Generative design

Professor: Danil Nagy
Monday 11:00am-1:00pm
202 Fayerweather

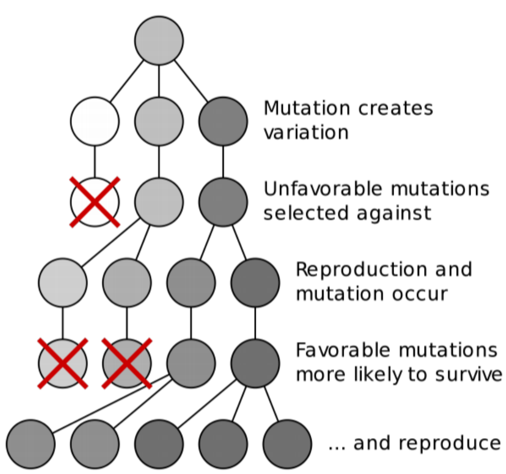
Look deep into nature, and then you will understand everything better.

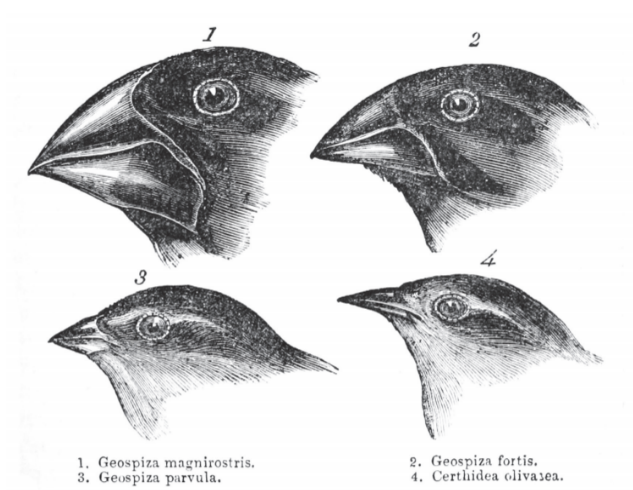
- *Albert Einstein*

I. Intelligent machines

In the past decade, our interaction with the world has been deeply affected by artificial intelligence. Many industries including finance, science, and manufacturing have been revolutionized by developments in Machine Learning, optimization, and other artificial intelligence technologies, which have allowed them to leverage the power of computing to solve complex problems in new and innovative ways.

Meanwhile, architectural design practice has been barely impacted by these developments. Although almost all designers use computers in their practice, the tools they rely on have not leveraged these emerging technologies. As a result, the design profession has not substantially evolved since computers were first introduced to the design world nearly four decades ago.

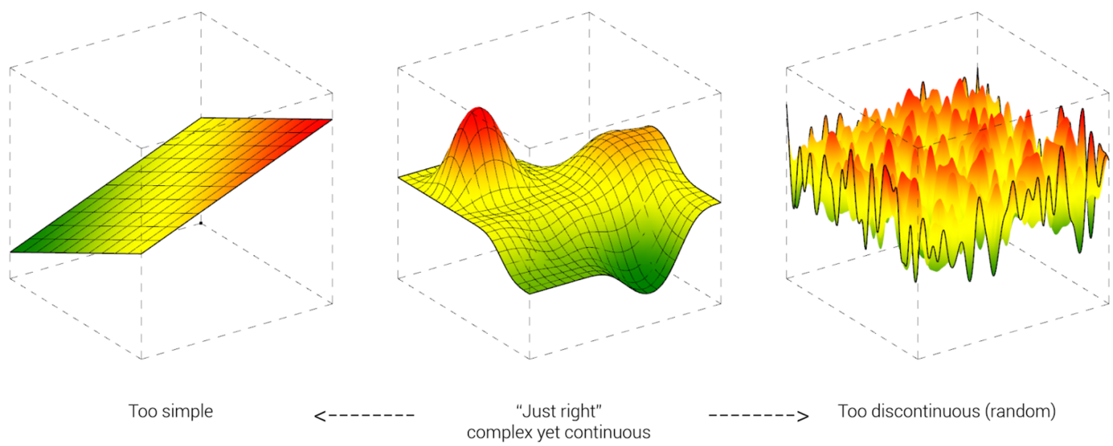




II. Learning from nature

Perhaps the greatest opportunity for artificial intelligence in design practice today is its ability to leverage another, much older form of intelligence - natural intelligence. Designers have always been inspired by the forms of nature, and their abilities to solve difficult problems in novel and beautiful ways. However, up to this point our inspiration from nature has been limited to ‘bio-mimicry’, or the reproduction of nature’s physical forms in new designs. Can we go a step further and actually design like nature?

