From Intricate to Coarse and Back

A voxel-based workflow to approximate high-res geometries for digital environmental simulations

Vasiliki Panagiotidou[1], *Andreas Körner*[2]
[1,2] *University College London, The Bartlett School of Architecture, Bio-integrated Design*
[1]*vasiliki.panagiotidou.19@alumni.ucl.ac.uk* [2]*andreas.koerner.14@ucl.ac.uk*

Digital environmental simulations can present a computational bottleneck concerning the complexity of geometry. Therefore, a series of workarounds, ranging from cloud-based solutions to machine learning simulations as surrogate simulations are conventionally applied in practice. Concurrently, contemporary advances in procedural modelling in architecture result in design concepts with high polygon counts. This leads to an ever-increasing resolution discrepancy between design and analysis models. Responding to this problem, this research presents a step-by-step approximation workflow for handling and transferring high-resolution geometries between procedural modelling and environmental simulation software. The workflow is intended to allow designers to quickly assess a design's interaction with environmental parameters such as airflow and solar radiation and further articulate them. A controllable voxelization procedure is applied to approximate the original geometry and therefore reduce the resolution. Controllable in this context refers to the user's ability to locally adjust the voxel resolution to fit design needs. After export and simulation, 3d results are imported back into the design environment. The colour properties are re-mapped onto the original high-resolution geometry following a weighted proximity technique. The developed data transfer pipeline allows designers to integrate environmental analysis during initial design steps, which is essential for accessibility in the design profession. This can help to environmentally inform generative designs as well as to make simulation workflows more accessible when working with a wider range of geometries. In this, it reduces the perceived discrepancy between the concept and simulation model. This eases the use and allows a wider audience of users to develop co-creation processes between computation, architecture, and environment.

Keywords: *Simulation, Accessibility, Computation, Environmental Data, Workflow.*

INTRODUCTION

Digital environmental simulations can present a computational bottleneck concerning the complexity of geometry (Figure 1). With rising awareness of the critical state of our natural environment, the demand for architectural proposals to be more sustainable is increasing. Contemporary solutions to evaluate and consequently optimize designs are abundant. Solar, wind, and comfort analysis are now at the core of many architectural design projects. Concerning the conference theme of 'co-creation' the research is both interested in synergies between computational design (intricacy) and the environment (energies), as well as cross-platform design methods, linking various tools in the process.

Contrary to structural design, environmental design strategies often remain hidden and are less visible. Cutting-edge computational design tools can lend them specific aesthetics. Some researchers and practitioners observe an increasing desire to use the results of structural analysis or environmental simulations to articulate architectural surfaces. The results are sometimes highly intricate and have ornamental qualities (Schumacher, 2018). Ornamental because such articulations convey a meaning. In their case, the meaning is an increasing concern to depict the environmental performance of buildings (Peters, 2018). A process, that leads to a bottleneck concerning resolution.

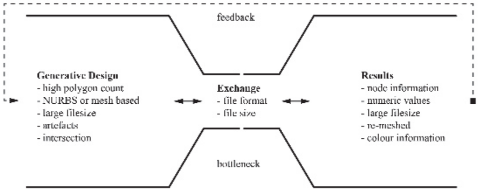


Figure 1
Digital environmental simulations can present a computational bottleneck concerning the complexity of geometry. This is especially relevant when it comes to the exchange of files from generative to simulation and analysis software design process.

CFD and resolution

The regular inability of Computational fluid dynamics (CFD) software to handle high-res geometry limits the ability to include environmental parameters when designing. Simulations allow designers to test their design options' future performance (Peters and Peters, 2018). In the field of CFD, the approach is often goal-driven, and optimization is the main driver. Occasionally, this contradicts their creative ambitions to use the analysis output for their architectural designs in an informed, but highly articulated, manner (Koerner, 2019). As a result, design and simulation are limited by a linear relationship of either deterministic or evaluative nature.

Software

Many programs allow for great formal expression, and sometimes excessive resolution (Carpo, 2016). This creates several issues concerning file size, management, and compatibility.

3d models created this way may run only inside the original software environment but must be reduced when transferred to analysis programs. As a response, some software have become increasingly closed environments. For example, the prioritization of integrated solutions like *Inventor* or *Revit* over standalone ones like *Vasari* in the case of *Autodesk*. Many analysis and simulation functions are now embedded within those programs and standalone tools, for example, *Autodesk CFD* (Autodesk, 2022), act as solvers. Parallel to this, many efforts are put into by an active developer community to create freely accessible solutions. One such example is *Ladybug Tools* (Ladybug Tools LLC, 2022) which is available for *Rhino Grasshopper* (McNeel R. & Associates, 2022).

