Towards DesignOps Design Development, Delivery and Operations for the AECO Industry.

Marcin Kosicki[1], Marios Tsiliakos[1], Khaled ElAshry[1], Oscar Borgstrom[1], Anders Rod[1], Sherif Tarabishy[1], Chau Nguyen[1], Adam Davis[1], Martha Tsigkari[1]

[1] Foster + Partners, 22 Hester Road, London SW11 4AN, UK
mkosicki@fosterandpartners.com

Abstract. The overwhelming success of companies build on top of cloud computing technologies has been driven by their ability to create systems for processing big data at scale and designing high-quality digital products as well as being agile and capable of handling constant changes in the market. This runs somewhat contrary to the AECO industry, which generates an abundance of multidisciplinary data and faces numerous design challenges but is not as prone to agile management. The entire methodology for designing and delivering projects has historically been oriented toward getting all requirements defined and specified in advance. In that context, "change" of the workflow in AECO is often seen as an exception. Not only this is far from the paradigm or principles of today's business technologies, but today's enterprises are characterized by an opposing set of values. Latest software engineering methodologies, like DevOps and its design incarnation – DesignOps were created solely to tackle those issues in the IT industry. This paper will present how those methodologies could be successfully implemented in the AECO industry and increase the efficiency of existing design pipelines. We demonstrate a prototype of a software platform, an entire automated ecosystem where design operations are made in the cloud by a collection of automatic or semi-automatic microservices and where data flows seamlessly between various disciplines. The system leverages the potential of distributed computing, performance-driven design, evolutionary optimization, big data, and modern web design.

Keywords: DesignOps, DevOps, Cloud Computing, Performance Design, Optimization, High Performance Computing.

1 Introduction

Design teams are frequently required to deliver iterations of their designs to the respective stakeholders, such as internal reviewers, clients, building authorities or consultants. This usually involves a time-consuming iterative process associated with the design cycle and its derivatives -such as documentation, costing or visualization. Some of those steps require intensive human input and analysis while some could be automated using various technologies. Since new design challenges constantly arise, these pipelines and the workflows build around them can never be static, they must be flexible to keep up. This poses many challenges like the ones that the user experience and interface (UX/UI) community is facing and could be seen in a wider context of Design Operations – DesignOps [1]. It is an emerging movement advocating a much closer integration between design and technology escalated by the rise of agile development. It is understood as a practice of reducing operational ineffectiveness in the design workflow, as well as providing better quality design through technological advancement. DesignOps is about implementing design improvements and deploying them to users (designers) as quickly and as frictionless a way as possible. It is still in its formational stages and is an intentionally broad topic, because there are many elements to factor in when enabling consistently good quality design.

1.1 Road to DesignOps

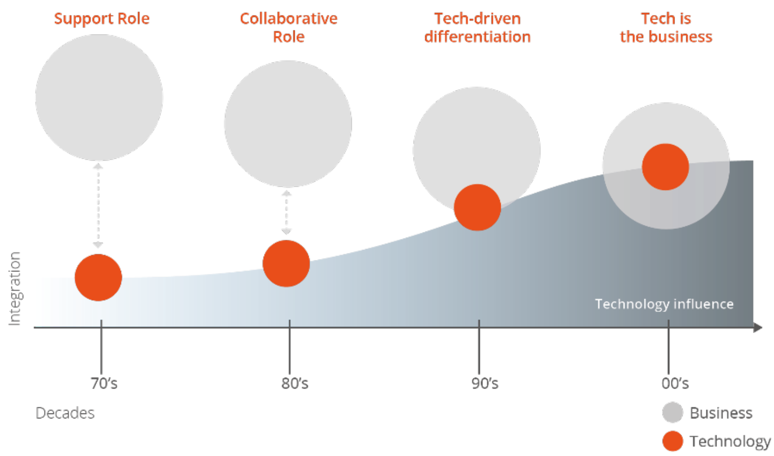

Fig. 1. Changing role of technology.

