## Introduction to Computational Design Part 3: Summary and Discussion

Presenter: Yuki Koyama (AIST, Japan)

2021-05-12 | CHI 2021 Courses | Online





**Design parameter tweaking** 

**Computational design:** a paradigm in which design problems are formulated as optimization problems and solved by computational techniques.



## **A Definition of "Computational Design"**

### **A general definition:**

### A more focused (narrower) definition:

Computational design is a paradigm in which design problems are formulated as optimization problems and solved by computational techniques.

### Computational design is a paradigm in which design problems are formulated *mathematically* and solved by computational techniques.



## **Computational Design Research**

### Approach:

- Formulate design processes that have been traditionally dependent on individual skills as mathematical optimization problems, and
- Support or augment design processes by devising new ways of utilizing computing power and mathematical tools

### Goal:

 Enable efficient design workflow or sophisticated design outcomes that are impossible in traditional approaches relying purely on the human brain



### **Computational design:**

### SIGGRAPH 2014







### **Fly-ability**

### Haptics

### [Recap]

Pacific Graphics 2016



### SIGGRAPH Asia 2015

VRST 2017



**Fly-ability** 

**Connect-ability** 

Haptics

The design goal (objective) is the functionality of the designed object





## Design Goals (Objectives)

## Functionality

We can compute the "goodness" of a design by predictive simulation

### Aesthetic preference

We cannot compute the "goodness" of a design as it is perceptual

Human assessment is necessary





## **A Systematic Approach to Input Human Assessment**

Human Computation: "[...] *a paradigm for* utilizing human processing power to solve problems that *computers cannot yet* solve." [von Ahn 05]



A. J. Quinn and B. B. Bederson. 2011. Human computation: a survey and taxonomy of a growing field. In Proc. CHI '11. 1403-1412. von Ahn, L. Human Computation. Doctoral Thesis. UMI Order Number: AAI3205378, CMU, (2005).



### Either crowd workers or the single user



## Human-in-the-Loop Optimization





Human-in-the-loop optimization is used for solving problems with "subjective" objective functions (e.g., preference)

## Summary

- Computational design is a paradigm in which design problems are formulated as mathematical optimization and solved using computational techniques
  - HCI topics: Functional fabrication, haptics, user interface, visual design, etc.
- Human-in-the-loop optimization can be used for solving problems with perceptual objective functions (e.g., preference)
  - Tight integration between algorithm design and interface design would be the key to achieve higher efficiency



9

## Discussion

## **Discussion: Exploratory Design**

"Exploratory modeling is open-ended: the user begins the design process with an under-specified goal, and the precise form of the final model is established through experimentation [...]"

Jerry O. Talton, Daniel Gibson, Lingfeng Yang, Pat Hanrahan, and Vladlen Koltun. 2009. Exploratory modeling with collaborative design spaces. ACM Trans. Graph. 28, 5 (December 2009), 1–10. DOI:<u>https://doi.org/10.1145/1618452.1618513</u>



## **Discussion: Exploratory Design**

"Exploratory modeling is open-ended: the user begins the design process with an under-specified goal, and the precise form of the final model is established through experimentation [...]"

# $\mathbf{x}^* = \arg \max f(\mathbf{x})$ $\mathbf{x} \in \mathcal{X}$

Jerry O. Talton, Daniel Gibson, Lingfeng Yang, Pat Hanrahan, and Vladlen Koltun. 2009. Exploratory modeling with collaborative design spaces. ACM Trans. Graph. 28, 5 (December 2009), 1–10. DOI:<u>https://doi.org/10.1145/1618452.1618513</u>

The objective function is unknown at the beginning, and it can even change during the design process

