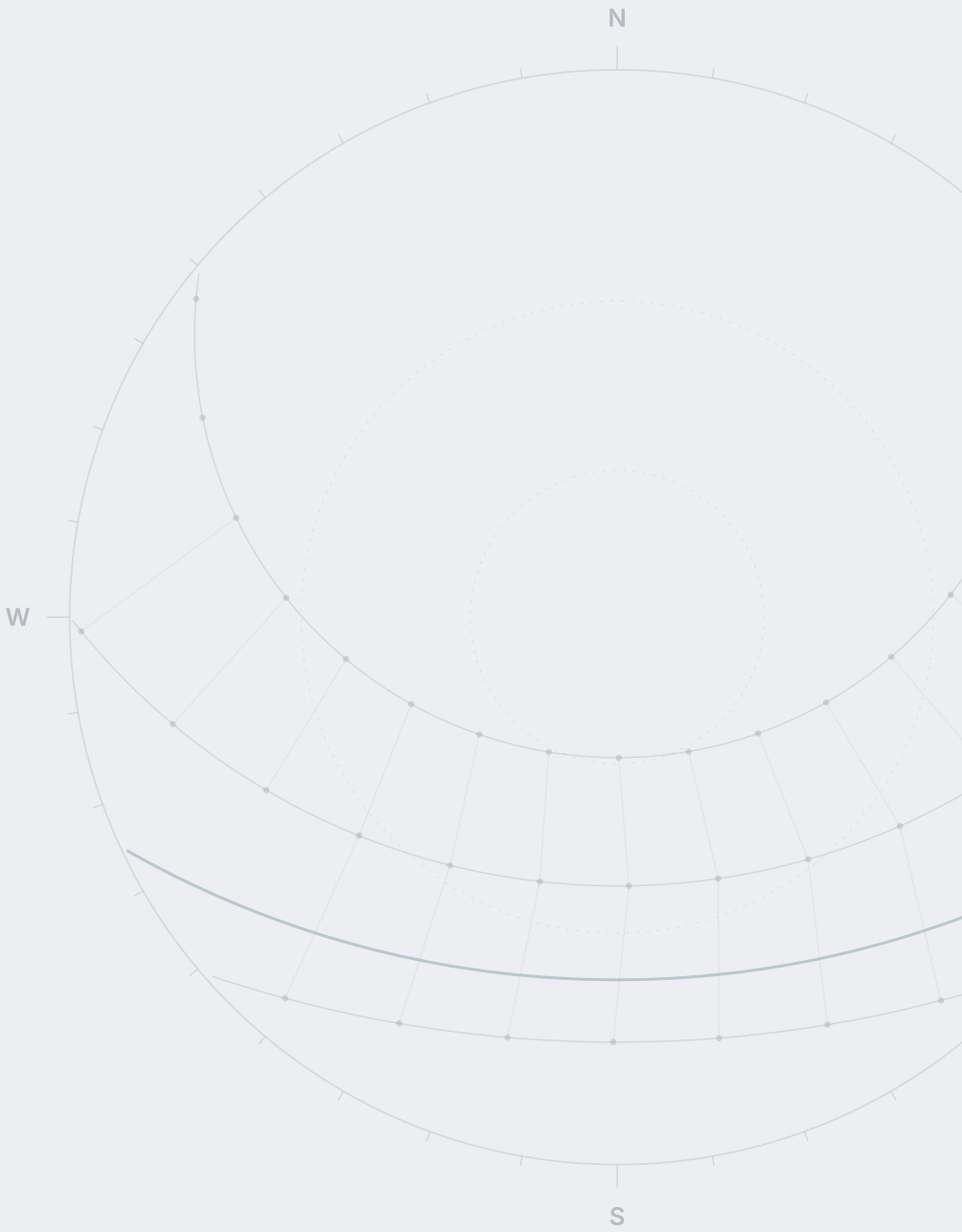
Mateusz Płoszaj-Mazurek

Architect, PhD. I build computational design tools for architects: Grasshopper definitions and plugins, Revit add-ins, web apps and fast prediction models, switching between C#, TypeScript with React, and Python as each job needs.

> Every project here takes a computational approach, and most start in Grasshopper: that is where I generate geometry, run simulations, build datasets and test rules.

01	Solaie: sunlight-access checks in Revit	own product
02	Lookout tower: parametric steel structure	Grasshopper
03	Hatlehol Church: parametric timber roof	Grasshopper, Robot
04	OmegaFlux: daylight prediction in milliseconds	ML surrogate
05	OmegaFlux back in Grasshopper: Hops and optimization	workflow
06	Carbon Designer 3D: engineer and product owner	product team
07	3D IFC web viewer for historical buildings	client work
08	LLM-assisted IFC data enrichment	published
09	Parametric daylight regulation studies	research
10	Open data and machine learning in Grasshopper	tools, PhD
11	Grasshopper practice: teaching, tools and built work	overview



Warsaw, Poland · mateuszploszajmazurek@gmail.com · +48 695 577 433
curiosit.github.io/Curiosit · github.com/Curiosit · solaie.pl