Such tools often focus on user-friendliness to allow their interdisciplinary application within the built environment (Mackey and Roudsari, 2017). They also aim to allow designers and architects to capitalize on the power of digital simulation and analysis tools early in the design process. In practice, those tools come in at later stages of the design process and focus on the optimization of a single design option (Yang et al., 2017). Other approaches aim to create surrogate models which allow refining design options in real-time. This makes them more dynamic and applicable to designers (Wortmann and Schröpfer, 2019).

Digital environments

Today's modelling software can handle large amounts of point and polygon data. Architects use them to investigate highly intricate morphologies. Some experimental computational design methods emphasize the importance of real-time feedback and simulation (Klingemann, 2017). A quicker response time for such simulations and accessible applications can further increase design efficiency and save costs (Hosain and Bel Fdhila, 2015). Today, generative tools such as physics simulations, particle systems, generative adversarial networks, and volumetric methods, for example, OpenVDB (ASWF, 2021) are used as the base for further explorations. The OpenVDB library has been integrated into several popular digital design solutions such as SideFX's Houdini (SideFX, 2020), Blender (Blender, 2022), and Maxon's Cinema4D (Maxon, 2022), etc. They are widely used in academic research and education for concept designs.

Existing workarounds

A series of workarounds, ranging from cloud-based solutions to ML simulations as surrogate simulations are conventionally applied in practice. These solutions require special equipment and knowledge, limiting the use of data to specialized users and specific design options.

The following review lays out how other researchers have dealt with the beforementioned bottleneck conditions. Parallel to the straightforward increase of processing power via increased hardware performance, other software-based solutions exist. The presented overview illustrates the broad scope of state-of-the-art approaches. The examples are not like the one presented in this paper but demarcate the broader surrounding field of research.

CAAD-CFD solutions that aim to integrate digital simulations at the early stages of the design process are offered in several ways. They can be fully integrated Plug-ins, scripts created using integrated programming languages or node-based graphic programming languages, and standalone software with a bridge to exchange data between design and simulation interface (Chronis, 2017). Several methods can be applied to deal with the issue of geometry complexity. The most direct way is to increase the computational power of the calculating machine. Alternatively, the processing power of cloud computing can be harnessed by uploading the cases and downloading the result. The solving is outsourced, and the local machine is only used for the setup and the result visualization. Even further, entirely cloud-based solutions are available through a web browser interface, such as SimScale (SimScale GmbH, 2020). Recent research suggests the use of machine learning (ML) to speed up the process and give immediate design feedback within the design software environment (Zaghloul, 2017). Based on an estimation method, such surrogate models are proposed as an alternative to time-consuming CFD simulations during the early design stages (Mokhtar,2020).

High-res and low-res

Concurrently, contemporary advances in procedural modelling in architecture often result in design concepts with high polygon counts. It is a method that is increasing in popularity amongst computational designers in architecture (Ahlquist, 2016). In return, this leads to an ever-increasing resolution discrepancy between design and simulation models. Here, high-resolution stands for highly intricate geometries which are increasingly present in computational design. Responding to this problem, this research proposes a step-by-step approximation workflow for handling and transferring high-resolution geometries between procedural modelling and environmental simulation software. It intends to allow designers to quickly, but not simultaneously, assess a design's interaction with environmental parameters such as airflow and solar radiation and further articulate them.

Bottleneck

The above-listed methods tackle the bottleneck of geometry complexity by either increasing computational power or creating surrogate systems for simulations. This paper proposes an alternative method that is inspired by a process that is already part of many simulations. The meshing process of CFD setups is an approximation process. The user defines the fineness of the simulation mesh and its resolution. The number of boundary layers is defined to ensure a mesh's validity for fluid simulations. The method explained in the following chapters is based on an active, quasi-manual, use of this principle. A highly complex mesh of roughly 20k polygons is approximated using a voxelization system. This acts as a digital low-poly doppelganger to the original high-resolution geometry. The simulation results of radiation and external flow analysis are then re-mapped onto the original mesh using a script inside Houdini. The initial geometry is a generic procedural geometry without a specific scale or typology. It acts only for illustration purposes and has no specific functions beyond topologic diversity.