Understanding the principles of the DesignOps movement, requires a step back to look at how the role of IT in the business environment has evolved over the years (Fig. 1). From the 1960s and 1970s when it played primary a supportive role, where IT systems were used solely to make the existing processes faster and reduce cost [1]. Through the 1980s some forms of collaboration started to emerge, since the spread of PCs put machines on peoples' desks. In response, companies started asking for customized software. However, IT was still considered a tool which resulted in a very clear separation between development teams (people building IT systems) and operations (business users). The need for custom tools stared pulling IT teams closer to the operations side and by the mid 90's IT had emerged as an enabler which could provide competitive advantage through technology-driven differentiation. The trend continued through the dotcom explosion in the 00's as technology became the core of many businesses. Some companies did not simply differentiate themselves through technology but delivered technological products and solutions as their sole function. This required more robust processes and workflows which culminated in the idea of agile software development formed in 2001. Adopting agile methodologies pushed many organizations like Spotify [2] to product-centric rather than project-oriented development. It promoted fast delivery cycles [3] further reinforced by build-measure-learn feedback loops using Minimum Viable Product ideas (MVPs) from Lean Start-Up principles [4].

Agile also had its shortcomings since success was primarily measured on time and budget delivery and less on the quality of the product. This resulted in the operations (Ops) teams becoming increasingly skeptical about the quality of new releases from the development (Dev) teams [1]. Promoting closer collaboration resulted in DevOps, a methodology aimed at removing uncertainty by aligning development and system operations through automation. Increased demand for the delivery of technological products at scale and short design cycles started putting pressure on user experience (UX) and user interface design. Designers often worked in silos or as remote parts of dev teams and were sometimes individually responsible for a wide range of tasks from conducing UX research, wireframing to front-end coding. This loose structure was not scalable and could not cope with growing complexity of digital products or the demand for high-quality, consistent user customer experience. As a result, a separate set of processes and workflows attempting to operationalize design, inspired by DevOps had emerged – DesignOps [5].

DesignOps has been adopted by many organizations including Airbnb [6] and Salesforce [7]. However, each of them tailored it to their own specific business needs, DesignOps as an idea is centered around four main goals [8]: building efficient design workflows and processes, ensuring cross-functional team collaboration and alignment with different stakeholders ranging from engineering, UX researchers, UI designers, motion designer to marketing; standardising tools and systems as well as promoting consistent design culture. Although many of those activities had already existed within those companies, DesignOps as a separate role introduced structural changes and cultural shift focused mainly on scaling and amplifying design processes. Instead of segregating teams, it is seen as a methodology that enables highly integrated and effective design organizations.

1.2 AECO perspective

There are many similarities between challenges addressed by DesignOps and the AECO industry. In architectural practices the design cycle timeline is a key factor in the success of any given project. The quicker a team can come up with design ideas and turn them into a viable design option, the better. Design is an iterative process; ideas are brainstormed then modelled and tested against others. As a design progress from the initial concept stage to final construction drawings, its concepts become more defined. This means that change comes with a constantly increasing overhead since design interventions can depend on decisions made earlier on in the process.

It is relatively easy to make changes during concept stages as they provide high flexibility, but the assumptions made during concept design can be critical to the future performance and operational cost of a buildings [9] [10]. Therefore, there is a constantly growing demand to guide the exploration of new design options not only based on their aesthetical criteria but also on hard metrics like environmental performance or user-experience. Such studies are usually conducted by consultants using specialized simulation software and the current process usually requires a design team to pass a digitized 3D massing model to a consultant who will run the required analysis and put together a report. This is a cumbersome workflow since massing options must be exported form the CAD software and sent over manually. At the same time, running performance analyses in a timely fashion comes with its own challenges. Models often need to be converted, the analysis can be computationally intensive and putting together a report which includes detailed feedback is usually a slow, manual process. All those steps add up to a considerable lag between the modeling of a design option and the understanding of its performance. The lag in the delivery cycle grows even more if the aspiration is to integrate data from many disciplines. It could take days or even weeks and by the time the performance of the design option is evaluated the actual design has progressed, rendering the returned feedback obsolete. This process is neither fast nor scalable and it is the result of workflows built around AECO software's that still follow a waterfall model [11]. By learning from the DevOps and DesignOps movements we can build more robust pipelines and deliver value at a much earlier stage in the project development cycle.

2 Methodology

To address these design cycle challenges, we have developed an approach that treats early-stage design like a modern software delivery cycle. In our approach an architectural design team would be analogue to the IT Ops team and a consultant team analogue to the IT Dev team. The product in this context is a set of highly optimized and performance-driven massing options tailored to a given architectural project and delivered at scale during early design stages.

The initial experiments started in late 2017 and were focused on accelerating existing performance analysis simulations using distributed computing. The initial goal was to build a pipeline inside a popular parametric CAD software which would execute a ray-tracing-based simulation with increased speed in the cloud, yet as seamlessly as the existing pipeline would on a single workstation. The case-study, which was a tower, provided a 3.8 times speedup by using 5 machines with 20 cores (40 threads) each in the cloud compared to a single 20-core Intel(R) Xeon(R) CPU E5-2660 v3 @ 2.60GHz with 64 GB of RAM (Fig. 2).

