DESIGNING WITH GRADIENTS

BIO-INSPIRED COMPUTATION FOR DIGITAL FABRICATION

Daniel Richards
Martyn Amos
Manchester Metropolitan University

1 Cross-section through the stalk of a "dead nettle" plant (Lamium maculatum) showing the complex organization of functional matter, which is difficult to model using traditional methods.

ABSTRACT

Digital fabrication technologies greatly enhance and extend manufacturing possibilities. However, we are still relatively limited in our ability to fully exploit these new methods and create complex architectural structures with *performance-driven* properties. We argue that entirely new computational approaches are needed, using scalable *generative* encodings and advanced bio-inspired *form finding* processes. This paper presents a novel generative model that can create functional and expressive geometries by evolving volumetric *gradient patterns*. Using three case studies, we demonstrate the key advantages of our approach. We demonstrate, using simulation followed by physical fabrication, that our approach is useful for exploring complex, yet buildable geometries in early stage design. Our new method is therefore suitable for performance-driven form finding tasks such as structural optimization, and holds vast potential for designing exotic multi-material and functionally graded materials in future applications.

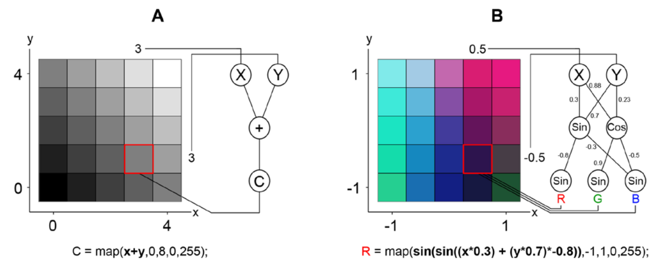
INTRODUCTION

Advances in fabrication technology make it possible to build increasingly complex designs. For example, additive manufacturing technologies (e.g. 3D printing) afford exciting opportunities to precisely control *how* and *where* material is distributed within geometrically complex objects (Oxman, Keating and Tsai 2011). However, the *computational* tools required to fully exploit these technologies and aid the discovery of geometrically diverse, yet high-performance structures, remain relatively underdeveloped (Lipson and Kurman 2013; Shea and Gourtovaia 2005).

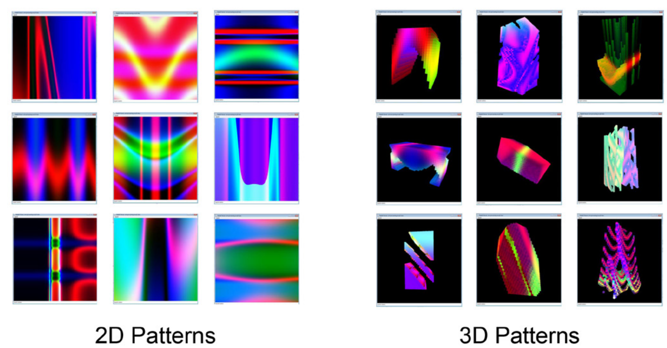
To address this challenge, emerging architectural research takes inspiration from form generation in natural systems. The main idea is to integrate simulation, analysis and physical testing into the early stages of design, in order to create performance-driven forms that can be realized with digital fabrication methods (Magna, et al. 2013; Menges 2012; Søndergaard, Amir and Knauss 2013). Notably, this approach does not aim to reduce complex design opportunities to one-dimensional optimization problems, but, in contrast, enables *new forms* of design instrumentality whereby designers work with materials, fabrication constraints and multiple performance considerations, early on in the design process, in order to *coax-out* desirable architectural characteristics in ways that were previously difficult to achieve.

We believe that this type of bio-inspired approach opens up radically new forms of architectural design and practice. However, we also believe that it is, in its current form, limited to creating relatively small-scale and simple designs. We suggest that to exploit the vast formal possibilities offered by advanced fabrication technologies, a key challenge is to develop new ways of *digitally representing* physical designs in ways that move beyond fixed geometric descriptions and/or linear associative models. Specifically, we think that the ability to control and explore complex material structures will be greatly improved with better *generative* design techniques, and that this demands experimental work at the intersection of architecture and a specific area of computer science that is concerned with generative and developmental systems (Stanley and Miikkulainen 2003).

For clarity, we use the term “generative” to explicitly refer to any design process that uses an *indirect* (non-linear) way of mapping between parameters and the final solution. For example, Nature uses truly extraordinary generative mappings to build complex forms. Consider that the human genome contains around thirty thousand genes, yet the human body contains about thirty *trillion* individual cells! (Bianconi, et al. 2013) In simple terms, this means that Nature uses indirect ways of describing form that