Aims

The goal of this research is to merge simulation and computation into a holistic design approach, by tackling the known issue of data transfer between generative and analysis software. The paper presents a method in principle. A rigorous numerical evaluation through comparing the time used simulating with and without the method is beyond the scope of this paper. To be conclusive, such an evaluation would require the testing of multiple geometries across a broad spectrum of hardware and software setups and a series of different simulation scenarios.

The transfer is based on step-by-step, non-linear mapping, resulting in a workflow able to integrate generation and analysis for iterative form-searching purposes. The use of such a method allows digital designers to integrate environmental responses into high-res geometry creation easier. The greatest potential lies in the application within generative design processes that create intricate geometries with high polygon counts and detailed surface articulations.

METHOD

In the following paragraphs, the outlined unified data transfer workflow comprised of analytical and generative software is presented. Geometry was generated in Houdini and simulated with two programs integrated into the method. These programs were (a) an educational license of Autodesk CFD and (b) Ladybug for Grasshopper. The topological data exchange between Houdini and the analysis software was achieved through a proposed voxel approximation method. This method of voxelization regards the analysis of the original geometry into three-dimensional cubical units, here reported as voxels, to approximate the topology into orthogonal surfaces.

This method can be repeated during the design process, according to the user's needs and goals and can be roughly divided into three steps. The first step is voxelization.

High-res to low-res

For this research, a test geometry was generated inside Houdini, by a particle system and an advection algorithm was applied afterwards. A Perlin noise vector field and a gravity force were implemented to inform each particle's position update, and all the updated positions of the system elements were traced per frame. The resulting polyline traces were converted into a volume through a volumetric data structure conversion method. Once the form was fixed, the volume was converted into a polygon geometry (Museth, 2013). Following that, the result was approximated by a set of voxels arranged in three-dimensional space. During this stage, unification and simplification levels are customizable. The choice of distributed voxel resolution, ranging from high to low, depends on the machine's computability, the target analysis software's 3d handling capacities and the user's goal. The output of this step was a rectangular cage, which approximates the original geometry and was saved in a *.STL format. Consequently, the final cage was automatically encoded as a mesh geometry to be imported into the environmental analysis software. The mesh conversion inside the generative environment, allows users to have control over the total number of generated mesh faces.

Simulations and Analysis

For the solar heat gain analysis, the *.STL file was imported into Autodesk CFD, where a mesh wrap was automatically generated around the imported geometry. The simulations were executed on a laptop with a 2.6GHz Intel® Core™ i7-6700 CPU and 16GB RAM. Once the simulation was completed, the generated RGB colour values were saved as texture maps in *.FBX format.

For the solar radiation analysis integrated into the herein presented platform, the *.STL file was imported into the Rhino6 environment, and the mesh was assigned inside Grasshopper. It should be noted, that upon simulation completion, the analyzed mesh changed regarding vertex count inside the Ladybug solver. Specifically, the input mesh was comprised of 2848 Vertices and 3332 Faces whereas the output mesh was comprised of 9996 Vertices and 3332 Faces. An RGB colour value was generated for each face and stored as a Houdini point attribute using the *HouGH* (Miroljub, 2022) plug-in.

HouGH is an interface that enables interoperability between Grasshopper and Houdini, as it encodes geometries and their embedded information within the *.JSON file format. At this point, it should be highlighted that mapping is not only part of the last step of the proposed methodology but is already an integral part of the discussed tools. Specifically, the RGB values generated from the analysis tools are colour maps from a topological point of view, as RGB values are automatically encoded into the mesh faces. These faces values are then re-mapped into the mesh vertices of the original geometry, either automatically or, in this case, manually after data import. Whether the data re-mapping process is accomplished automatically or manually, depends on the destination generative software.

Low-res to high-res

After export and simulation, this approximated geometry is transferred back into the design environment. The colour properties are re-mapped onto the original high-resolution geometry following a weighted proximity technique.

For this last step, both simulation data files were imported to Houdini, to re-map the RGB values onto the original complex geometry through an attribute transfer method, based on a weighted proximity technique. Each point's imported data, encoded as an RGB value, constitutes its attribute and is transferred from each point of the simulated geometry to the respective closest point of the original geometry. Houdini allows for the transfer of any attribute from source to destination, disregarding topological differences.