Adaptive techniques (e.g., online machine learning) should be used



## **Discussion: "Optimization-in-the-Loop" Design Iteration**



13

## **Discussion: "Optimization-in-the-Loop" Design Iteration**



### Design iteration

## The designer decides what to do next



### Manual content edit

## Automatic optimization



## **Discussion: "Optimization-in-the-Loop" Design Iteration**

### Sketchplore (DIS 2016)

### MenuOptimizer (UIST 2013)

**Tool/UI Design** 

DIS 2016, June 4–8, 2016, Brisbane, Australia

### Sketchplore: Sketch and Explore with a Layout Optimiser

Daryl Weir<sup>1</sup>

daryl.weir@aalto.fi

Kashyap Todi<sup>1,2</sup> kashyap.todi@uhasselt.be <sup>1</sup>Aalto University

Antti Oulasvirta<sup>1</sup> antti.oulasvirta@aalto.fi <sup>2</sup>Hasselt University - tUL - iMinds



Figure 1: Sketchplorer is an interactive layout sketching tool supported by real-time model-based optimisation. The tool is designed to facilitate the creative and problem-solving aspects of sketching without requiring extensive input. While a designer is sketching, a design task is automatically inferred. The optimiser uses predictive models to make suggestions for local and global changes that improve usability and aesthetics. Suggestions appear on the side, and never override the designer's work.

### ABSTRACT

This paper studies a novel concept for integrating real-time design optimisation to a sketching tool. Although optimisation methods can attack very complex design problems, their insistence on precise objectives and a point optimum is a poor fit with sketching practices. Sketchplorer is a multitouch sketching tool that uses a real-time layout optimiser. It automatically infers the designer's task to search for both local improvements to the current design and global (radical) alternatives. Using predictive models of sensorimotor performance and perception, these suggestions steer the designer toward more usable and aesthetic layouts without overriding the designer or demanding extensive input.

### Author Keywords

Sketching; Model-based optimisation; Visual Layouts

### **ACM Classification Keywords**

H.5.2. Information Interfaces and Presentation: User Interfaces

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed ercial advantage and that copies bear this notice and the full cita for profit or con tion on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or re-publish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org. DIS 2016, June 04-08, 2016, Brisbane, QLD, Australia.

Copyright is held by the owner/author(s). Publication rights licensed to ACM.

### INTRODUCTION

This paper is motivated by the observation that optimisation methods have great untapped potential in design tools. We focus on the activity of sketching layouts, in which a designer places, colours, organises, and defines elements on a canvas. From a combinatorial perspective, the design of layouts is notoriously hard. For a canvas of  $1024 \times 768$  pixels, divided into a  $24 \times 32$  grid, as in the tool presented here, there are 158,400 one-element layouts and a whopping 1041 eight-element layouts. Although algorithms may not be able to find the option timal solution in such large search spaces, they can "parallelise" search, and find candidate solutions and suggest them to designers. This could help designers in exploration, who are known to be limited to a handful of designs per iteration [8]. Also, algorithms can complement designers by exploring design spaces neutrally without being constrained by past experiences, to produce designs that the designer might not otherwise conceive. Employing an optimiser might also improve the quality of designs for end-users (see [12, 33, 45]).

However, several hard research challenges emerge. First, layout design is a complex, multi-objective task addressing not only usability but also aesthetic qualities [15, 44]. Presently no algorithmic approach exists that can address both. Second, optimisation typically takes a long time, due to combinatorial complexity, and no solution has been shown for fastpaced, iterative design of layouts. Third, although optimisation methods can attack very complex design problems, their

### GUI

### MenuOptimizer: Interactive Optimization of Menu Systems

**Gilles Bailly**<sup>1</sup> Antti Oulasvirta<sup>1</sup> Timo Kötzing<sup>2</sup> <sup>1</sup>Max Planck Institute for Informatics and Saarland University



Figure 1: MenuOptimizer assists in the design of menus: While the designer edits the menu (action in red), a model-based optimizer updates itself to provide feedback and suggestions (in blue): A) Item feedback indicates the frequency (line width) and user performance over time (color gradient). B-C) Hotkeys and separators are automatically assigned. D) Item placements to improve user performance are suggested. E) Designers can normally edit items (move, delete, etc.) and also F) lock items to constrain them together to accelerate optimization. design strongly affects their usability. However, despite