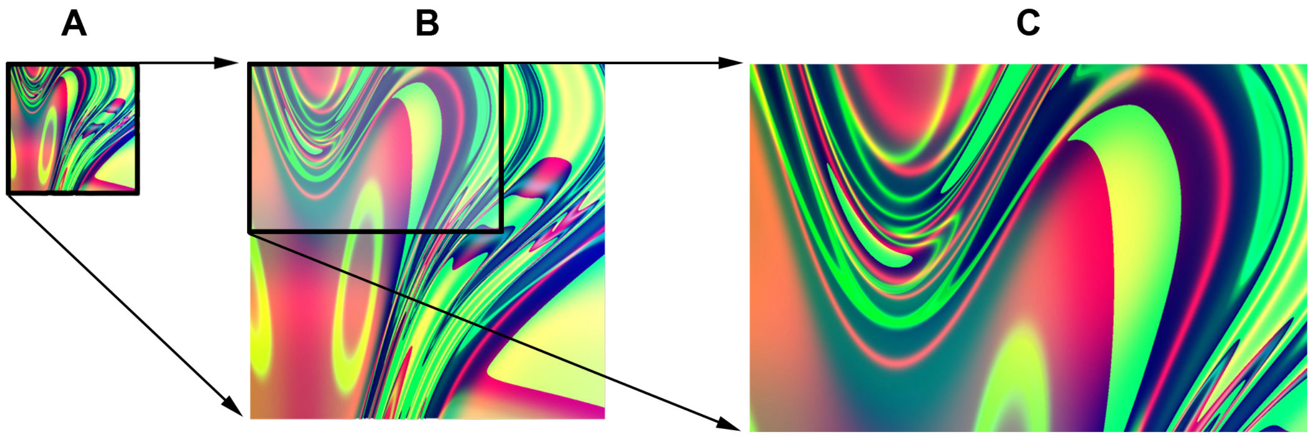


2 (A) Simple gradient pattern generated by summing the x and y coordinates of each pixel to generate a color: C. (B) CPPN generated pattern. To generate the color of the highlighted pixel, the coordinates: x = 0.5, y = -0.5, are fed into the CPPN and a red, green and blue (RGB) value is output. The equation below (B) shows the calculation of the red value.

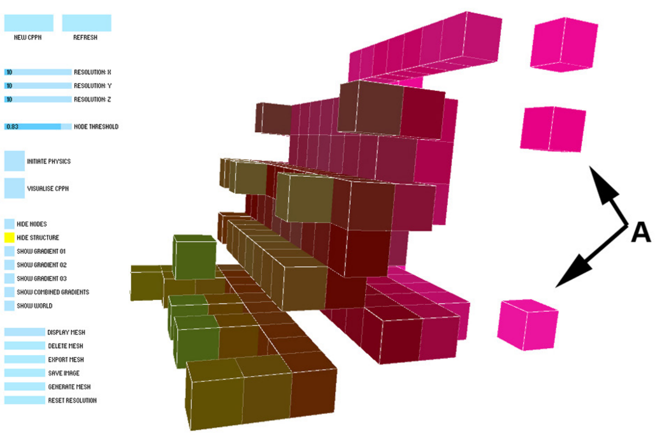


3 Diverse 2D (pixel) and 3D (voxel) patterns generated by random CPPNs. The 3D patterns are produced using a CPPN with four outputs: the three outputs shown in figure 2, for RGB values, and a fourth output which returns a threshold value. If the threshold value is less than zero, the voxel is not created. This process allows us to create different geometric shapes, as well as different colors.

reuse information, and in doing so creates truly awe-inspiring physical designs that are vastly more complex than the encoded instructions used to build them. Critically, it is this ability to *reuse* information during construction and describe complex physical designs with highly compressed encodings that we believe we can improve existing computational design methods. To justify this claim, consider using a traditional associative modeling approach (direct mapping between parameter and geometric transformation) to create a complex structure that comprises heterogeneous materials and exhibits specific performance-driven properties (such as Figure 1). This task is impossible using traditional methods, due to two major problems.



4 Infinite resolution of CPPN generated pictures. (A) Shows the original resolution of the image. (B) The resolution of the image is increased, and the pixels re-generate new colors based on the CPPN. (C) One corner of the image is isolated and increased in resolution. Notice that the image can never pixelate and has, in theory, infinite resolution.



5 CPPNs can generate unbuildable designs. (A) highlights voxels which are completely disconnected and unsuitable for fabrication.

Firstly, modeling this type of structure would be incredibly difficult because of the large number of parameters required to describe the geometry, topology and materiality of the design. Yet beyond the initial challenge of considering how and what to parameterize, and then laboriously modeling the structure, the resulting description would be extremely inflexible. Indeed, many of the *necessary* design decisions made to create a workable model will have constrained the geometry and topology of the design to a small number of possible permutations that may not enable suitable performance-oriented properties to be discovered (Hanna 2012). Additionally, making major retrospective changes to the model in order to accommodate new formal possibilities (following initial testing) will often be too difficult and require the designer to remodel the entire design (Davis 2013). Secondly, to create performance-oriented formal attributes, various simulation and optimization processes are needed. However, designs that contain large numbers of parameters are *not* easy to optimize and quickly become completely intractable. Our central argument is that to improve computational design processes and control significantly more complex designs (such as Figure 1), we need new types of active computational models that embrace notions of *self-organization* and material agency.

Previous work demonstrates the usefulness of generative design methods for creating complex geometries and performance-driven designs (Ayres 2012; Hornby 2004; Menges 2012). However, generative approaches can also present significant challenges when applied in practice. For example, multi-agent systems use impressively simple rule sets to create complex geometries, which exhibit emergent formal properties. However, the forms they create are generally not functional, and can even be completely unbuildable. Additionally, whilst the rule sets themselves are compact, and therefore potentially useful for scalable optimization, the algorithmic methods needed to *steer* these types of designs towards useful high-performance solutions are still severely lacking (Ayres 2012). Consequently, most studies have been limited to creating provocative ornamental geometries (Ramirez-Figeroa and Dade-Robertson 2013; Snooks 2012).

This paper presents a generative design process which facilitates speculative early-stage exploration of complex forms (yet also ensures buildable structures), and can aid the discovery of high-performance designs. First, we contextualize our approach and review related work. Second, we describe our model. Third, we present three experimental case studies that highlight the various advantages of our approach using 3D printed prototypes and finally, we conclude with a discussion that summarizes our findings, and highlights exciting possibilities for further research.