01 Own product · 2026 · solaie.pl

Solaie: sunlight-access checks in Revit

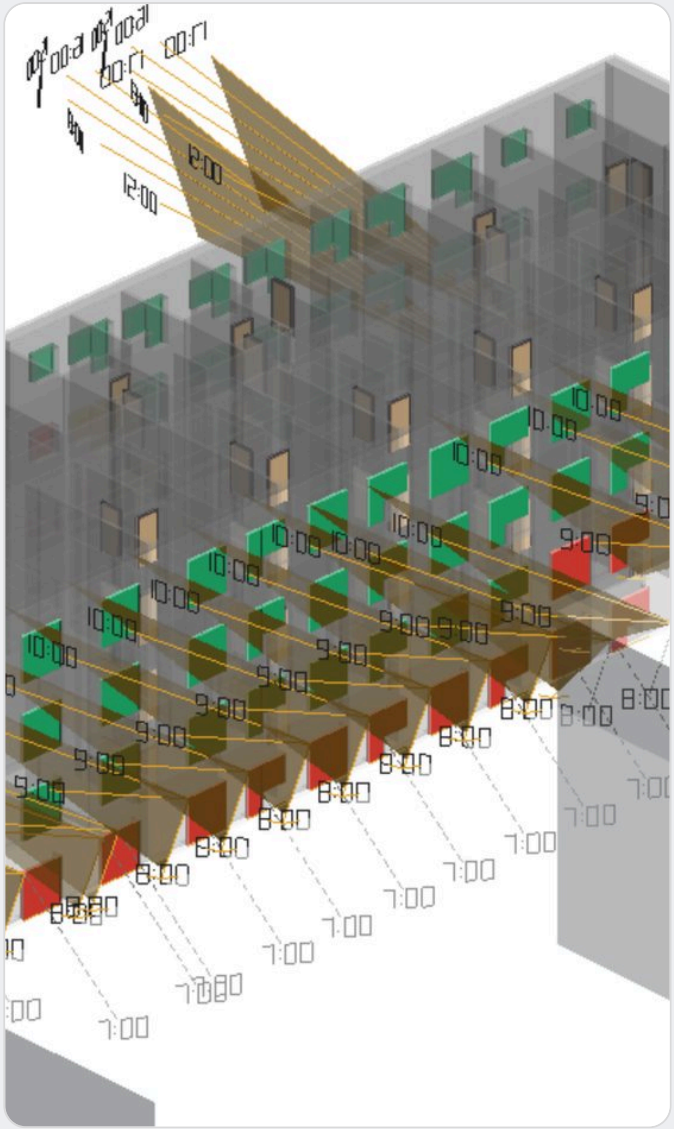
A Revit add-in that computes how long each room receives direct sun under § 60 of the Polish technical conditions, on both equinoxes, and flags rooms whose verdict depends on how the rule is read.

Stack	C#, .NET and WPF add-in with a compiled Python analysis engine
Engine	Model scanning, geometry export, vectorised ray tracing against context meshes
Output	Room-by-room report for the building authority, signed time-limited licences
Status	Public website live; test builds sent to the first architecture offices

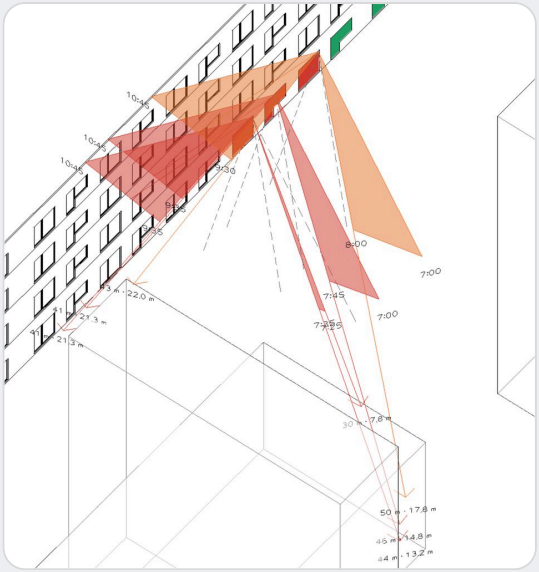
Prototyped in Grasshopper

The rule engine was built and tested as a Grasshopper prototype first, then moved into Revit, where the client offices work.

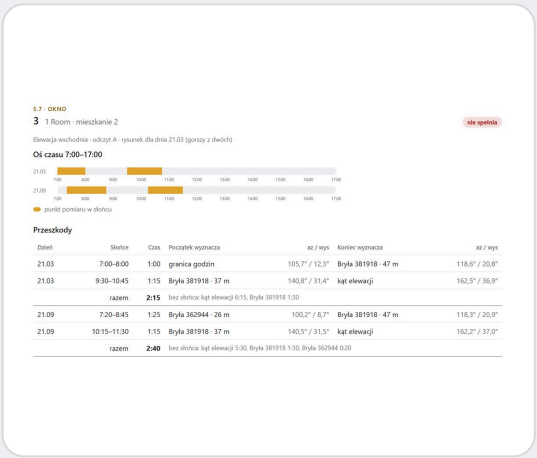
Grasshopper prototype › Python engine › Revit add-in (C#) › Report



Solaie in Revit: sun wedges from every window on the equinox, coloured by verdict, with the obstructing buildings.



One window: sun periods, the obstructions that end them, and distances.



Report entry per window: timeline and obstructions on both equinoxes.

02 Parametric structure · steel

25 m lookout tower, designed in Grasshopper

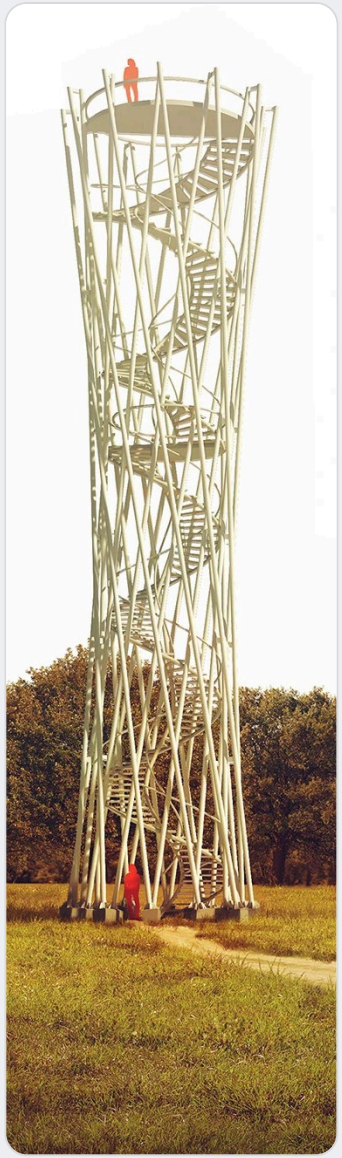
An hourglass-shaped tower of steel tubes in two structural shells, inner and outer, twisting around a spiral stair to an observation platform at 24.3 m.