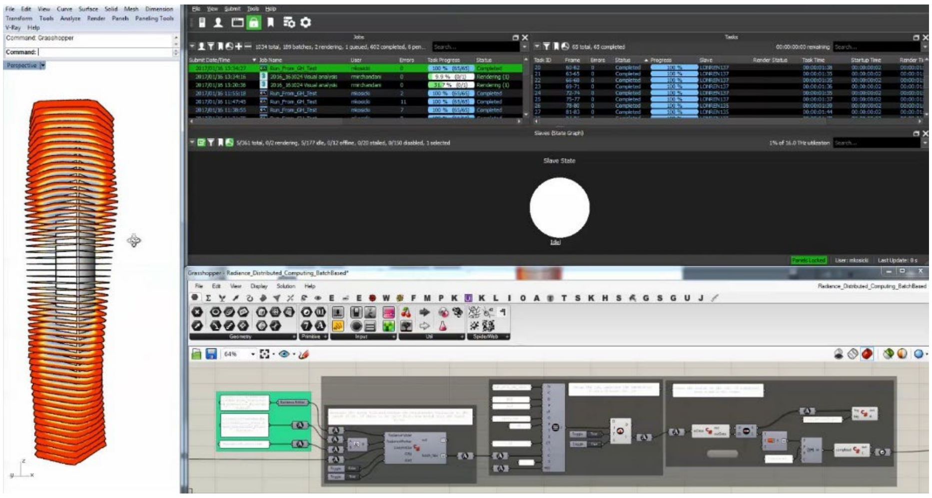

Fig. 2. Distributed Daylight Factor simulation. Each test floorplate was simultaneously calculated by 5 machines with 40 threads each using Rhino and Grasshopper software as an interface.

The success of this case-study sparked many ideas on how this could be efficiently scaled up to run various complex analysis even on large-urban scale models. Such an approach would not only need to automatically coordinate jobs on hundreds of machines simultaneously but also communicate and store the generated data between them while visualizing the results. In this scenario, a rigorously tested and optimized massing option becomes a product of a software platform – an entire automated ecosystem where design operations are made by various automatic or semi-automatic microservices [12] and data flows seamlessly between various disciplines. Some of the components in such a system like classic parametric modeling [13] with performance simulations [14] [9] and multi-objective optimization [15] or interactive data dashboards [16] are not new to the AECO industry. However, combining them with distributed computing at scale and both modern web and cloud technology would require considerable new research far exceeding traditional computational design. Since there is no architectural software that could take full advantage of such a technology stack, coordination of knowledge and expertise from different domains including full-stack software development, data science or parallel computing [17] would be required. Both DevOps and DesignOps workflows and practices would provide an efficient framework for how to drive this coordinated cross-domain software development effort while ensuring both consistency and high-quality design output.

Having that in mind, the first prototype of the platform was launched in early 2018 and tested on a large-scale urban project [18]. It had the core components in place namely a database with interoperability models, an evolutionary optimization solver capable of distributing performance simulations and a rudimentary webpage to display the results. The components were split into micro-services ensuring that in the future small but quick and incremental changes could be implemented and immediately tested, using Continuous Integration and Deployment on live projects. The approach was successful and set out a roadmap for future development to improve the speed and reliability of the platform.

2.1 Producing Design Data at scale

Interoperability. Having a common way of communicating coherent information between the various parts of the platform is essential for an uninterrupted workflow. In AECO different teams and disciplines use different tools and data exchange formats, which is a major obstacle for automation. For that reason, we chose to integrate our inhouse interoperability standard based on JavaScript Object Notation (JSON) [19] handling both geometry and analysis data. We developed shared data schemes which were made available to all disciplines and users in a format familiar to them through our bespoke messaging application. This eliminates the need to exchange static files in the traditional way. Interoperability allows us to streamline, compress and modify design data on demand through different components of the platform. It also enables any geometry of selected design options to be seamlessly imported directly into a CAD package and evaluated by an architectural team. The direct feedback was used to modify both parametric model and analytical objectives.