BACKGROUND

Michalatos and Payne (2013) recently demonstrated an interesting way of digitally *representing* geometry that uses volumetric gradient patterns to control a voxel-based model. This process allows them to vary the material properties of 3D printed structures, throughout their volume, by defining the property of each voxel by a function of its Cartesian *(x,y,z)* coordinates. The significance is that the method provides a viable way of controlling multi-material physical objects with complex internal architectures. This idea of digitally representing each part of an objects geometry as a mathematical *function* of Cartesian coordinates is also referred to as "functional representation" (Pasko, Sourin and Savchenko 1995), has been widely used in computer graphics, and was recently applied to the design of 3D printed microstructures (Pasko, et al. 2011). We suggest that *functional representations* could offer vast potential for architectural design by providing a compact method of describing complex material structures, while maintaining the ability to explore expressive formal designs. However, a major challenge for this area is how to manipulate these types of volumetric gradient patterns so that the physical designs they encode can develop specific mechanical properties. We believe that recent work in the field of *evolutionary computation* may offer a solution.

Evolutionary algorithms simulate the process of Darwinian evolution and have been widely used in design and engineering to generate physical structures that have specific properties (Frazer 1994; Holland 1975; Kumar and Bentley 2003). However, recent work on *neural evolution* and *evolutionary robotics* has demonstrated that gradient-based patterns (similar to Michalatos and Payne's model (2013)) can be evolved by combining a generative representation called *CPPN* (Stanley 2007) with a state-of-the-art neuroevolution technique called *NEAT* (Stanley and Miikkulainen 2002). CPPN stands for *Compositional Pattern Producing Network*, and NEAT refers to an evolutionary algorithm called *Neuroevolution of Augmented Topologies*. *CPPN-NEAT* has been described in detail(Stanley and Miikkulainen 2002; Stanley, D'Ambrosio and Gauci 2007; Stanley 2007), so here we provide a brief non-technical overview, and focus on describing the new design opportunities that this approach affords.

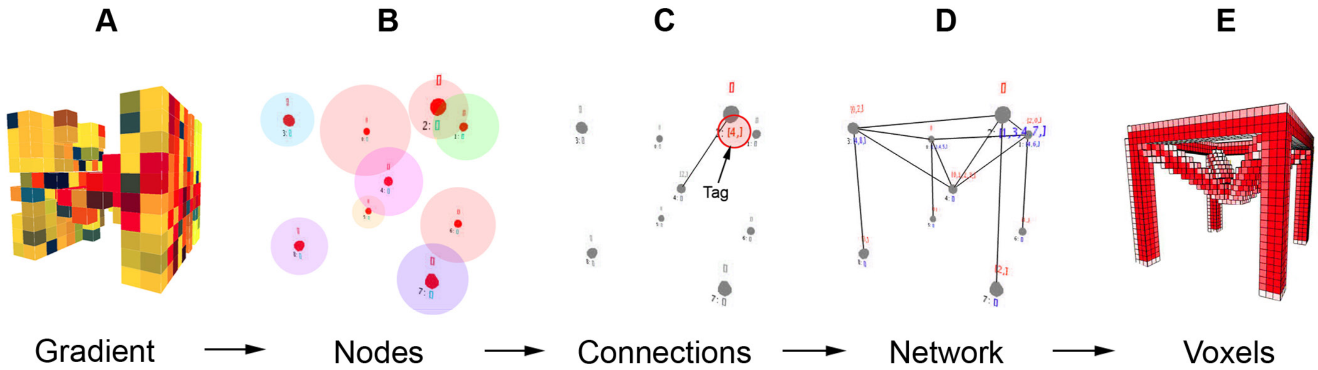
Simple 2D gradient patterns can be easily generated by defining the color of each pixel on a canvas as a function of its x and y coordinates. For example, a simple gradient pattern can be created by summing the *x* and *y* coordinates of each pixel and using the result to define that pixels RGB value, as shown in Figure 2A. More complex 2D patterns can be generated with a similar approach, but using slightly more complicated pattern producing networks, or *CPPNs*. These may use a variety of different mathematical functions, such as Sigmoid, Gaussian and Cosine, and are connected with weighted links (Figure 2B). Critically, Stanley and Miikkulainen's NEAT algorithm can *evolve CPPNs* using an evolutionary approach, whereby nodes and connections are added and manipulated over time (i.e. where networks are *augmented* by evolution). Combining *CPPNs* with NEAT, Clune et al (2011) use interactive evolution to create small 3D printed sculptures; Cherney et al (2013) evolve soft robot designs with multiple materials to achieve interesting locomotion; Hiller and Lipson (2009) use *CPPNs* to evolve 3D solutions that meet high-level goals, such as specific deflection of beam structures; and Auerbach and Bongard (2010) evolve virtual creatures with diverse behaviors.

We now outline our evolutionary model, which extends *CPPN-NEAT*, and allows us to explore complex and performance-driven material structures by manipulating volumetric gradient patterns.

