogenetic character of developmental systems biology, cell-based algorithms have also been used to generate forms for artistic purposes. George W. Hart generated geometries based on "bud" cells which divide and form new tissue (Hart 2009). Andy Lomas programmed systems of cell division based on the availability of nutrients (Lomas 2014).

In the iterative algorithms of Kaandorp, Hart and Lomas, individual cells are simulated as points in three-dimensional Cartesian space. The cells react to their neighbours, especially in order to maintain a certain distance between each other (Kaandorp et al. 2005, Kaandorp and Kübler 2001).

Specific rules control the behaviours of cell proliferation. Kaandorp uses diffusion limited aggregation to control the cell division (Kaandorp et al. 2005, Kaandorp and Kübler 2001, Hart 2009, Lomas 2014). Hart uses bud cells which divide (Hart 2009), while Lomas uses nutrient values in the cells to define the cell proliferation (Lomas 2014).

Further behaviours are used to control cell differentiation (Hart 2009) and nutrient distribution (Kaandorp et al. 2005, Kaandorp and Kübler 2001, Lomas 2014). The morphogenesis is further controlled by external factors such as the initial setup conditions or global geometric constraints (Kaandorp et al. 2005, Kaandorp and Kübler 2001, Hart 2009, Lomas 2014).

VENATION SYSTEMS BASED ON CELL DIVISION

Implementation

The algorithms for cell division and venation have been combined into a single simulation. The aim is to eventually integrate behaviours and constraints into the system so that it can be used to generate architectural and structural morphologies.

Cells in plant and animal tissue grow in volumetric accumulations (Wolpert et al. 1998). Organs with cavity spaces, which are filled with non-cellular material such as air or liquids, have cellular walls which are built up of multiple cells. Even thin layers such as the lamina of leaves have a thickness of several cells.
Morphologies which can be used in spatial design need to be able to enclose large volumes, the spaces which are to be occupied, by enveloping them with ideally thin structures and walls. In order to reduce the amount of cells necessary in the calculations, tissue thicknesses were defined as being one cell thick.

This is in line with the simulations of Hart and Lomas (Hart 2009, Lomas 2014), however it is significantly different from the approach of Shirinifard et al. (Shirinifard et al. 2009). This simplification greatly enhances the speed of calculation for the enclosure of a given volume, however many intercellular behaviours need to be adjusted from their biological precedent. Especially morphogenetic processes such as cell to cell adhesion or cell migration follow very different behavioural rules (Wolpert et al. 1998).

The algorithms have been developed on different platforms in parallel. Python, Processing (Java) as well as Softimage ICE have been used. The simulations are calculated in iterative time steps, with the cells as points in three-dimensional Cartesian space. (The Python implementation was calculated as four-dimensional points based on a Library by John Hull.) Simulations have been run with up to 50,000 vertices.

**Basic Cell Behaviour**

For the movement of the cells, both a velocity-only as well as a velocity-acceleration based model were implemented. The velocity-acceleration based model appeared to result in more coherent simulations.

Hart as well as Lomas are using a system in which each cell is part of a manifold enclosed mesh, similar to the arrangement within a spherical mesh. The neighbours of a cell in the mesh are defined from the beginning according to its insertion point (Hart 2009, Lomas 2014).

A venation simulation based on the Calgary algorithm however is able to compute any arrangement of target points, and in the original case of simulating leaf veins, the target points can be described as a flat open mesh (Runions et al 2005).

As the aim of the algorithm for this paper is to create architectural morphologies and to be able to enclose various types of volumes and spaces, the mesh arrangement of the cells does not need to be manifold nor completely open nor closed, but instead it needs to be able to enclose different volumes while continuing to grow. The mesh therefore needs to be able to grow back onto itself and cells need to be able to form new connections to already existing cells. The mesh in this case will cease to be manifold.

This demanded a dynamic cell neighbourhood: If available, the six closest neighbours of any cell are chosen as the direct neighbours, and this set of cells is updated whenever another cell comes into closer proximity.

A feedback between the intercellular forces and the adjustment of the direct neighbours is a likely reason for inconsistencies in simulations which are not using acceleration but velocity only. Due to a static direct neighbourhood, Andy Lomas does not get this feedback in his simulations.