RESULTS

The final output of the proposed workflow is a set of voxels with embedded attributes/data resulting from the interaction between voxels and context. The workflow suggests a novel way of analyzing intricate geometries, by dealing with the issue of environmental analysis of complex topologies through the proposed voxel approximation method (Figures 2 and 3).

As a method, it makes otherwise tedious three-dimensional analysis possible both at early design stages as well as through widely used design tools. For example, the here presented high-res mesh might need over seventy minutes to load into a CFD program on the used system. The voxelated version, on the other hand, requires only ten to twenty minutes to load, mesh and simulate. In addition, the method allows controlling the geometry resolution inside the design software, rather than through meshing in the simulation environment.

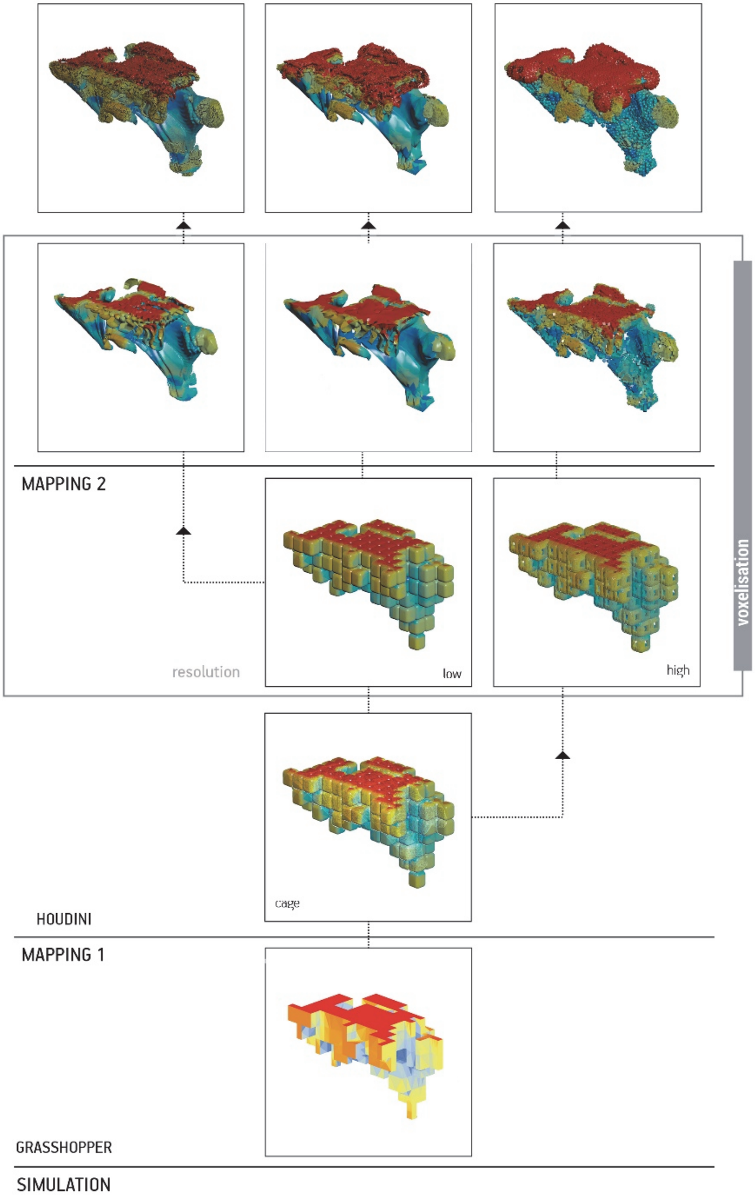
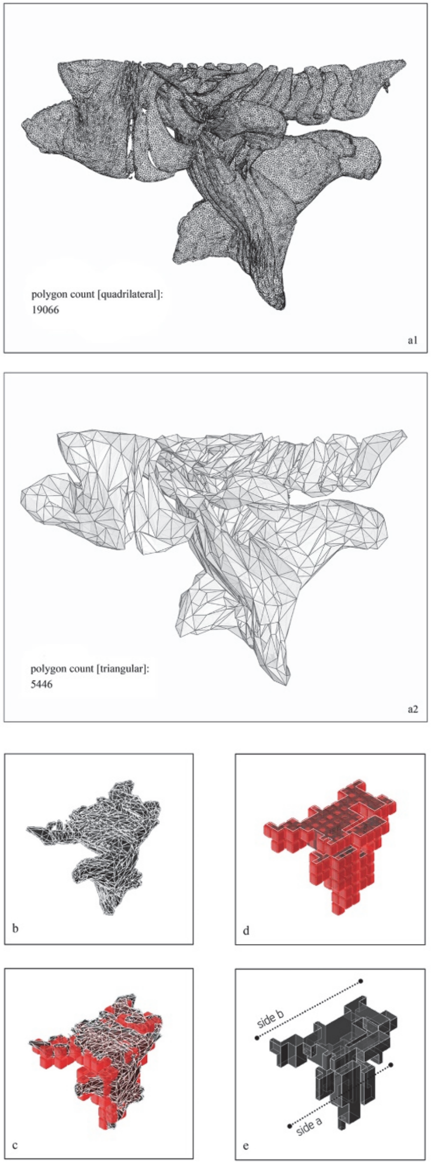