Height	25.6 m, platform at 24.3 m
Structure	Two shells of Ø150 mm steel tubes on a continuous footing
Stair	Steel stair on two twisted beams that narrow with the tower

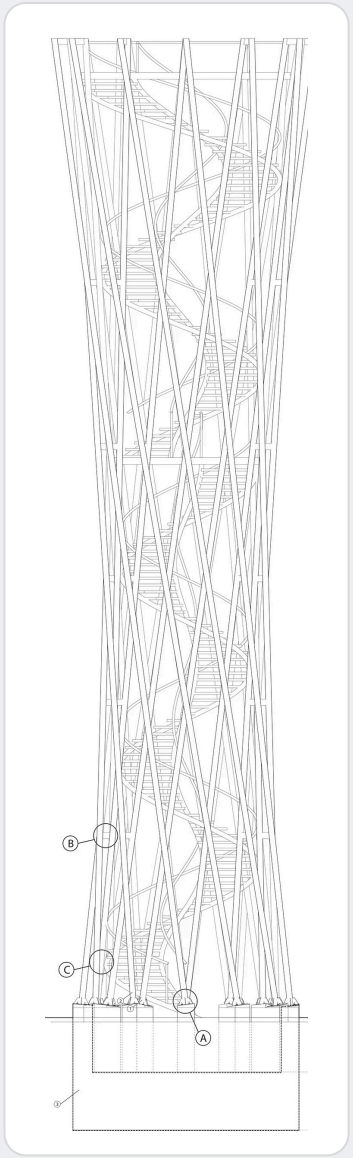
Built in Grasshopper

The tower was modelled as one parametric system: both tube shells and the stair share the same form, so twist and proportions could be tuned together.

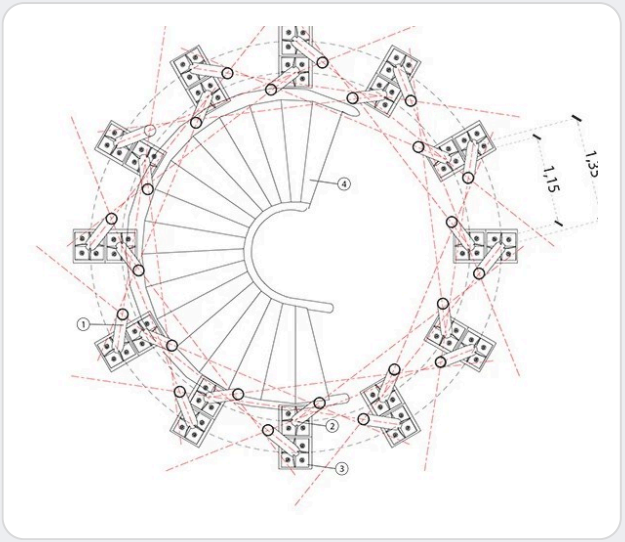
Grasshopper › Two shells › Stair › Drawings



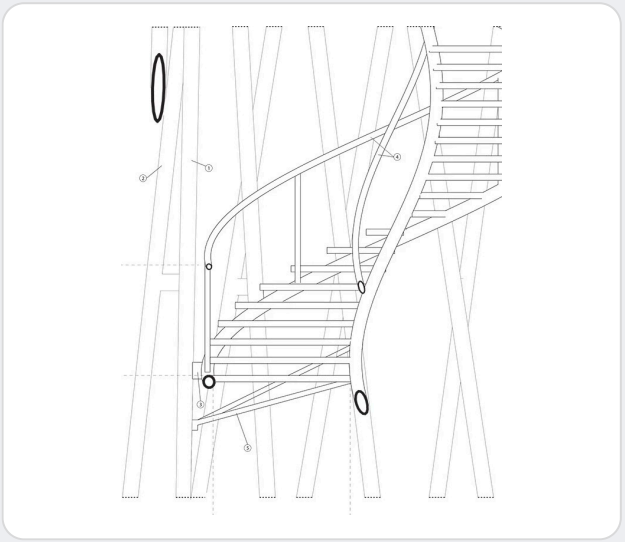
Visualisation.



Elevation, 1:50.



Ground floor plan: inner and outer tube rings.



Stair detail: twisted stringers between the shells.

Hatlehol Church: a roof structure from Grasshopper

The timber roof of the church was developed as a parametric structure: a series of A-frame section options, then a full frame model with purlins and bracing, checked in structural analysis.

From section to structure

Section studies set the form, the Grasshopper model generated the frames and members, and the structure was calculated in Autodesk Robot.

Result

Exposed frames that carry the roof and also shape the interior light and rhythm of the nave.

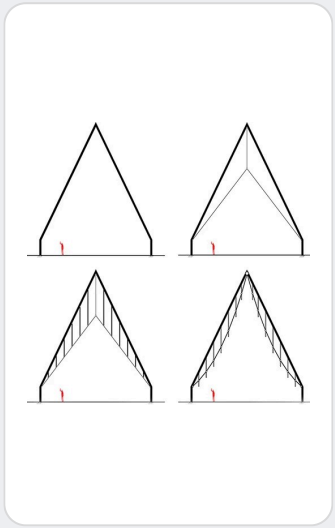
Built in Grasshopper

Geometry and member layout came from one definition, so form and structure changed together during design.