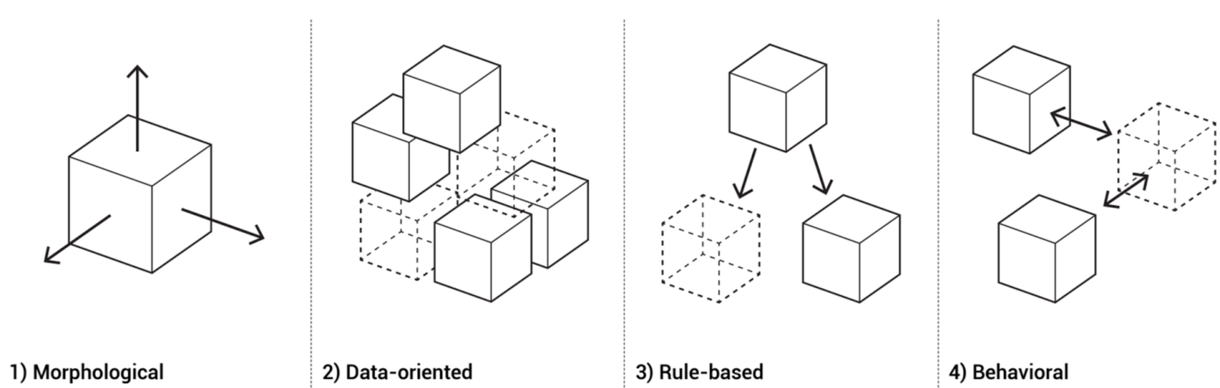
To do this we have to first understand how nature designs. The basic element of nature's design is the species, a kind of *model* which encodes all of the unique properties and abilities of its individual members. The basic tool of nature's design is evolution, which is an *iterative process* by which species are able to adapt and improve based on interaction with other species and their environment.



III. <u>Automating design</u>

This class will explore how we can use new technology to leverage nature's design methods to create new design workflows:

1. Instead of designing objects, we will learn to *design systems* which encode the full range of possibilities of a particular design concept
2. We will then learn methods for *measuring and quantifying the performance* of these systems so that each design can be evaluated automatically by the computer
3. Finally, we will create *automated evolutionary processes* which will allow the computer to search through our design systems to find novel and high-performing designs.