Simulation Engines. Essential to the success of a delivery cycle is the speed of which the relevant performance simulations can run. Some the most common analyses in architecture are based on raytracing. It is a key component in calculations of the vertical sky component which assess daylight potential, of sunlight hour analyses and when calculating annual solar radiation. Initially the platform used third party simulation software like Radiance [20] through the Ladybug plugin [21] and distributed the analysis on CPUs. But since raytracing also is a key component for creating renderings, which are used to visualise design options and assess quality of views for each option, we decided to develop a bespoke ray tracing software. This software supplies highly parallelized and performant analysis and rendering pipelines by utilizing Graphical Processing Units (GPUs). It implements the CUDA-centric [22] Nvidia Optix API [23], takes advantage of RTX-technology [24] and provides a comprehensive raytracing framework combined with lightweight scene representation.

With control over both geometry, ray generation, ray intersections and data transfer, the engine enables a flexible multi-GPU workflow which can easily be extended with new analysis pipelines to fit the varying needs of design projects. These pipelines can reach speeds of up to 15 giga rays/s on a single Nvidia A6000 GPU and run city scale models with hundreds of thousands of analysis points in seconds. The speedup offered by these bespoke GPU analyses does not only greatly improve the efficiency of the delivery cycle, but it also allows to run analyses at a higher resolution. Compared with previous studies [11], the analysis runs on average 6 times faster while being more accurate, with a 260x higher resolution.

2.2 Automated Reporting

The ability to quickly interpret and display results from tests and simulations is a key component in every feedback system. From extensive discussion with design teams, we developed a two-level reporting.

In the first stage a fast, triage-like, near real time feedback from the simulations was need. Since the platform could generate and test thousands of design options within hours, a tool for traversing their performances across key objectives and comparing them against each other was required. The tool had to be interactive and easily accessible to all stakeholders, so it was built as modern web application running in a web browser. For the front-end technologies like Bootstrap, developed initially at Twitter for responsive and consistent looking layouts, and D3.js for data-driven visualisation were used. Design data was pulled from options' database using JSON-based interop data formats.

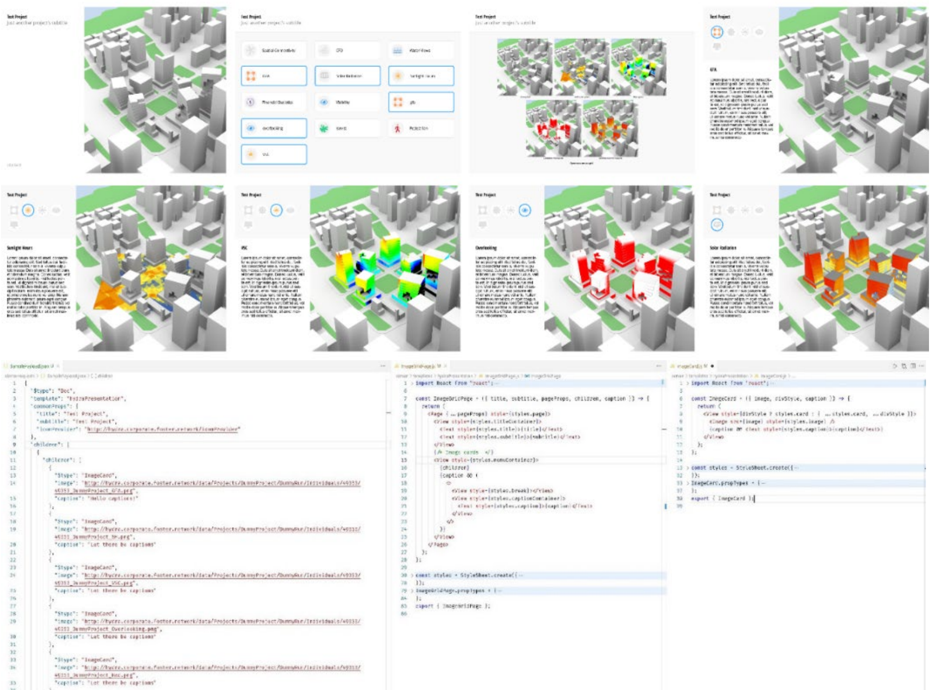

Fig. 3. Sample payload sent from a client to the microservice defining the template to be used and the component types the template exposes. Sample template for a page containing a grid of captioned images in JSX.