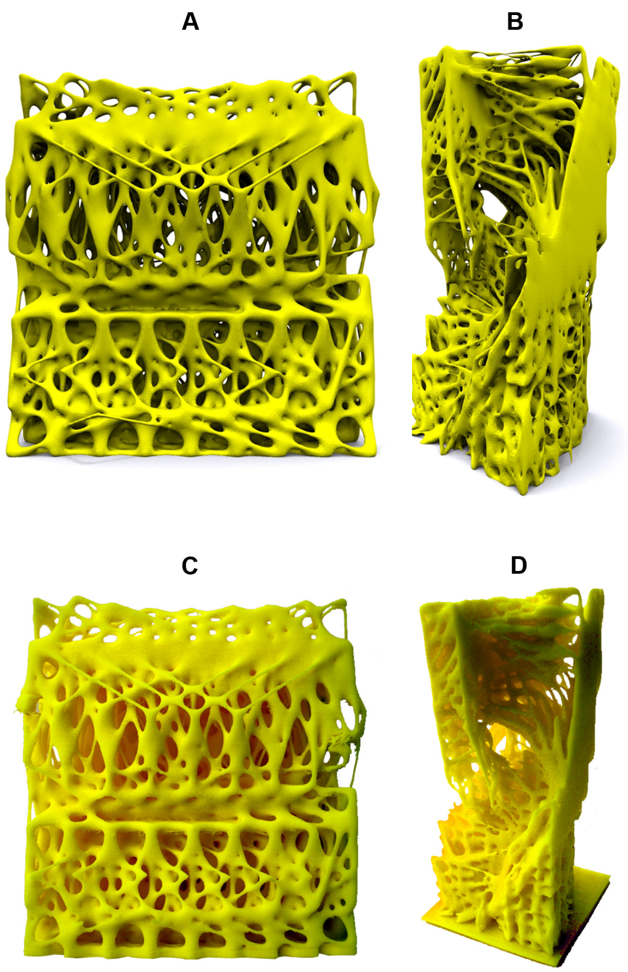
METHODS

CPPNs can generate diverse 2D and 3D patterns (Figure 3). These patterns have several exciting properties, which we believe to be useful for computational design. However, they also have one key problem that currently makes them difficult to apply to real-world design and engineering domains. This section will first outline the desirable properties of *CPPNs* and then describe how our extended method is able to exploit gradient patterns to generate useful (real-world) designs.

As shown in Figure 3, *CPPNs* can create regular geometric patterns with repeating motifs, which is a consequence of the periodic mathematical functions that they use (such as *cosine*). This means that designs encoded with *CPPNs* can exhibit symmetries, and perhaps, more interestingly, imperfect symmetries that will likely be useful for complex designs. A second interesting property of *CPPN encodings* is that the designs they encode effectively obtain infinite resolution. That is, as shown in Figure 4, 2D patterns generated with *CPPNs* can never pixelate, because as the resolution is increased each pixel simply re-queries the *CPPN* to generate a new pixel color. The implication of this is that structures described with *CPPNs* are extremely flexible and easy to manipulate during the early stages of design. Indeed, not only can the dimensions and resolution of designs be varied whilst retaining their evolved logics (in a similar manner to *NURBS surfaces*), but the connection weights of evolved *CPPNs* can actually be extracted and manipulated in real-time, providing similar interactivity as designs described by popular associative modeling platforms, such as



6 Growth process to generate buildable CPPN generated designs. (A) CPPN gradient, (B) nodes query the CPPN and obtain growth properties, (C) connections are grown between nodes. (D) Network paths are generated by growth, (E) Network paths are used as guides to paint through a volumetric space of voxels.



7 Complex geometries generated with model. Geometries exhibit repeating features, symmetries and imperfect symmetries, and have been fabricated with a desktop 3D printer. (A-B) Computer renders, (C-D) Actual 3D printed models.

McNeel's *Grasshopper*. Yet, unlike existing associative modeling platforms, *CPPNs* can be easily (and automatically) adapted with NEAT. Note this behavior follows for 3D gradients, and highlights the most significant feature of *CPPNs* for optimizing complex designs - *scalability*. Generative encodings such as *CPPNs* are useful for evolutionary design problems because they do not become more difficult to optimize as designs scale up (Hornby 2004). In simple terms, the *mathematical search space* (that contains all possible solutions) does not become larger and more difficult to search as designs increase in resolution and/or more elements are added. Critically, we believe that this ability to create smaller search spaces, which contain diverse solutions, will provide a significant advantage when aiming to design truly complex and performance-oriented structures.

As we have discussed, *CPPNs* have many desirable qualities for describing complex 3D designs. However, as with other generative design techniques, they also have problems ensuring *buildable* solutions. For example, when creating voxel designs with *CPPNs*, it is easy to generate designs that have disconnected (Figure 5) or non-manifold elements, which render them unbuildable. To exploit the beneficial properties of *CPPNs* and create functional 3D designs, our model incorporates an additional *growth* process, which uses local *agent-based* interactions to make *all* designs buildable.

We have previously described our growth process in detail (Richards and Amos 2014), so here we provide only a high-level description of the procedure (see Figure 6). The key insight for our model is that, instead of using *CPPNs* to describe the absolute position and properties of each voxel, we use CPPN generated gradient patterns to define connected paths for *painting* through voxel space.

As shown in Figure 6, we begin with a 3D gradient pattern, and instead of using outputs to describe properties of voxels (such as RGB values as Figure 6A), we use this information to seed the properties of 3D grid of nodes. Nodes have three different properties:

(1) a range, which describes which other nodes they can see and communicate with (Figure 6B);

(2) a brush size, which describes how thick paths between nodes will be; and

(3) a concentration variable that is useful during evolution.

Using this information, each node constructs connections with surrounding nodes that are within their range (Figure 6C) – note this is done using a "data-tagging" method which we have previously described (Richards et al, 2012). Following growth of connections, we are left with a network (Figure 6D) that is always suitably connected (no disconnected elements). This is achieved by ensuring minimum connectivity of nodes. Finally, we *paint* through a voxel space, with a specified resolution, and create a voxel model (Figure 6E). Again, node growth ensures a minimum "brush" thickness while painting through voxels, in order to avoid non-manifold elements.