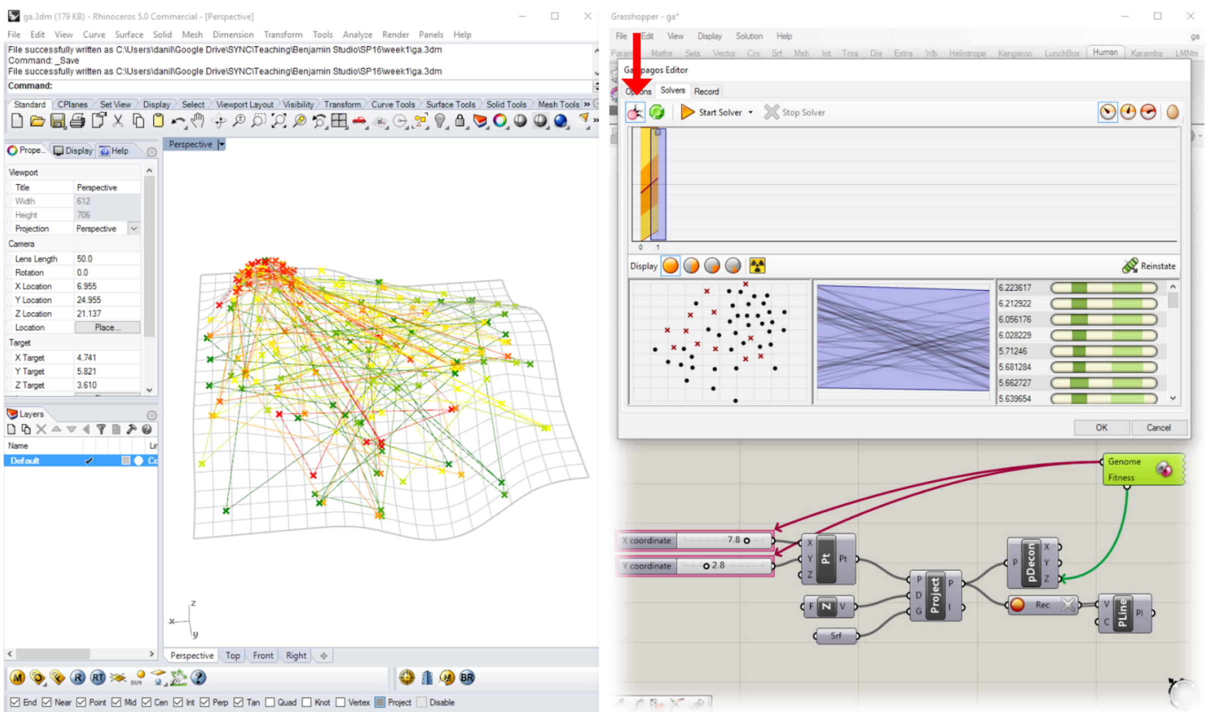


IV. From tools to partners

These new workflows will allow us to explore a much wider space of design than possible through traditional intuitive methods, and lead not only to the discovery of novel and unexpected solutions, but to a deeper understanding of the design problem itself.

To take advantage of these possibilities, we will have to learn how to work with computers in new ways. Instead of thinking of computers as *tools* that accomplish specific tasks in predictable ways, we will think of computers and algorithms as *partners* in our design process.

We will discover that orchestrating such a human/machine design collaboration is actually quite difficult. Artificial and human intelligences work in very different ways, and in order to work together we will have to be much more explicit in how we describe our design concepts and intentions to the computer. However, if we succeed, this interaction will not only create new opportunities for design, but will make us more thoughtful, more responsible, and better human designers.



V. Technology

The course will teach students new workflows for generating, evaluating, and evolving physical designs using custom *Python* scripts running on top of the *Grasshopper* environment for *Rhino*. Prior knowledge of computer programming in Python is encouraged but not required. A working understanding of Rhino and Grasshopper are a prerequisite for taking the course. For basic training in Grasshopper students are encouraged to work through the relevant tutorials on the http://skilltree.gsapp.org/ prior to the first day of class.

VI. Schedule

Session 1 - Design computation

The first session will introduce students to the basic concepts of generative design and teach them how to create complex models that can be controlled and evaluated by an automated search algorithm. The Python programming language will be introduced as a way to amplify the generative complexity of parametric models in Grasshopper. This session will also cover techniques for evaluating designs including using third-party Grasshopper plugins for structural and environmental analysis.

Jan 22 Introduction to Generative Design - http://bit.ly/sp18-gd-lecture-intro

- Course notes:
  - https://medium.com/generative-design/generative-design-introduction-64fb2db38e1
  - https://medium.com/generative-design/learning-from-nature-fe5b7290e3de
  - https://medium.com/generative-design/step-1-generate-6bf73fb3a004
  - https://medium.com/generative-design/designing-measures-2c66a71b2ff3
- Readings:
  - Danil Nagy, et al. - *Project Discover: An application of generative design for architectural space planning*, SimAUD (2017) [pdf]

Jan 29 Introduction to computational design

- Course notes:
  - https://medium.com/generative-design/introduction-to-computational-design-6c0fdfb3f1
  - https://medium.com/generative-design/computational-design-in-grasshopper-1a0b62963690
- Additional grasshopper resources:
  - Intro to Grasshopper: https://www.youtube.com/playlist?list=PLpuejoPydMLW9y_EKRhNECkyD_gfCf9nb
  - Modeling geometry: https://www.youtube.com/playlist?list=PLpuejoPydMLVxIbi61j0D6stCIYMPZw0z
  - Data structures: https://www.youtube.com/playlist?list=PLpuejoPydMLUsEeTEMgNxGbGTZTWca76e

Feb 5 Control strategies I

- Demo model: https://www.dropbox.com/s/9leyzsm0upe3puh/demo_bridge.gh?dl=0
- Due: Group definitions and initial project ideas
- Course notes:
  - https://medium.com/generative-design/using-python-in-grasshopper-77bfca86e84b
  - https://medium.com/generative-design/fundamentals-of-python-variables-b0523dd698a7
  - https://medium.com/generative-design/fundamentals-of-python-conditionals-and-loops-77ab7cbf4038
  - https://medium.com/generative-design/fundamentals-of-python-functions-and-objects-bf8a4953068a
  - https://medium.com/generative-design/working-with-geometry-in-python-a256de7bb1b1

Feb 12 Control strategies II

- Course notes:
  - https://medium.com/generative-design/control-strategies-b4cf07b26cda
  - https://medium.com/generative-design/recursive-systems-5b1f813b2b8b
  - Behavioral systems (Cellular automata) - TBD
- Readings:
  - Danil Nagy, Dale Zhao, David Benjamin - *Nature-Based Hybrid Computational Geometry System for Optimizing Component Structure*, Design Modelling Symposium (2017) [pdf]