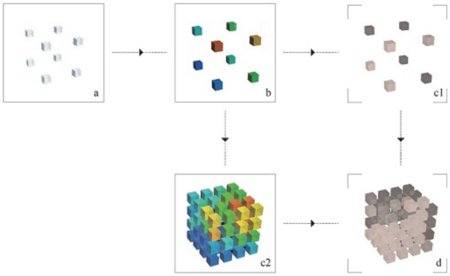
Figure 2
The voxel-based workflow to approximate high-res geometries for digital environmental simulations operates as follows (bottom to top).

Simulation: The initial, fine, high-res geometry is approximated using coarse voxels. A solar radiation (or other) analysis is performed.

Mapping 1: The colored mesh containing the results is imported to Houdini and the resolution is adapted.

Mapping 2: The coarse, voxelized results are mapped onto the initial, fine geometry using closest points.





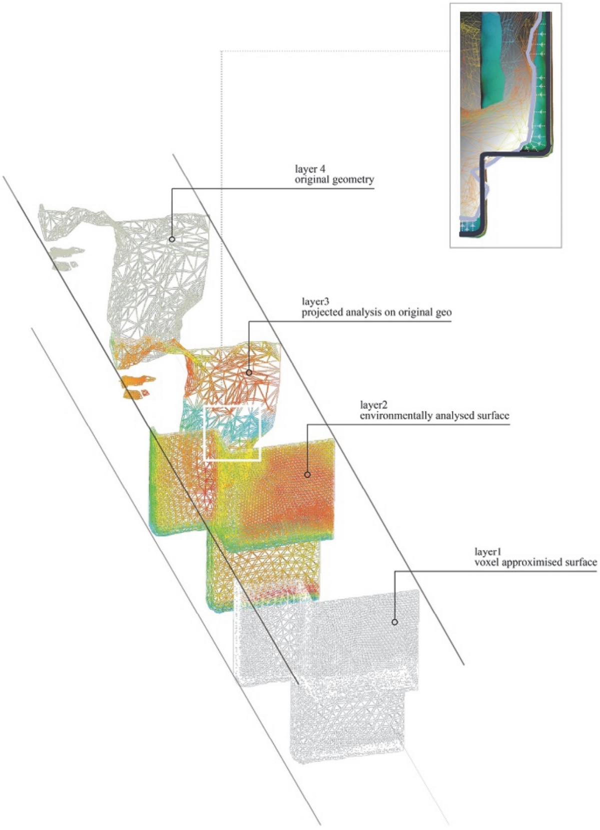


Figure 3
Left: Polygon reduction and mesh optimization from fine to coarse and back.

Right top: In a further step the colour values can be used as a base for other design processes. E.g., to control material deposition in additive manufacturing processes as indicated above.

Right bottom: The colour values of the voxelated mesh are re-mapped onto the initial geometry inside Houdini. Challenges arise when undercuts occur, or when the voxelization deflects too much from the base mesh.

Colour transfer

The colour transfer method affects the output's accuracy, which ranges from minor to considerable, and is directly linked to the projection radius and the number of interpolation samples, between the simulated orthogonal surface and the intricate, original topology. Therefore, it is not solely a transfer process but a mapping method that influences the data output, meaning position on and affected area of the studied geometry. In detail, the utilization of voxel approximation for translating orthogonal cages embedded with data into voxel groups of various resolutions unlocked an additional design potential. The voxel size agility, allows for the combination of different resolutions into one topology, allowing for the concurrency of detailed and abstracted mappings across different regions of the same geometry (Figure 4).