### ABSTRACT

Menu systems are challenging to design because design spaces are immense, and several human factors affect user behavior. This paper contributes to the design of menus with the goal of interactively assisting designers with an optimizer in the loop. To reach this goal, 1) we extend a predictive model of user performance to account for expectations as to item groupings; 2) we adapt an ant colony optimizer that has been proven efficient for this class of problems; and 3) we present MenuOptimizer, a set of interactions integrated into a real interface design tool (QtDesigner). MenuOptimizer supports designers' abilities to cope with uncertainty and recognize good solutions. It allows designers to delegate combinatorial problems to the optimizer, which should solve them quickly enough without disrupting the design process. We show evidence that satisfactory menu designs can be produced for complex problems in minutes.

### Author Keywords

Menus; Predictive models; Interactive optimization.

ACM Classification Keywords H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

### INTRODUCTION

Menu systems, consisting of menus, hotkeys or toolbars, are widespread interfaces for selecting commands. Interface

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full cita on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissi JIST'13, October 8-11, 2013, St. Andrews, United Kingdom

### DesignScape (CHI 2015)

UIST'13, October 8–11, 2013, St. Andrews, UK

Sabrina Hoppe<sup>1</sup> <sup>2</sup>University of Jena

apparent simplicity, designing usable menu systems is challenging because the number of alternative designs grows superexponentially as a function of the number of commands. For instance, a linear menu with *n* items can, in theory, be organized in n! ways. However, professional applications comprise hundreds of items organized in hierarchical menus. A menu hierarchy can be organized in about (2n)! ways. For 50 items, the size of the search space is a whopping  $100! \approx 10^{158}$ . Design heuristics, such as placing frequently used items at the top [5], may be effective for small *n* but fail with larger *n* or if additional human factors such as semantic relationships among items are considered. Although experts can quickly generate a handful of solutions to hard design problems [6], they cannot examine all promising solutions. Novices, known to search the space depth-first [3], are likely to get stuck in a local search space.

Combinatorial optimization methods (e.g., [29]) have been successfully used to generate user interfaces such as virtual keyboards [8,21,24,34]. These methods explore a large number of designs in order to find ones that minimize or maximize a pre-specified objective function. Computation time is on a scale of hours, days, or weeks. While empirical evidence confirms improvements in usability in other contexts (e.g., [24,34]), there is reason to suggest that they may be impractical for the design of menu systems. First, designers cannot be expected to wait days or weeks for a solution. Moreover, designers may not be able to define the optimization problem completely in advance. Interaction design, in general, is rather more an iterative process of redefinition and refinement. Finally, predictive models for menu systems performance are only just emerging [5,20,25], are limited to linear menus, and have yet to cover all important human factors that affect design choices.

Supporting Creativity through UX Design

CHI 2015, Crossings, Seoul, Korea

### **DesignScape: Design with Interactive Layout Suggestions**

Peter O'Donovan<sup>1,2</sup> <sup>1</sup>Dept. of Computer Science University of Toronto odonovan@dgp.toronto.edu

Aseem Agarwala<sup>2</sup> <sup>2</sup>Adobe Research Seattle, WA, USA asagarwa@adobe.com

Aaron Hertzmann<sup>1,</sup> <sup>3</sup>Adobe Research San Francisco, CA, USA hertzman@adobe.com

### ABSTRACT

Creating graphic designs can be challenging for novice users. This paper presents DesignScape, a system which aids the design process by making interactive layout suggestions, i.e., changes in the position, scale, and alignment of elements. The system uses two distinct but complementary types of suggestions: refinement suggestions, which improve the current layout, and brainstorming suggestions, which change the style. We investigate two interfaces for interacting with suggestions. First, we develop a suggestive interface, where suggestions are previewed and can be accepted. Second, we develop an adaptive interface where elements move automatically to improve the layout. We compare both interfaces with a baseline without suggestions, and show that for novice designers, both interfaces produce significantly better layouts, as evaluated by other novices.