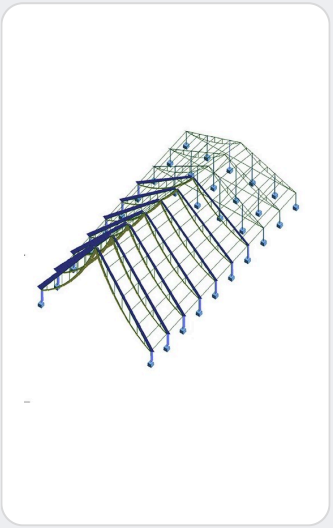
Grasshopper › Frame model › Autodesk Robot › Timber roof



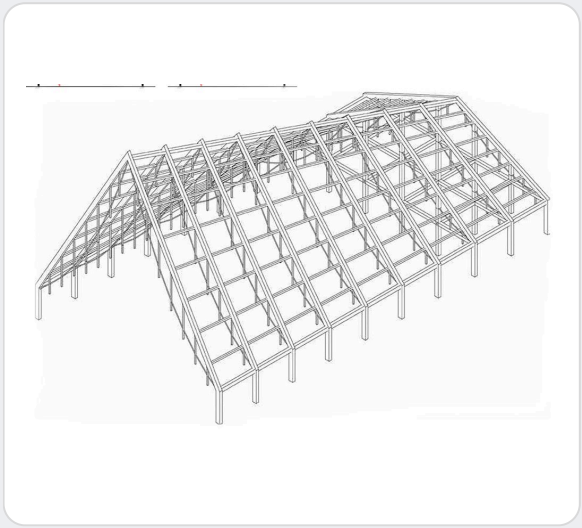
The nave under the exposed timber frames.



Section options.



Structural model in Autodesk Robot.



Roof frame generated in Grasshopper.

04 ML surrogate · 2025-2026

OmegaFlux: daylight prediction in milliseconds

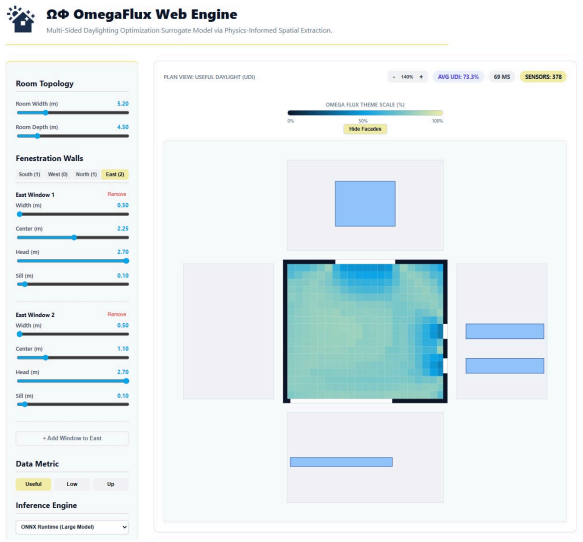
A machine-learning surrogate that predicts climate-based useful daylight illuminance point by point from room geometry, without running a simulation.

Accuracy	R^2 0.986 on held-out rooms, validated against Radiance
Speed	About 200 times faster than the simulation it replaces
Dataset	6,749 rooms, 3.27 million sensor points
Limits	R^2 0.72 on room shapes unseen in training, reported openly
Code	github.com/Curiosit/OmegaFlux · Zenodo DOI

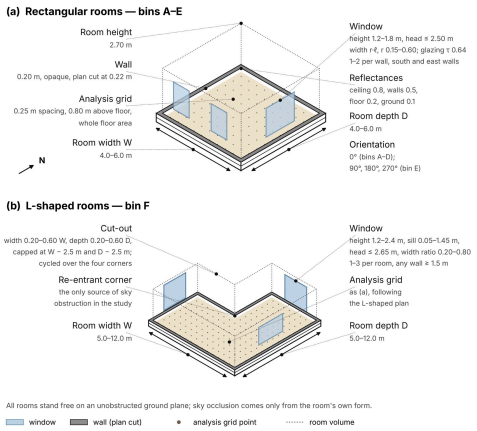
Built in Grasshopper

The whole dataset is a Grasshopper definition: Latin hypercube sampling drives the room geometry, Honeybee runs Radiance, and the per-point geometric descriptors are computed from the same model.

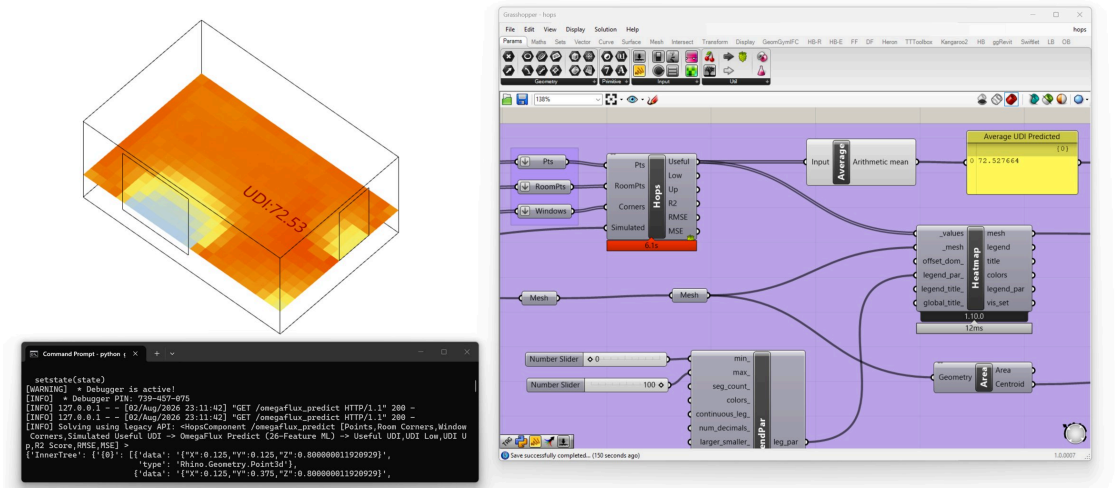
Grasshopper › Honeybee / Radiance › Python, XGBoost › ONNX › Browser