The benefit of this additional agent-based growth process is twofold. Firstly, we can embed important *buildability constraints* and ensure that all designs generated are fully buildable and can be subjected to performance simulations without causing errors. The second advantage is that we can perform optimization and analysis procedures at various levels of abstraction. For example, during early stage design exploration it may be desirable to perform quick structural analysis at the level of connected bars (Figure 6D) and later fine-tune the design at the level of voxels (Figure 6E) or mesh, thereby creating significant *CPU* savings.

Our model is implemented with *Java* and *Processing*, and utilizes Karsten Schmidt's *toxiclibs* library to paint through voxel space and control meshing procedures. The following section presents the results of our initial case studies, and demonstrates that we can exploit the various formal benefits of *CPPN encodings* and create functional designs.

DESIGN EXPERIMENTS: GEOMETRIC COMPLEXITY

This case study demonstrates that desirable properties of the *CPPN encoding* persist when we introduce our growth process, and that this allows us to generate complex, yet buildable, geometries.

(Figure 7) shows two structures created with our *CPPN-based method*. To generate these designs we first situate a grid of nodes within a 3D physics simulation. We then grow network structures with *CPPN* instructions, and treat nodes as particles and connections as springs (using Verlet Integration). Finally, we create mesh-based structures by painting through voxel space (as (Figure 6E)) and applying an iso-surface to the solution. During this process, if nodes move (due to physics), new connections are constructed and old connections are destroyed as nodes move in and out of "range" of their neighboring nodes.

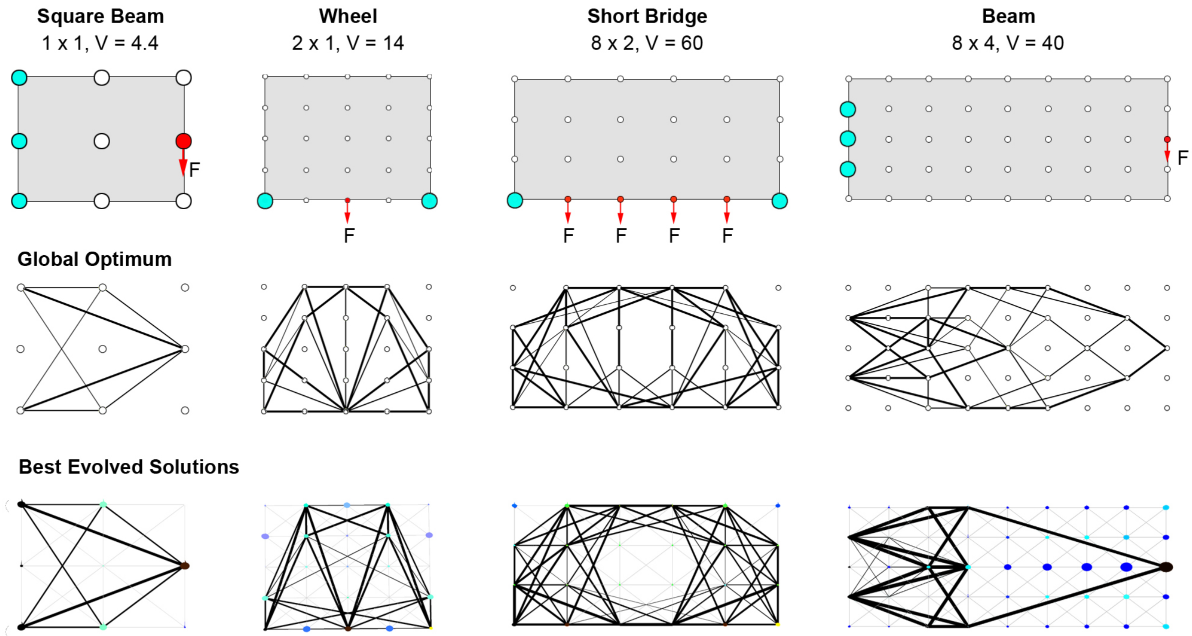
Using this model, we can explore a diverse range of forms in the early stages of design. Interestingly, the forms generated exhibit visually regular features with repeating motifs, as well as symmetries and even imperfect symmetries (Figure 7A) and (Figure 7B). Yet due to the physics-based growth process, the geometries are also *always* buildable (Figure 7C) and (Figure 7D).

EVOLVING FUNCTIONAL PERFORMANCE

Generative encodings often cause problems when creating functional designs. However, because our *CPPN-based model* is built on top of the powerful evolutionary algorithm: NEAT (Stanley and Miikkulainen 2002), we can easily use our method to discover functional morphologies. We now show this using some simple 2D and 3D topology optimization problems.

Topology optimization is widely used in structural engineering to increase the overall stiffness of designs whilst minimizing weight (Søndergaard, Amir and Knauss 2013). The goal is to create physical structures that resist deflection for a given loading, using a restricted amount of material. To validate that our model can discover functional, performance-driven designs, we evolved solutions to four well-known 2D truss optimization problems (Achtziger 2007). (Figure 8) illustrates the four benchmark problems, our best evolved solutions and the known global optimum. For all benchmark problems, the objective is to create a structural form that can hold the imposed loads, F, with minimum deflection, using a limited amount of material, V.