When candidate options were selected, a second more in-depth feedback stage was required in the form of a detailed PDF report. To previously produce such reports required considerable manual work. A specialist software such as Adobe InDesign had to be opened to place images, adjust a layout, write descriptions and analysis results while ensuring consistent graphical quality, something that could take 2-3 hours per report. A manual workflow would therefore be highly inefficient, given that the platform delivers thousands of analyses results. Thus, the process was automated by creating a reporting microservice driven purely from data using Node.js technology. It was developed using React [25], a front-end JavaScript library for building user interfaces created and maintained by Facebook (Meta). The service exposes different endpoints for document types which, based on JSON data, could automatically return a PDF document within a minute (Fig. 3).

3 Findings – The Process

The development of the platform has been highly successful. Since 2018 it has been used on 23 projects ranging from 50,000 to 2.17M sqm of GFA, most of whom were large-scale masterplans. It has generated over 300,000 design options and conducted 1.3M performance simulations. The decision to break down early design stages to subtasks such as project-specific massing creation, performance simulations, reporting and decision making and develop them as independent microservices fall under the DevOps model. As development progressed, the overall optimization design cycle time for a single project was significantly reduced, from 20 to 4 days. A development that could be implemented while the entire system was online, sometimes testing up to 140,000 options per project (Fig. 4). This was possible due to significant improvements in robustness and standardization of the analysis and reporting pipelines as well as to the progressive development of adaptive parametric models covering main building typologies. It also allowed further reduction in the time required to build custom parametric models for live projects from weeks to days. In recent projects the full cycle of model updates and simulation runs has decreased from the initial 1.5 months to 2-3 days per project while handling up to three large-scale optimizations projects simultaneously.

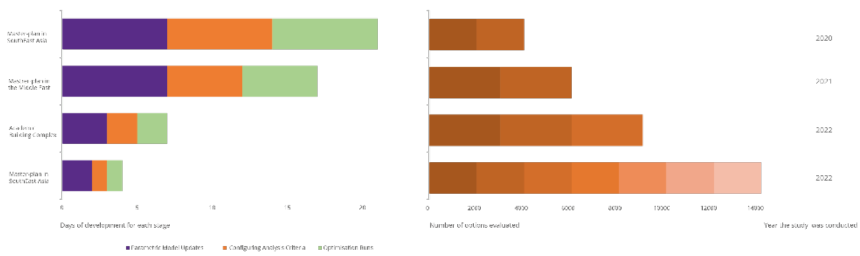

Fig. 4. Comparison between single design cycle time and a total number of options tested for an indicative project.

A good example of the capabilities in the DesignOps approach is the studies prepared for the Guangming Hub project. The project was a large-scale transport-oriented (TOD) design competition in China [11], that required four site-specific typologies covering a total GFA of 2,700,000 m2. The challenge was to produce a wide variety of massing options while maximizing environmental performance, cutting down average walking times and increasing views of greenery. For this study, three agile-like design optimizations sprints were conducted producing 18,000 options and running 80,000 analyses in total. Design solutions derived from this process and selected by the architectural team from each iteration were refined and used as basis for subsequent optimisations. The results from the last run were then manually finetuned and postprocessed by the architectural team to become the basis of the final massing distribution for the comple-tion-wining proposal.

4 Conclusions

As demonstrated in the case study, the use of recent advancements in cloud computing supported by implementation of the latest software development standards, like DevOps and its design incarnation – DesignOps, can be transformative to architectural design. It has the capacity to reduce delivery cycles from months to hours by automating many complex design steps. This in turn can give architects an opportunity to re-think current design pipelines and workflows by making them more efficient and data oriented.

Now, this approach requires cross-domain teams of experts collaborating on a single software platform as well as a significant investment in dedicated hardware and high-performance computing infrastructure. Another obstacle is standardisation of both work stages and data formats which has traditionally been a major issue in the AECO industry and makes this methodology initially more suitable for large organisations which integrate many disciplines under a single roof. However, the wide spreading access to massive cloud computing power and data storage for a fraction of the previous cost is beginning to blur the boundaries and will in turn result in a wave of new cloud-based applications that expose elements of design-related functionality as a service. They could then be consumed by various stakeholders and integrated into their design workflows. Their adoption would be a direct reflection of both the design and data culture of the organisation that implements them. The implementation would require a comprehensive DesignOps approach including the combination of philosophy and tools of a given organization which in turn can facilitate an organization's capacity for delivering high-quality designs at scale and at a much faster pace than traditional practices.

References

[1] E. Dornenburg, "The Path to DevOps," *IEEE Software,* pp. vol 35. no.5 pp. 71-75, 2018.

[2] H. Kniberg and A. Ivarsson, "Scaling Agile at Spotify with Tribes, Squads, Chapters and Guilds," 2012. [Online]. Available: https://blog.crisp.se/wp-content/uploads/2012/11/SpotifyScaling.pdf.