**Intercellular Forces**

For the simulation, let a cell have position \( C \), and \( n \) direct neighbours with positions \( P_r \). The target distance (spring rest length) between two cells be \( d \). Let a point or vector have the coordinates

\[
\begin{bmatrix}
C_x \\
C_y \\
C_z \\
\end{bmatrix}, \quad P_r = \begin{bmatrix}
P_{rx} \\
P_{ry} \\
P_{rz} \\
\end{bmatrix}
\]

**Spring Force.** The cells attempt to maintain a constant distance towards their direct neighbours. This is calculated by adjusting the cell’s acceleration (or velocity in the velocity-only model) according to the distance to its direct neighbours by a spring system:

\[
\text{accelerationSpring} = \frac{1}{n} \sum_{r=1}^{n} \left( \frac{d - |P_r - C|}{|P_r - C|} \right),
\]

(Lomas describes the same behaviour in writing, however the formula in his paper appears to have a different result.)
**Planarisation.** In order to achieve a surface like arrangement of cells rather than a pile shaped accumulation, the cell position needs to be moved towards the mesh surface of the direct neighbours, similar to a mesh smoothing. Lomas uses a force towards the average location of the direct neighbours to achieve this, similar to a cohesion behaviour of the Boids algorithm (Lomas 2014, Reynolds 1987).

However in the system for this paper, the morphology has edge conditions, cells which only have neighbours towards one side. A cohesion behaviour would contract a system like this. Therefore the planarisation was achieved by a force which pulls the cell towards the plane through its three closest neighbours:

\[
\text{accelerationPlanar} = \frac{(C - P_1)\cdot((P_2 - P_1) \times (P_3 - P_1))}{|(P_2 - P_1) \times (P_3 - P_1)|^2} \quad (3)
\]

**Cohesion.** A cohesion force as in the Boids algorithm can be applied in order to achieve certain results, however its factor may need to be relatively smaller than that of the spring force which will keep the attracted cells apart. Cohesion was used only to a limited extend in the simulations due to its contracting behaviour (Reynolds 1987).

\[
\text{accelerationCohesion} = \frac{1}{n} \sum_{r=1}^{n} \frac{P_r - C}{|P_r - C|^2} \quad (4)
\]

**Separation.** A separation force as in the Boids algorithm pushes cells apart, and it can have a planarisation effect similar to the planarisation as described above if applied to the neighbours of direct neighbouring cells (Reynolds 1987).

\[
\text{accelerationSeparation} = \frac{1}{n} \sum_{r=1}^{n} \frac{C - P_r}{|C - P_r|^2} \quad (5)
\]

**Unary Force.** A unary force can be used to simulate gravity or other directional forces onto the cells. It needs to act either in combination with static cells or with obstacles such as a ground plane which can keep certain cells in a fixed position. A unary force will become useful for the generation of structurally acting geometries.

\[
\text{accelerationUnary} = \begin{pmatrix} U_x \\ U_y \\ U_z \end{pmatrix} \quad (6)
\]

with \(U_x, U_y, U_z\) being the forces in each Cartesian direction.

**Direction Dependent Factor (Drag).** A cell can be accelerated or decelerated in its movement within a certain direction. A deceleration in the \(z\) direction will result in the generation of more horizontal layers of the mesh. This factor is applied not as a force, but as an adjustment of the cell’s velocity. An even deceleration in all directions will result in a drag effect on the cell:

\[
\text{velocityAdjusted} = \begin{pmatrix} \text{velocity}_X \cdot a \\ \text{velocity}_Y \cdot b \\ \text{velocity}_Z \cdot c \end{pmatrix} \quad (7)
\]

with \(a, b, c\) being factors in each Cartesian direction.

**Cell Proliferation**

Cell division has been programmed to occur either by cell age or according to the amount of neighbours a cell has within a certain range. The cell proliferation by amount of neighbours can be used to generate marginal growth as described by Runions et al. (Runions et al. 2005). The cells along the edges have fewer neighbours and therefore divide.

Cell division by age can generate a uniform growth, but it can also be used to generate uniform isotropic (isogonic) growth, uniform anisotropic growth or non-uniform anisotropic growth as described by Runions et al. (Runions et al. 2005).

Of importance for this is the direction in which the cell splits into two, and the trigger of the division. If the split direction is random, a uniform growth will occur. A uniform anisotropic growth can be generated by aligning the split direction across the cells, and a non-uniform anisotropic growth can be generated by making the split direction dependent on, for
example, the local direction of auxin flow.