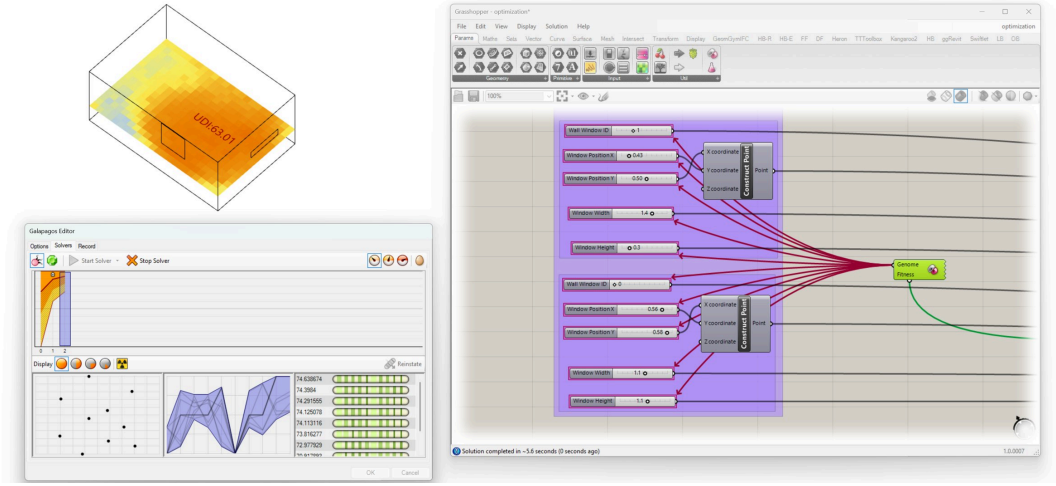
Browser app: edit the room and its windows; the daylight map updates in about 70 ms.



Parametric dataset design: rectangular and L-shaped room families, all generated in Grasshopper.



Grasshopper calls the prediction service through Hops; results return as a heatmap on the room.



Galapagos searches window position and size against predicted daylight, seconds per iteration.

05 Workflow

Fast prediction, back inside the design model

A surrogate is only useful where designers work. OmegaFlux returns to Grasshopper in two ways.

Hops service

A local Python service answers Grasshopper in about 4 seconds per room, against roughly a minute for a full Radiance run.

Optimization loop

Each evaluation is fast enough for an evolutionary solver to search window layouts interactively, which full simulation makes impractical.

The pattern I reuse

Any slow analysis can follow it: generate variants parametrically, train a fast model, and wire it back into Grasshopper with its limits stated.

Grasshopper › Hops › Python service › Galapagos

o6 Product team · One Click LCA · 2024-2026

Carbon Designer 3D

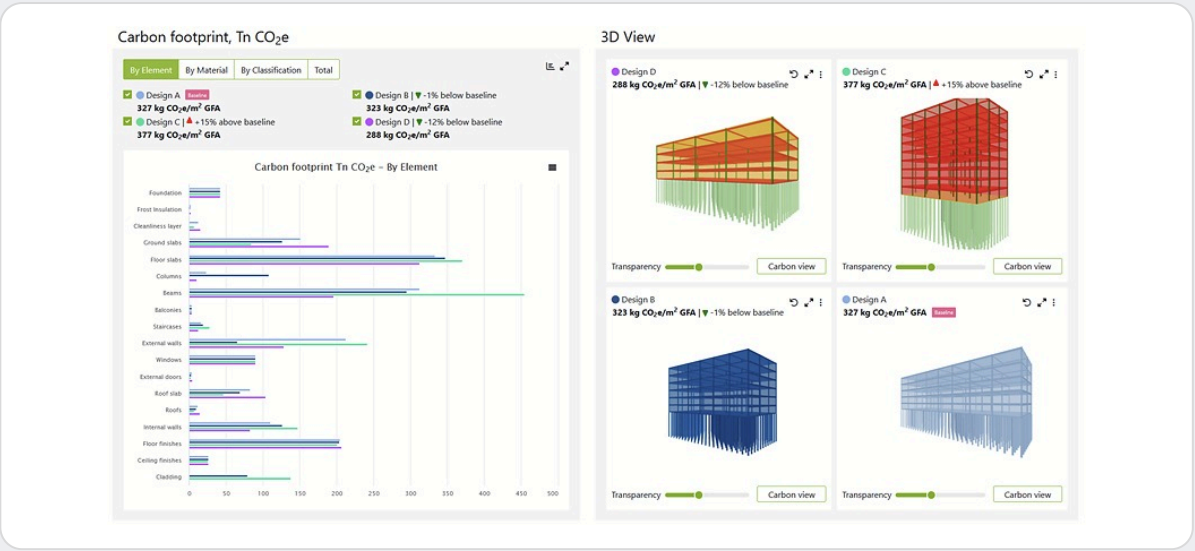
Software engineer and interim product owner on a browser-based early-design tool in which architects model building massing and see embodied carbon and energy results as they design.