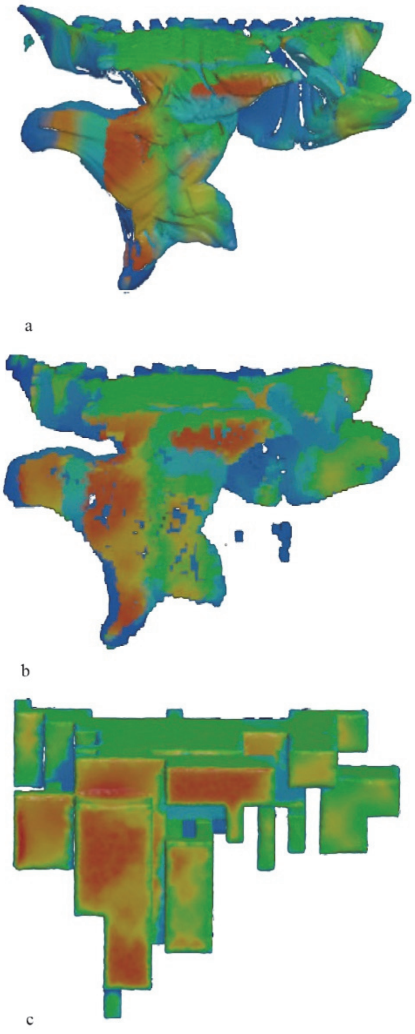


Figure 4
The voxelization process incrementally reduces the number of mesh faces while the general outline is approximated. (a) original geometry (b) voxelated geometry (c) approximated outline for simulation.

DISCUSSION

During the workflow creation process, there were noteworthy challenges faced. One of them regards the HouGH plug-in, used here for data transfer between Houdini and Grasshopper, which was recently released in January 2020. The interface currently holds a considerable limitation regarding the available data types for export, which are restricted to the following: integer, number, and string. Therefore, in the context of the presented research, the colour data were not exported in the form of a 3D vector, which is the colour data encodement type in Houdini, but as a number list. This demanded an extra step of data translation from float to vector.

Furthermore, regarding the selected data type, colour encodement was chosen as the primary data representation type over numerical values as there were extremely high, inaccurate values generated by the analysis software. This error was due to the chosen simulation settings, as convection was omitted for example in the solar heating analysis and could be overcome in more advanced setups. In this case, it was also avoided for computational efficiency purposes, to limit calculation time. Another important reason for this choice was that this type of data encodement further enhances the proposed platform's accessibility. This is achieved by allowing the exploitation of less complex and lighter (processing) simulation tools. Finally, through the proposed workflow the role of optimization through analysis can be redefined in the architectural context. The ability to define and change each step's output resolution is an option provided by the presented pipeline. At the same time, it also provides novel options regarding environmental data handling and integration in multiple steps of the design process. As a result, the promotion of geometrical or material differentiation in several topological regions is enabled, similarly to the concept of Variable Property Design (Oxman, 2010).

Additionally, the possibilities of layering multiple simulations as well as blending between results are feasible paths forward. There is noteworthy improvement potential when it comes to the mapping process onto more complex geometries.

On the one hand, the presented method is a simulation-based design method that raises analysis accessibility when designing highly intricate geometries using advanced design tools. On the other hand, the method can help with scalability in the design process. Therefore, it offers human designers an accessible solution to translate data between design and analysis environments. In doing so, environmental simulations can be implemented into any stage of design workflows and can reduce 'design-downtime' by using lightweight approximated geometries for simulation which are then re-translated back to heavy-weight designs.

Limitations

The research presented here is not a distinct solution to a specific problem. It is rather a suggested method to deal with existing bottleneck conditions and circumvent them creatively. This approach is platform-independent and can be replicated within a wide range of software environments. It adopts a series of tasks that are commonly automated when exchanging files between software and therefore maintains its computational logic. The approach gives users a large amount of control over the translation between form generation and analysis. Compared to other approaches identified in the literature review, the suggested method does not require deep knowledge of software or coding outside the design programs already used. This highlights an opportunity especially in architectural design education when advanced design software is taught already. Consequently, it allows students to negotiate complex environmental challenges without the need for complexity-reduction when designing. Therefore, environmental simulations can be further stripped from a remaining dogma of optimization and used as creative tools in the early stages of the design process (Hensel and Menges, 2008).