We evaluate structures using the linear direct stiffness method (Felippa 2013), which is a common finite element method (FEM). This process involves modeling the stiffness properties of each element in the structure and using this information to assemble a larger global stiffness matrix, which describes the mechanical properties of the entire structure. Using the global stiffness matrix, it is relatively straightforward to obtain mechanical properties of the designs, such as node displacements, and use this information to evolve



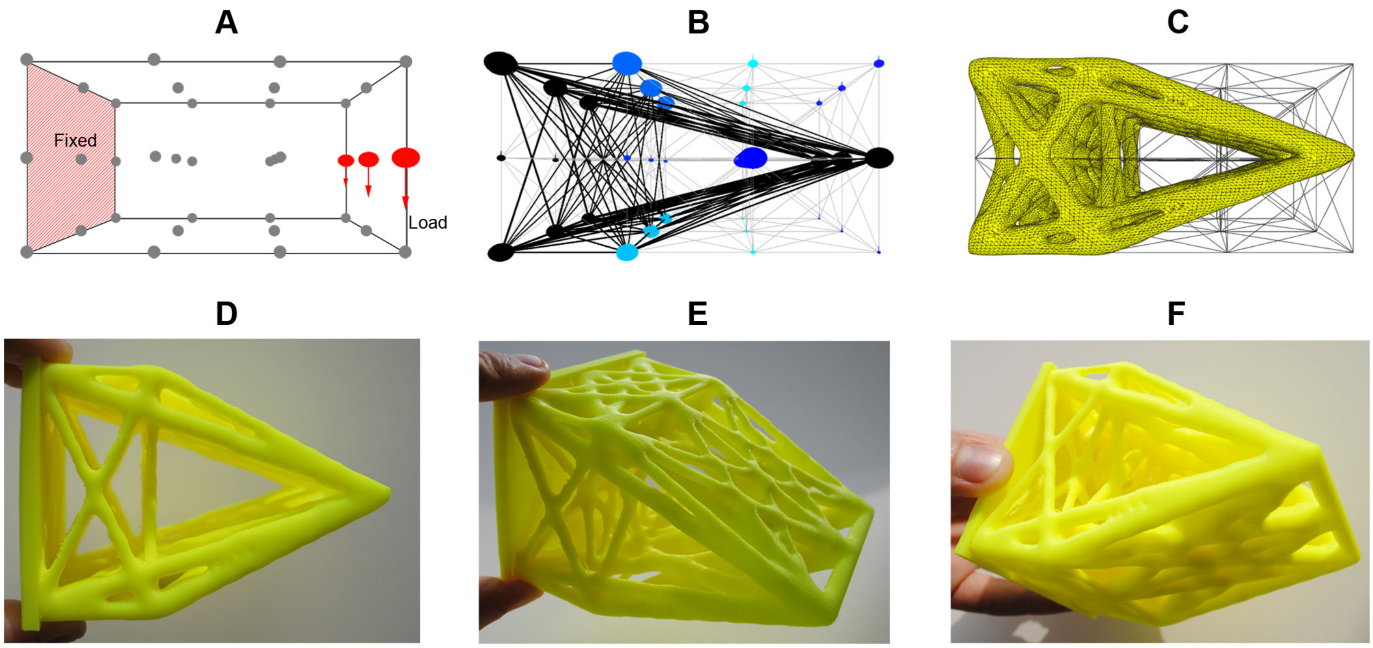
8 2D benchmark problems. (Top row) Description of the four problems. Blue nodes represent nodes with restricted degrees of freedom, whereas red nodes are subjected to imposed unit loads in the direction of the associated arrow. (Middle row) Best known solutions. (Bottom row) Best evolved solutions with our CPPN-NEAT model.

designs with desirable functional properties. As shown in (Figure 8), our model is able to generate 2D designs that have good structural performance, which share many commonalities with the known global optima (for further details see Richards and Amos (2014)). However, our approach also works for 3D problems and can be therefore be used to explore *physical* performance-driven designs (Figure 9).

TOWARDS MULTI-MATERIAL AND FUNCTIONALLY GRADED STRUCTURES

The two previous case studies illustrate that our *CPPN-based model* can successfully create expressive (yet buildable) geometries, and address simple performance-oriented optimization problems. However, we believe that this could also offer game changing possibilities to design complex multi-material composites and/or functionally graded structures with advanced additive manufacturing technologies. This final case study highlights what we believe to be an exciting trajectory for further work.

We evaluate our previous truss designs as connected bars (not voxel or mesh designs) with variable cross-sectional area and *homogeneous* material, in order to reduce the computational expense of simulation (Figure 9B). However, by adding an extra output to our *CPPN* encoding, we can also vary the *material properties* across voxel and mesh-based designs, and thereby optimize much more complex structures with heterogeneous material composition. (Figure 10) illustrates how we can use our *CPPN-based model* to define variable material properties across non-standard geometry. Additionally, because our approach utilizes a scalable evolutionary algorithm, we can feasibly search large mathematical spaces (of possible designs) for material structures that have *any* performance-driven properties, and are thus not limited by traditional homogenization techniques (Bendsoe and Sigmund 2003).



9 Example truss solution obtained with CPPN-NEAT. (A) Problem definition, (B) Evolved truss design, (C) Meshed truss design for 3-D printing, and (D-F) Photographs of fabricated example truss design.