Engineer	Features, integration and maintenance in TypeScript, React, Three.js, AG Grid, CI/CD
Product owner	Interim PO from H2 2025: specification, task breakdown and coordination for Free Drawing, alongside my own code
Domain	Methodology specifications for foundations and operational energy
Timeline	H2 2025 Free Drawing · Q1 2026 specifications · Q2 2026 Energy feature · Q3 2026 code quality
Outreach	Presented the product at Carbon Experts Summit 2025 and 2026

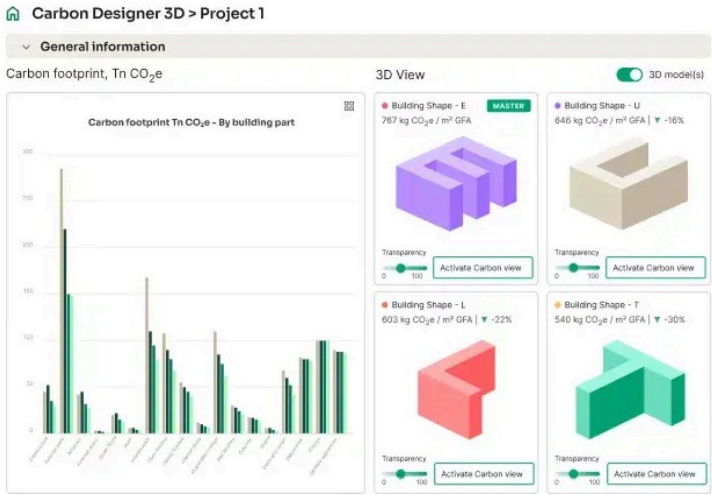
Computational design in the browser

Free Drawing turns an architect's sketch into massing, then into elements and quantities the carbon model can calculate: the geometry-to-analysis logic I build in Grasshopper, shipped to every user.

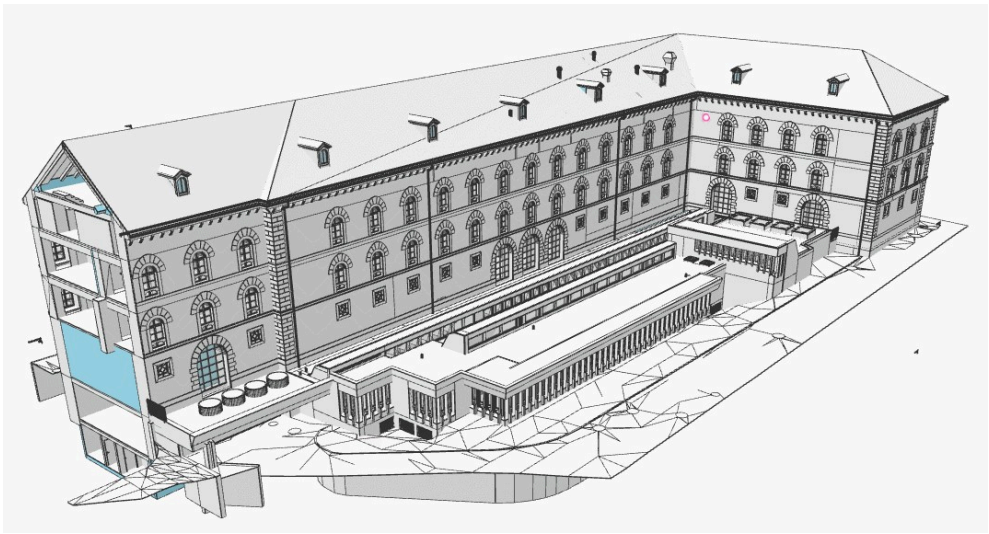
Sketch › Massing geometry › Building elements › Carbon and energy



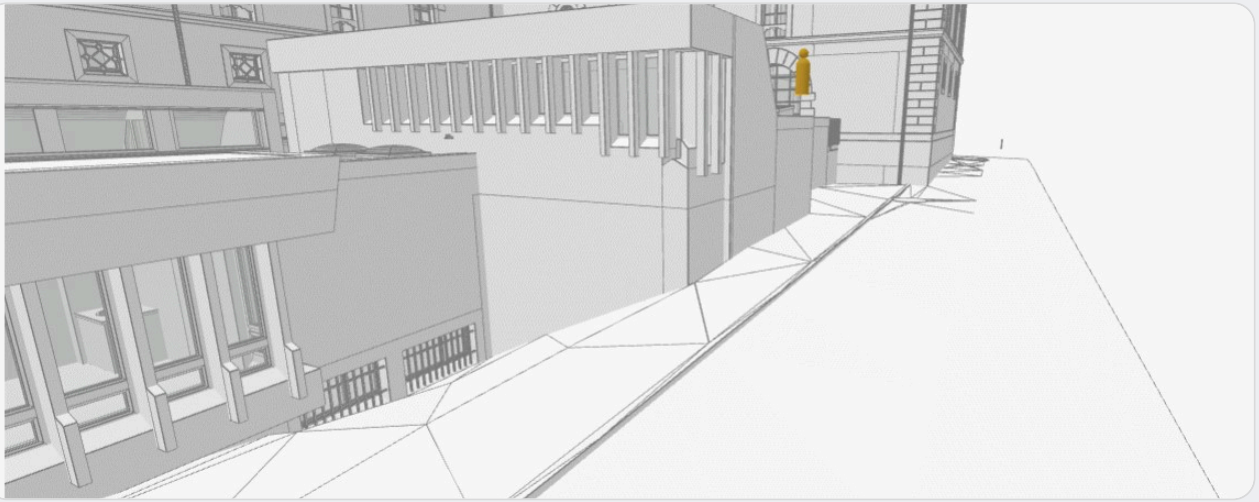
Design options compared: carbon by building element against the baseline, with each option in 3D.



Massing studies in Carbon Designer 3D: four building shapes and their embodied carbon per m² GFA.



The IFC model of a historical building complex in the viewer, with section, layers, points of interest and walk mode in the toolbar.



Walk mode at eye level, with the visitor marker, for exploring the building as a person would.