**Cell Differentiation**
The algorithms use different types of cells:

A petiole cell, which forms the start point of the simulation. Its position can be fixed and may be attached to a ground plane. The petiole cell is of importance for the development of the venation system as it forms the sink for the auxin flow. A system can have several petiole cells, and during a simulation new petiole cells can be created.

A general type of cell, similar to a meristem cell in plants. This cell can divide and turn into any other type of cell. For the simulations, a separation between meristem cells and mesophyll cells has not been made, all the cells can cause photosynthesis and produce auxin.

Vein cells are formed during the venation process and have different behaviours from the other cells. Vein cells are placed within the cell surface, rather than underneath as in many plants. Vein cells have a parent and children, which define their direction and auxin canalisation behaviour. The parent and children always remain in the group of direct neighbouring cells for the calculations.

Vein cells have been programmed to have a tendency to straighten between their parent and their children. This has been done by increasing the cohesion towards its parents and children while at the same time increasing the separation from the parent’s parent and the children’s children.

**Venation Growth**
The development of veins in leaves has been shown to be influenced by sources of the plant hormone auxin in the leaf (Sack and Scoffoni 2013). In the algorithm, every cell is being treated as an equal source of auxin.

The flow of the auxin towards the petiole cell is calculated, in two separate phases: The flow of auxin towards the closest vein cell, and the flow through the veins.

The flow towards the closest vein is calculated in reverse. Every cell in direct neighbourhood to a vein is updated and finds its neighbouring vein cell with the largest flow rate. Then iteratively always the next layer of adjacent cells is updated and finds its direct neighbour with the largest auxin flow rate. This neighbour's flow rate and that of its downstream cells is updated with the additional auxin from the newly incorporated cell.

In a second step, the cells adjacent to a vein are checked for their flow rates, and if this surpasses a certain value, i.e. a certain amount of upstream cells, this cell turns into a vein cell. It connects to its neighbouring vein cell which becomes its parent vein. As with the basic cells, the auxin flow rates, which define the diameter of the veins, is updated for all downstream vein cells.

The flow rate which defines if a cell becomes a vein cell fulfils a similar function to the "kill distance" in the Calgary venation algorithm, it controls the distance between adjacent veins (Runions et al. 2005).

**RESULTS**
The types of geometries which can be generated by the algorithm are surprisingly rich in their geometric behaviour and range from relatively flat, waving surfaces (figures 2, 3), to very intricate formations reminiscent of flowers or corals (figures 4, 5, 6, 7).

When using a direction dependent movement factor, horizontal planes can be created which can be used to simulate leaf surfaces. The venation patterns are most clearly visible in those structures. (figures 2, 3)
CONCLUSIONS AND FUTURE WORK

The geometries generated with the algorithm are very promising and show how a cell based simulation can be used to create complex morphologies. Very different types of structures can be generated by the system, which is controlled on a local cell level but as well can react to global constraints.

The geometries generated so far seem very suitable for the creation of art projects or experimental architecture. However they will need significant adjustments in order to become applicable for a functioning building design.

Various adjustments and additional behaviours should be incorporated into the algorithm:

- global constraints and influences such as site conditions to control and guide the growth
- growth of shell structures by reverse gravity acting on the cells
- venation beams which connect between petiole cells
- differentiation of vein cells of different orders as they appear in leaves
- stress between cells as a driver for venation, as proposed by Laguna et al. 2008

ACKNOWLEDGEMENTS

The paper was written by Christoph Klemmt as part of a doctorate under the supervision of Prof. Dr. Klaus Bollinger at the University of Applied Arts Vienna.

REFERENCES

Hart, G 2009 ‘Growth Forms,’ Proceedings of Bridges
Lindenmayer, A 1986, 'Mathematical models for cellular interactions in development I. Filaments with one-

Lomas, A 2014 'Cellular forms: an artistic exploration of morphogenesis', *ACM SIGGRAPH 2014 Studio*


Runions, A, Lane, B and Prusinkiewicz, P 2007 'Modeling trees with a space colonization algorithm', *Proceedings of the Eurographics Workshop on Natural Phenomena*, p. 63–70


Sack, L and Scoffoni, C 2013, 'Leaf venation: structure, function, development, evolution, ecology and applications in the past, present and future', *New Phytologist*, 198(4), pp. 983–1000


Volstad, NL and Boks, C 2012, 'On the use of Biomimicry as a Useful Tool for the Industrial Designer', *Sustainable Development*, 20(3), pp. 189–199