Feb 19 Metrics I

- Course notes:
  - https://medium.com/generative-design/structural-analysis-with-karamba-a73b959587c0
  - https://medium.com/generative-design/solar-analysis-in-grasshopper-5dae76c9b6cb

Feb 26 Metrics II

- Course notes:
  - https://medium.com/generative-design/view-analysis-with-isovist-587fce149956
  - Graph-based methods - TBD
- Readings:
  - Danil Nagy, Lorenzo Villaggi, Jim Stoddart, David Benjamin - *The Buzz Metric: A Graph based Method for Quantifying Productive Congestion in Generative Space Planning for Architecture*, Technology | Architecture + Design (Taylor & Francis, 2017) [pdf]

March 5 Design space analysis and visualization

- Readings:
  - Danil Nagy, Lorenzo Villaggi, Dale Zhao, David Benjamin. - *Beyond Heuristics*, ACADIA (2017) [pdf]

March 12 SPRING BREAK - NO CLASS

Session 2 - Design evolution

The second session will dive deeper into the generative design workflow, and focus primarily on the automated search engine itself. Students will learn how to use state-of-the-art genetic algorithms to automatically search through their design models for high-performing solutions, and how to evaluate the search process to derive new knowledge about their design.

March 19 Introduction to design optimization, elements of genetic algorithms

- Due: Midterm assignments (design space model report)
- Course notes:
  - https://medium.com/generative-design/design-optimization-2ec2ba3b40f7
  - https://medium.com/generative-design/evolving-design-b0941a17b759

March 26 Setting up an optimization

- Discover documentation:
  - https://github.com/danilnagy/discover/blob/master/README.md

April 2 Input types, single vs. multiple objective optimization

April 9 Series and permutations, handling constraints

April 16 Optimization visualization

Friday, May 4 Final assignments (report and animation) due

VII. Links

- http://bit.ly/sp18-gd-syllabus - this page
- https://github.com/danilnagy/discover - Github page of Discover/Explore tool
- https://medium.com/generative-design - collection of class notes and tutorials
- http://bit.ly/sp18-gd-tools - Discover and all dependencies for use on school computers

VIII. Additional Readings

Articles, papers, and book chapters:

1. Alan Turing - Computing Machinery and Intelligence (1950) [pdf]
2. Christopher Alexander - Notes on the Synthesis of Form, *Introduction* (1964) [pdf]
3. Nicholas Negroponte - Computational Design Thinking, *Towards a Humanism Through Machines* (1969) [pdf]
4. William J. Mitchell - Computer-Aided Architectural Design, *Chapter 2: The Computer's Role in Design* (1977) [pdf]
5. Kevin Kelly - Out of Control, *Chapter 15: Artificial Evolution* (1994) [pdf]
6. John Frazer - An Evolutionary Architecture, *Introduction* (1995) [pdf]
7. H. Lipson and J. B. Pollack - *Automatic design and Manufacture of Robotic Lifeforms*, Nature (2000) [pdf]
8. Peter J. Bentley and David W. Corne - Creative Evolutionary Systems, *An Introduction to Creative Evolutionary Systems* (2002) [pdf]
9. A. Konak, D. W. Coit, A. E. Smith - *Multi-Objective Optimization Using Genetic Algorithms: A Tutorial* (2006) [pdf]
10. Ian Keough and David Benjamin - *Multi-objective Optimization in Architectural Design*, SimAUD (2010) [pdf]
11. Daniel Shiffman - The Nature of Code, *Chapter 9: The Evolution of Code* (2012) [link]
12. David Benjamin, Danil Nagy and Carlos Olguin - *Growing Details*, AD (2014) [pdf]
13. Kostas Terzidis - Permutation design, *Introduction* (2015) [pdf]

Books

14. James C. Spall - Introduction to Stochastic Search and Optimization (2003) [Amazon]
15. J. J. Schneider and S. Kirkpatrick - Stochastic Optimization (2006) [pdf]
16. Xin-She Yang - Nature-Inspired Metaheuristic Algorithms (2nd edition) (2008) [pdf]
17. Stuart Russell and Peter Norvig - Artificial Intelligence, A Modern Approach (3rd edition) (2010) [pdf]

18. Achim Menges and Sean Ahlquist, ed. - Computational Design Thinking (2011) [Amazon]
19. A.E. Eiben and J.E. Smith - Introduction to Evolutionary Computing (2nd edition) (2015) [pdf]

IX. Semester project

The main deliverable for the class is a design project which students will develop either individually or in groups of two during the semester.