07 Client work · enxess, Liechtenstein · 2024

3D IFC web viewer for historical buildings

A browser viewer for IFC models of historical building objects, built for a client in Liechtenstein. Users orbit the model, cut sections, switch layers, add points of interest and walk through the site at eye level.

Stack	Three.js, React and TypeScript, with IFC parsing in the browser
Features	Orbit and walk modes, section cuts, layer control, points of interest, model reset
Role	Contract developer, end to end

Computational geometry for BIM data

Heavy, detailed IFC geometry has to load, cut and navigate smoothly in a browser: the same model-to-geometry work I do in Grasshopper with IFC, delivered as a web app anyone can open.

Revit / IFC › IFC geometry › Three.js › Browser viewer

LLM-assisted IFC data enrichment

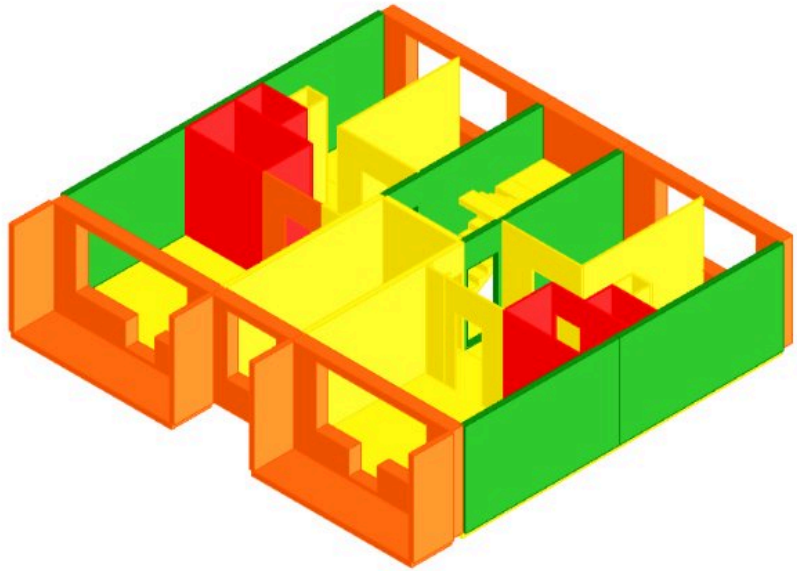
A workflow that uses a language model to fill in missing data in IFC models of Polish large-panel buildings, so prefabricated elements can be classified and assessed for reuse.

Published	Journal of Building Engineering 120 (2026), 115397, open access
Funding	Warsaw University of Technology research grant, as principal investigator
Code	github.com/Curiosit/circular-gh-ifc-ai, with example IFC and Revit models

Built in Grasshopper

The Grasshopper definition reads the IFC model, extracts element geometry and properties, sends them to the language model, and writes the enriched data back for analysis and visual checking.

Revit / IFC › Grasshopper › LLM API › Enriched IFC



Example output: building elements colour-coded by the enriched data for reuse assessment.

Parametric daylight regulation studies

My research tests how prescriptive building rules behave against climate-based simulation. The same 5 × 6 m room in Warsaw, analysed four ways: orientation and an opposite building decide most of the daylight, and a window-to-floor ratio rule sees neither.

What is compared

Poland's 1/8 window-to-floor ratio and sunlight-hour rules against EN 17037 and climate-based metrics (sDA, ASE, UDI), across parametric room, window and street variants.

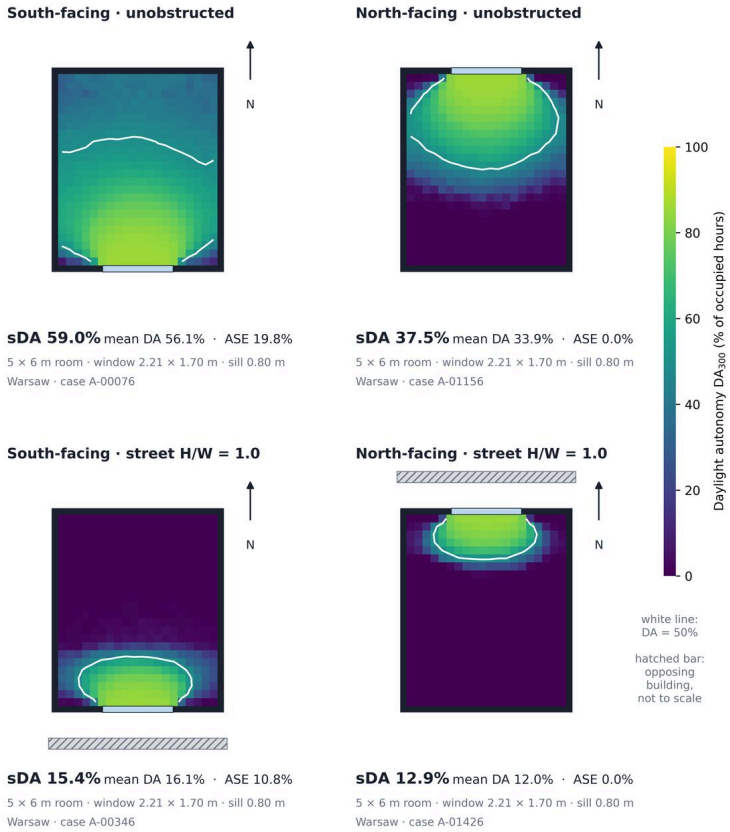
In practice

Daylight simulation studies at Bjerg Arkitektur, including projects with VELUX and Active House specialists.

Built in Grasshopper

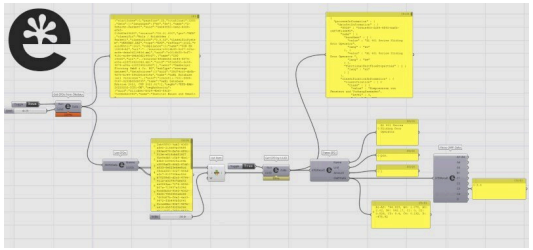
Rooms, windows and urban canyons are parametric definitions; Ladybug and Honeybee run the climate-based simulations in batch, with results exported for statistical analysis.