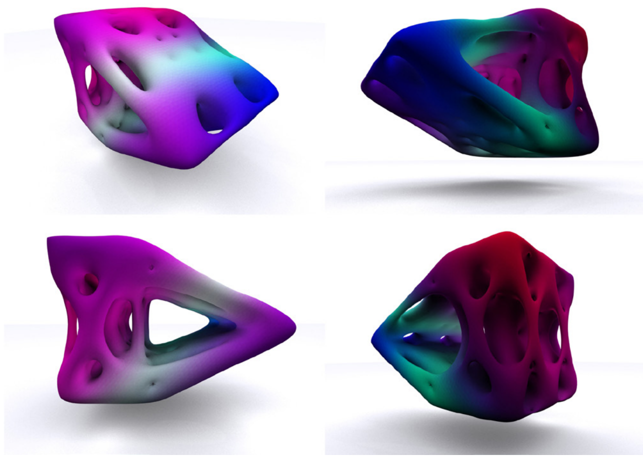
The idea of simultaneously defining geometry, topology and material composition of efficient large-scale physical structures (as suggested by Figure 10) represents a long-term goal of this research. However, our *CPPN-based model* may also provide more immediate benefit to design areas that apply shape optimization. For example, it is entirely possible to vary the surface thickness and/or materials across typical shell structures (as a function of the UV properties) in order to achieve specific mechanical performance, but this remains for further research.

Critically, we believe that future work in this area will facilitate new opportunities to explore complex heterogeneous material, and enable less wasteful designs by exploiting emerging fabrication advances (Oxman, Keating and Tsai 2011) using *CPPN-based methods*.

DISCUSSION

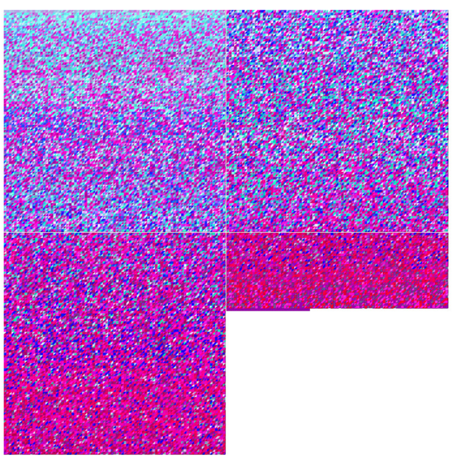
This paper demonstrates a novel method of working with volumetric gradient patterns to control performance-oriented material structures. We argue that this approach offers a valuable way of extending emerging architectural research (Menges 2012; Michalatos and Payne 2013) with a powerful generative encoding. Our case studies show that our *CPPN-based model* can create expressive, yet buildable, structures, which exhibit desirable geometric regularities and symmetries. Additionally, we show that we can discover performance-driven designs that fulfil 2D and 3D topology optimization problems. Finally, we highlight exciting new opportunities to describe and control complex heterogeneous material structures to exploit emerging additive manufacturing technologies.

Further work may explore three key areas. Firstly, the computational time needed to evolve voxel-based and mesh-based designs is currently significant. To address this limitation, we intend to exploit distributed or cloud-based computing, and significantly reduce the required processing time to facilitate experimentation with more complex performance-oriented designs such as compliant

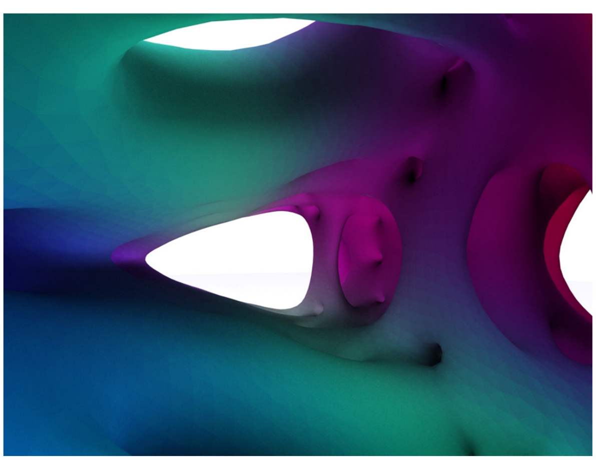


10 Illustrative truss design with varying (yet non-optimized) material properties.

A



B



11 On-going challenges with file formatting. (A) UV map of all mesh faces. (B) CPPN controlled geometry where colors illustrate varying material properties.

mechanisms. Notably, evolutionary algorithms are particularly well suited to parallel computing due to their structure, as are our matrix calculations for the FEM analysis. Secondly, to explore multi-material and/or functionally graded solutions, we will initially develop and/or integrate suitable file formats to control fabrication machinery (such as Hiller and Lipson (2009)). For example, our current approach requires us to generate cumbersome UV maps (Figure 11A) that specify various colors and/or material properties of 3D structures (Figure 11B), which is undesirable and computationally expensive. Finally, further work will scale-up of our initial studies to consider possibilities for large-scale architectural structures, as well as targeted application as part of larger integrated design processes.

REFERENCES

Achtziger, W. "Topology optimization of discrete structures: an introduction in view of computational and nonsmooth aspects." In Topology Optimization in Structural Mechanics, edited by G Rozvany, 57-100. Springer, 2007.

Auerbach, J, and J. C Bongard. "Evolving CPPNs to grow three-dimensional physical structures." Proceedings of the 12th annual conference on Genetic and evolutionary computation. ACM, 2010. 627-634.

Ayres, P. Persistent Modeling: Extending the role of architectural representation. Oxon: Routledge, 2012.

Bendsoe, M. P, and O Sigmund. Topological optimization: theory, methods and applications. Berlin: Springer-Verlag, 2003.

Bianconi, E, et al. "An estimation of the number of cells in the human body." Annals of human biology, 2013: 463-471.