- Brief

    Choose a design problem which you want to solve using the tools of generative design. The scale of the problem is up to you, and can relate to any field of design including but not limited to industrial, architecture, or urban design. For example, you might want to optimize the design of a chair, a room, a building, or an entire city block. You can use your studio project or another current or past project as a starting point, but I would recommend that you recalibrate the scope of your design problem specifically for this class. This will allow you to fully explore the generative design methods without being burdened by too many unnecessary factors.

- Components

    1. Design space model - create a model that parametrizes the design problem and defines all possible solutions that can be searched by the genetic algorithm. You should be clear in your choice of parameters, and develop a good intuition for how your model navigates the tradeoffs of bias/variance and complexity/continuity.
    2. Design metrics - define the objectives and constraints of your model. You should be clear about how these measures relate to the requirements of the design problem, how you value your design, and how you communicate these values to the search algorithm. You can use whatever metrics relate to your design problem, but *must include at least one of the simulation methods studied in class* (FE structural analysis, ray-based solar analysis, isovist-based view analysis, graph-based flow analysis).
    3. Design evolution - using the tools covered in class, run your model through a series of optimization 'experiments' to derive novel and high-performing solutions to your design problem. You should be clear about how you are specifying the algorithm's parameters before each experiment, how you are analyzing and learning from the results, and how you are adjusting the process each time based on what you have learned.

Deliverables

Design Space Model report - Due Monday, March 19

- Document your design space and system of measures as a unique design strategy for solving your chosen design problem. Your document should include a description of the following aspects of your design:
    1. What are the dimensions (parameters) of your design space and how does each parameter affect the design?
    2. What variable types and control strategies are you using for your parameters, and how does this relate to the complexity and continuity of your design space?

3. What are the boundaries of your design space and how does this relate to the bias vs. variance tradeoff of your design space model?
4. What evaluation metrics are you using and how do they relate to your design goals?
5. What are the “intuitive” solutions in your design space?
6. Do you think your design space is searchable?

- A template** of typical design space visualization strategies will be provided as a starting point for your report following this general outline:
    1. Overview of design problem
    2. Description and diagrams of system for generating design solutions including parameterization
    3. Description and diagrams of each metric used to evaluate potential design solutions and how it is computed
    4. Examples of good and bad designs for each metric
    5. Description of potential conflicts or tradeoffs between metrics
    6. Design space model overview diagram
    7. Parameter study diagram showing the geometric expression of individual inputs
    8. Response surface analysis showing relationship of inputs to outputs, internal structure, and boundaries of the design space
    9. Collection of ‘intuitive’ designs sampled manually from the design space

Final report and animation - Due Friday, May 4

- For the final deliverable you will update your midterm report with any changes or developments to the designs space model, and add to it documentation of the optimization process. This documentation should include a description of the following aspects of the optimization process:
    1.
- An updated template** including typical optimization visualization strategies will be provided following this outline:
    1. Updated design space overview model showing the exact parameter types (continuous, categorical, series, or sequence) and output types (objective min/max or constraint to be met)
    2. *time vs. objective* - is the algorithm learning to pick designs with higher performance over time? (ID or generation along X-axis, objective value along Y-axis, color, and/or size)
    3. *time vs. input* - is the algorithm narrowing in on particular strategies over time? (ID or generation along X-axis, input value along Y-axis, color, and/or size)
    4. *objective vs. objective* - is there a tradeoff in your objectives that is creating a Pareto front of optimal designs. Is the algorithm able to target the front more and more over time? (objective(s) along X and Y-axes, ID or generation along color)
    5. *input vs. objective* - are there particular input values that tend to create higher performing designs? (input(s) along X and Y-axes, objective(s) along color and size)
    6. Scatter plot
    7. Final outcomes (selection of high-performing designs or description of a novel set of strategies discovered through the process)
- A final animation showing the process of optimization over time. The animations can visualize each design in the order it was generated, or you can create composites that collapse all the designs in a single generation into a single image. Then you can either cycle through generations in an animation, or create a grid of images that show the whole optimization all at once. All videos should meet the following criteria:

1. between 0:30 and 1:00 long
2. HD widescreen resolution at least 1280x720
3. .mp4 video format

** You should use these templates as starting points and produce the included diagrams for your project as a minimum requirement. However, since the Generative Design process is still a relatively new methodology, there are few rules of thumb for the types of graphics or diagrams you should produce. Thus you are also encouraged to invent and develop your own graphic strategies that can best visually communicate your particular project and the generative design workflow. You should use this project as an opportunity to not only hone your own graphic style as a designer, but experiment and speculate on new representational methods that relate specifically to the generative design methodology. For reference, a collection of sample graphics can be found here:

http://bit.ly/sp17-gd-examples