CONCLUSIONS

This research holds potential worth future development. The integration of various data types, at different stages of the design process, can lead to the development of computational tools able to merge environmental information and design inside feedback loop systems. This can help to environmentally inform generative designs as well as to make simulation workflows more accessible when working with a wider range of geometries. In this, it reduces the perceived discrepancy between the concept and simulation model. This eases the use and allows a wider audience of users to develop co-creation processes between computation, architecture, and environment.

ACKNOWLEDGEMENTS

The foundation of the research presented in this paper has been conducted as part of Vasiliki Panagiotidou's Master's Research Thesis under the supervision of Andreas Körner in 2020 in the Bio-Integrated Design program at the Bartlett, UCL. The authors would like to thank the program's co-directors Brenda Parker and Marcos Cruz for their valuable feedback during the research and writing process.

REFERENCES

ASWF (2021). *OpenVDB*. Available at: www.openvdb.org (Accessed 21 January 2022)

Ahlquist S. (2016). 'Procedural Design', in *Proceedings of ACADIA 2016*, p. 11.

Autodesk (2022). *Autodesk CFD*. Available at: www.autodesk.com/products/cfd (Accessed 21 March 2022)

Blender (2022). *Volume Object*. Available at: www.wiki.blender.org (Accessed 21 January 2022)

Carpo M. (2016). 'Excessive Resolution', *Architectural Design,* 86(6), pp. 78-83.

Chronis A., Dubor A., Cabay E., and Roudsari M. S. (2017). 'Integration of CFD in Computational Design', in *Proceedings of eCAADe 2017*, pp. 601-610.

Hensel M. and Menges A. (2008). 'Inclusive Performance', *Architectural Design*, 78, pp. 54-63.

Hosain M. L. and Bel Fdhila R. (2015). 'Literature Review of Accelerated CFD Simulation Methods towards Online Application', in *Proceedings of ICAE 2015*, pp. 3307–3314.

Klingemann O. (2017). 'Simulation and Real-Time Design Tools for an Emergent Design Strategy in Architecture', in *Proceedings of Design Modelling Symposium 2017*, pp. 295–309.

Koerner A. (2019). 'Thermochromic Articulations' in *Proceedings of ACADIA 2019*, pp. 332–337.

Ladybug Tools LLC (2022). *Ladybug Tools*. Available at: www.ladybug.tools (Accessed 28 February 2022)

Mackey C. and Roudsari M. S. (2017). 'The Tool(s) Versus the Toolkit' in *Proceedings of Design Modelling Symposium 2017*, pp. 93–101.

Maxon (2022). *Cinema 4D*. Available at: www.maxon.net (Accessed 21 January 2022)

McNeel R. & Associates (2022). *Rhinoceros.* Available at: www.rhino3d.com (Accessed 27 March 2022)

Miroljub A. (2022). *HOUGH*. Available at: www.food4rhino.com/en/app/hough (Accessed 28 March 2022)

Mokhtar S., Sojka A., Cerezo Davila C. (2020). 'Conditional Generative Adversarial Networks for Pedestrian Wind Flow Approximation', in *Proceedings of SimAUD 2020*, pp. 463–470.

Museth K. (2013). 'VDB', *ACM Transactions on Graphics*, 32(3), pp. 1-22.

Oxman N. (2010). 'Material-based Design Computation', PhD thesis, MIT, Cambridge.

Peters B. (2018). 'Defining Environments', *Architectural Design,* 88(1), pp. 82–91.

Peters T. and Peters B. (2018). *Computing the Environment*. Hoboken: Wiley.

Schumacher P. (2018). 'Tectonism in Architecture, Design and Fashion', *Architectural Design*, 87, pp. 106–113.

SideFX (2022). *Houdini.* Available at: www.sidefx.com (Accessed 3 January 2022)

SimScale GmbH (2020). *SimScale*. Available at: www.simscale.com (Accessed 21 January 2022)

Wortmann T. and Schröpfer T. (2019). 'From Optimization to Performance-Informed Design', in *Proceedings of SimAUD 2019*, pp. 261–268.

Yang D., Sun Y., Di Stefano D. and Turrin M. (2017). 'A computational design exploration platform supporting the formulation of design concepts', in *Proceedings of SimAUD 2017*, pp. 35–42.

Zaghloul M. (2017). 'Machine-Learning aided Architectural Design', PhD thesis, ETH Zürich, Zürich.