Grasshopper › Ladybug / Honeybee › Radiance › Python analysis

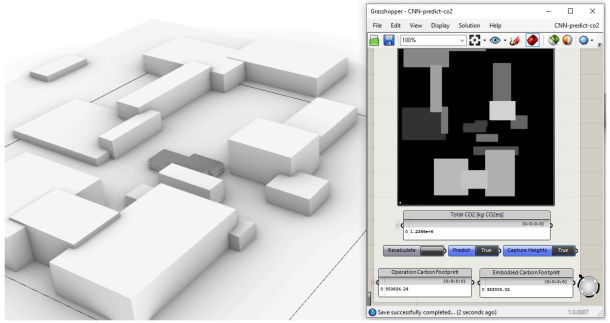


Climate-based simulation, daylight autonomy at 300 lx: south vs north, unobstructed vs a street with H/W = 1.0.

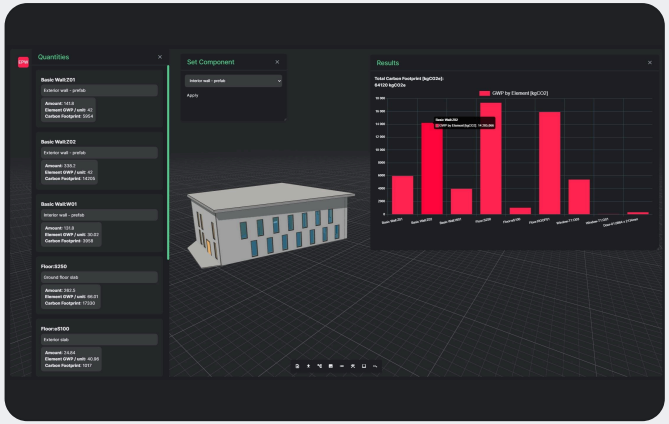
Open data and machine learning in Grasshopper



goeko.gha: Ökobaudat product data straight into a definition.



CNN-predict-co2: a height image of the massing in, carbon out, inside Grasshopper.



IFC carbon app (Energies 2024): quantities and carbon by element in the browser.

goeko.gha · 2023-2024

Open-source Grasshopper plugin in C# that reads environmental product declarations from Germany's public Ökobaudat database through its API. MIT licence, on Food4Rhino. The same logic also runs as a no-code web configurator on ShapeDiver.

PhD: CNN for carbon · 2017-2022

Parametric models generated the training variants; a convolutional network predicts operational and embodied carbon from massing, fast enough to drive optimization. Energies 2020, 130+ citations.

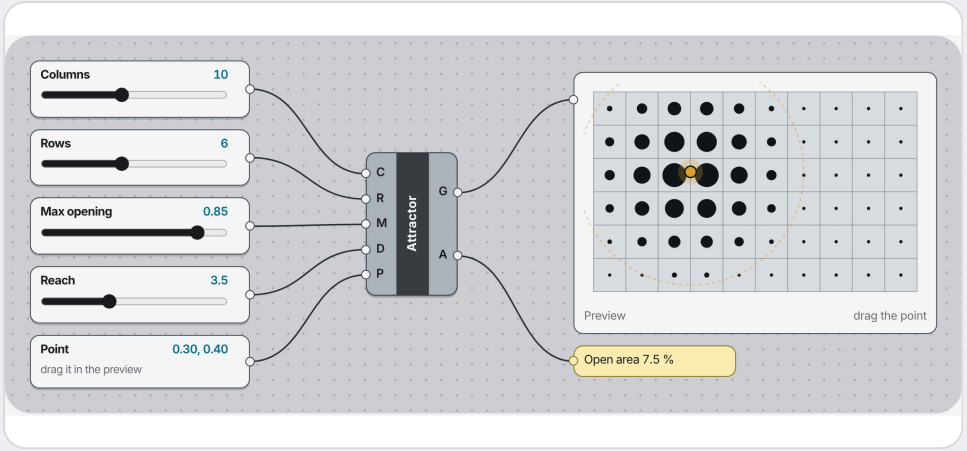
IFC carbon app · 2024

Machine learning, language models and BIM brought together for assessment tools (Energies 2024), with a browser prototype working directly on IFC models. Co-author of Poland's Whole Life Carbon Roadmap (2021).

Built in Grasshopper

Open data APIs › Grasshopper › Parametric variants › ML / CNN › Web and IFC tools

Grasshopper practice: teaching, tools and built work



Live parametric demo on my website: openings grow near an attractor point. Most automation projects start like this, then become a tool a team can run.

Seven years of teaching Grasshopper and ten years of practice behind it: I know how to build definitions that other people can read, run and maintain.

Teaching

Since 2017 at the Faculty of Architecture, Warsaw University of Technology: introduction to Grasshopper and multi-criteria optimization of energy, daylight and carbon. Since 2020 on the iBIM postgraduate programme: advanced Grasshopper, Rhino.Inside.Revit and Python / C# scripting. 400+ students.

Software and hackathons

2024-2026 software engineer and interim product owner on Carbon Designer 3D; 2024 IFC web viewer for a client in Liechtenstein. AEC Hackathons in Copenhagen and Zurich, including Best Mashup Project 2024. Rhino.Inside.Revit workflows taught to practising architects.

Architecture and built work

2014-2024 at Bjerg Arkitektur, Denmark and Poland, with in-house Grasshopper tools used on live projects. Parametric structures: the lookout tower and the Hatlehol Church roof (sections 02 and 03). Also a folded-steel public bench, 1st prize, with nine units installed in Horsens.