Cheney, N, R MacCurdy, J Clune, and H Lipson. "Unshackling Evolution: Evolving Soft Robots with Multiple Materials and a Powerful Generative Encoding." Proceedings of the Genetic and Evolutionary Computation Conference. 2013. 167-174, ACM

Clune, J, and H Lipson. "Evolving 3d objects with a generative encoding inspired by developmental biology." ACM SIGEVOlution 5, no. 4 (2011): 2-12.

Davis, D. "Design Ecosystems: Customising the Architectural Design Environment with Software Plug-ins." Architectural Design 83, no. 2 (2013): 124-131.

Felippa, C A. Introduction to finite element methods. University of Colorado. 2013. http://www.colorado. edu/engineering/CAS/courses.d/IFEM.d (accessed March 28, 2014).

Frazer, J. An Evolutionary Architecture. London: Architectural Association , 1994.

Hanna, S. "Defining adequate models for adaptive architecture." In Persistant Modeling: extending the role of architectural representation, by P Ayres, 91-104. Oxon: Routledge, 2012.

Hiller, J, and H Lipson. "Design Automation for Multi-Material Printing." Solid Freeform Fabrication Symposium (SFF'09). Austin, TX, 2009.

—. “STL 2.0: A Proposal for a Universal Multi-Material Additive Manufacturing File Format.” Solid Freeform Fabrication Symposium (SFF’09). Austin, TX, 2009. 266-278.

Holland, J. Adaptation in Natural and Artificial Systems. Michigan: University of Michigan press, 1975.

Hornby, G. “Functional scalability through generative representations: The evolution of table designs.” Environment and Planning B 3 (2004): 569-587.

Kumar, S, and P, J Bentley. On growth, form and computers. Academic Press, 2003.

Lipson, H, and M Kurman. Fabricated: The New World of 3D printing. Indianapolis: John Wiley & Sons, 2013.

Magna, R L, et al. “From Nature to Fabrication: Biomimetic Design Principles for the Production of Complex Spatial Structures.” International Journal of Space Structures, 2013: 27-40.

Menges, A. “Higher Integration in Morphogentic Design.” Edited by A Menges. Architectural Design 82, no. 2 (2012).

Michalatos, P, and A O Payne. “Working with Multi-scale Material Distributions.” In Proceedings of ACADIA 13: Adaptive Architecture. Cambridge, 2013. 43-50.

Oxman, N, S Keating, and E Tsai. “Functionally graded rapid prototyping.” Innovative Developments in Virtual and Physical Prototyping: Proceedings of the 5th International Conference on Advanced Research in Virtual and Rapid Prototyping. 2011.

Pasko, Alexander, Adzhiev, V, A Sourin, and V Savchenko. “Function representation in geometric modeling: concepts, implementation and applications.” The Visual Computer, 1995: 429-446.

Pasko, Alexander, O Fryazinov, T Vilbrandt, P A Fayolle, and V Adzhiev. “Procedural function-based modeling of volumetric microstructures.” Graphical Models, 2011: 165-181.

Ramirez-Figeroa, C, and M Dade-Robertson. “Adaptive Morphologies: Towards a morphogenesis of material construction.” ACADIA 13: Adaptive Architecture. Cambridge, 2013. 51-60.

Richards, D, and Amos, M. “Evolving morphologies with CPPN-NEAT and a dynamic substrate.” the Fourteenth International Conference on the Synthesis and Simulation of Living Systems. New York. 2014. (To Appear).

Richards, D, and Amos, M. “An Evo-Devo Approach to Architectural Design.” Proceedings of the Genetic and Evolutionary Computation Conference. Philadelphia, 2012. 569-576.

Shea, K. Aish, R, and M Gourtovaia. “Towards integrated performance-driven generative design tools.” Automation in Construction 14, no. 2 (2005): 253-264.

Snooks, R. “Volatile Formation.” Log 25, 2012: 55-62.

Søndergaard, A, O Amir, and MI Knauss. “Topology optimization and digital assembly of advanced space-frame structures.” ACADIA 13. Cambridge, 2013. 367-378.

Stanley, K, D B D’Ambrosio, and J Gauci. “A hypercube-based encoding for evolving large-scale neural networks.” Artificial Life, 2007: 185-212.

Stanley, K. O. “Compositional pattern producing networks: A novel abstraction of development.” Genetic programming and evolvable machines 8, no. 2 (2007): 131-162.

Stanley, K. O, and R Miikkulainen. “A taxonomy for artificial embryogeny.” Artificial Life 9, no. 2 (2003): 93-130.

Stanley, K. O, and R Miikkulainen. “Evolving neural networks throgh augmenting topologies.” Evolutionary computation 10, no. 2 (2002): 99-127.

IMAGE CREDITS

Figure 1. Micropix (2012) Lamium Etzold Green 2. Wikimedia Commons / CC-BY-SA 3.0

Figure 2-11. Image credit to Authors (2014).

DANIEL RICHARDS is a research associate in the Biological and Sensory Computation Group at MMU. His interests span architectural design, bio-inspired engineering and computer science. He originally trained as an architect in Manchester before completing a PhD in 2013, which focused on the automatic design and optimisationzation of architectural structures using bio-inspired algorithms. His current postdoctoral research aims to develop scalable computer-aided-design tools that use generative encodings to leverage digital fabrication technologies and construct high-performance physical designs.

MARTYN AMOS is Professor of Novel Computation at MMU. His interests lie at the intersection of computer science, engineering and biology, and include synthetic biology, nature-inspired computing and artificial life. Projects he has directed have been funded by the UK EPSRC, the Wellcome Trust, and the European Union. He is the author of Genesis Machines: The New Science of Biocomputing.