[3] K. Beck, M. Beedle, A. van Bennekum, A. Cockburn, W. a. F. M. a. G. J. a. H. J. a. H. A. a. J. R. Cunningham, J. Kern, B. Marick, R. C. Martin and S. Mellor, "Manifesto for Agile Software Development," 2001. [Online]. Available: http://www.agilemanifesto.org/.

[4] E. Ries, The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses., New York: Crown Business, 2011.

[5] D. Malouf, "DesignOps Handbook," 2017. [Online]. Available: https://www.designbetter.co/designops-handbook/introducing-designops.

[6] A. Cleave, "DesignOps at Airbnb, How we manage effective design at scale," 2018. [Online]. Available: https://airbnb.design/designops-airbnb/.

[7] Salesforce, 2020. [Online]. Available: https://medium.com/salesforce-ux/team-ops-and-product-ops-the-perfect-designops-pair-eb6508f1b435. [Accessed 11 06 2022].

[8] B. Nick, "Shopify Partners," Shopify, 13 12 2018. [Online]. Available: https://www.shopify.co.uk/partners/blog/designops. [Accessed 11 06 2022].

[9] S. Li, L. Liu and C. Peng, "A Review of Performance-Oriented Architectural Design and Optimization in the Context of Sustainability: Dividends and Challenges," *Sustainability,* p. vol. 12 no.4/1427, 2020.

[10] P. Stamatios, M. Kosicki, K. El-Ashry, S. Tarabishy, M. Tsigkari, A. Davis and F. Aish, "SandBOX - An Intuitive Conceptual," in *Impact: Design With All Senses. DMSB 2019*, Cham, 2020.

[11] M. Kosicki, M. Tsiliakos, K. ElAshry and M. Tsigkari, "Big Data and Cloud Computing for the Built Environment," in *Industry 4.0 for the Built Environment* , Springer, 2022, pp. 131-155.

[12] S. Newman, Building Microservices, Sebastopol : O'Reilly Media, 2015.

[13] D. Martyn, "Rhino Grasshopper," *AEC Magazine,* pp. 14-15, 42(May/ June) 2009.

[14] A. Nguyen, S. Reiter and P. Rigo, "A Review on Simulation-Based Optimization Methods Applied to Building Performance Analysis," *Applied Energy,* no. 113, , pp. 1043-1058, 2014.

[15] D. Rutten, " Galapagos: On the Logic and Limitations of Generic Solvers," *Architectural Design,* pp. 132-135, 2013 .

[16] A. Chaszar, P. von Buelow and M. Turrin, "Multivariate Interactive Visualization of Data in Generative Design," 2016.

[17] J. Branke, H. Schmeck, K. Deb and S. Reddy, "Parallelizing multi-objective evolutionary," *Proceedings of the 2004 Congress on Evolutionary,* p. 1952–1957, 2005.

[18] M. Kosicki, M. Tsiliakos and M. Tsigkari, "HYDRA: Distributed Multi-Objective Optimization for Designers," in *Design With All Senses. DMSB 2019*, 2019.

[19] D. Crockford and C. Morningstar, "Standard ECMA-404 The JSON Data Interchange Syntax". Patent 10.13140/RG.2.2.28181.14560, 2017.

[20] G. J. Ward, "The RADIANCE Lighting Simulation and Rendering System," in *Computer Graphics (Proceedings of '94 SIGGRAPH conference)*, 1994.

[21] M. Roudsari, M. Pak, A. Smith and G. Gill, "Ladybug: A parametric environmental plugin for grasshopper to help designers create an environmentally-conscious design," in *Proceedings of BS 2013: 13th Conference of the International Building Performance Simulation Association*, 2013.

[22] J. a. K. E. Sanders, CUDA by example: an introduction to general-purpose GPU programming, Addison-Wesley Professional, 2010.

[23] S. G. Parker, J. Bigler, A. Dietrich, H. Friedrich, J. Hoberock, D. Luebke, D. McAllister, M. McGuire, K. Morley, A. Robison and M. Stich, "OptiX: a general purpose ray tracing engine," *ACM Transactions on Graphics,* vol. 29, no. 4, 2010.

[24] J. Burgess, "Rtx on—the nvidia turing gpu," *IEEE Micro,* vol. 40, no. 2, pp. pp.36-44, 2020.

[25] React, 2022. [Online]. Available: https://reactjs.org/.