### Author Keywords

Graphic design, suggestion interfaces, adaptive design

### ACM Classification Keywords

D.2.2 Software Engineering: Design Tools and Techniques User Interfaces

### INTRODUCTION

Graphic design is ubiquitous in modern life. Unfortunately, creating designs can be difficult, particularly for novices, who often wish to create simple posters, cards, or social media designs. Starting from a blank canvas can be overwhelming, and exploring alternatives is time-consuming. Novice designers also make a variety of mistakes, from misalignment to incorrect emphasis of elements. Existing tools range from simple template-based interfaces like PowerPoint, to complex systems like Illustrator. However, these tools provide no suggestions when modifying templates or designs.

This paper presents a novel system for graphic design using layout suggestions, i.e., changes in the size, position, and alignment of elements. Our system proposes two complementary types of suggestions: refinements which improve the current layout, and brainstorming suggestions which explore alternative layouts with large changes in style (see Fig. 1). Exploration and refinement are critical and complementary

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or re-publish, to post on servers or to redistribute to lists, requires prior specific permission sions from Permissions@acm.org and/or a fee. Request permissions from Permissions@acm CHI 2015, April 18 - 23, 2015, Seoul, Republic of Korea



Figure 1. DesignScape Interface. The central canvas allows the user to create layouts in a simple editor. On the left, the system provides refine ment suggestions, layouts similar to the canvas, but slightly improved On the right, the system provides *brainstorming suggestions* large-scale layout changes in a variety of styles. Photos courtesy of Wilhelm Joys Andersen and Martin Fisch.

tasks in design. However, exploration is difficult since a designer must imagine possible layouts, and modify many elements. Refinement is also difficult, since a single modification can necessitate many other changes. Our system includes both types, allowing users to easily switch between exploring alternative layouts and refining the current layout.

We use an energy-based model to generate designs that encode design principles such as symmetry, alignment, and overlap. User constraints are used to infer the designer's intent, and to make refinement suggestions on the current layout. We also learn a "style space" from examples, which can be used to generate new layouts in a variety of styles, providing starting points for design. The system can also retarget layouts, allowing the user to easily modify the design size.

We also investigate different ways users can interact with su gestions. First, we develop a suggestive interface, where suggestions are previewed and accepted. Second, we develop an adaptive interface which moves elements automatically. The two modes are compared to a baseline without suggestions by novice users on Mechanical Turk, and the quality of the resulting layouts are also evaluated. Both modes produce significantly better designs than the baseline on average. Lastly, we demonstrate the system's use for tablet-based design.

### RELATED WORK

Exploring alternatives is a vital part of the design process. Gross and Do [2] present a prototyping interface which allows users to sketch drawings and store alternatives. Terry et al. [7] present an interaction technique which allows users to save and embed alternatives during the design process, and easily manipulate alternatives at a later point. Dow et al. [1] find that forcing users to create multiple design alternatives, instead refining a single design leads to improved results. Lea



## **Discussion: UX of Human-in-the-Loop Optimization**



Yijun Zhou, Yuki Koyama, Masataka Goto, and Takeo Igarashi. 2021. Interactive Exploration-Exploitation Balancing for Generative Melody Composition. In Proceedings of the 26th International Conference on Intelligent User Interfaces (IUI '21), pp.43–47. DOI:<u>https://doi.org/10.1145/3397481.3450663</u>

### [Zhou+, IUI 2021]

This paper investigates how the user feels when the variation of the candidates (exploration vs. exploitation) can be manually controlled

C.f., Human-Al collaboration



16

## C++ Implementation & Python Bindings Available on GitHub

- https://github.com/yuki-koyama/mathtoolbox
  - Bayesian optimization (BO)
- https://github.com/yuki-koyama/sequential-line-search
  - Preferential Bayesian optimization (PBO)
  - Sequential line search
- https://github.com/yuki-koyama/sequential-gallery
  - Sequential gallery



The core algorithm is implemented in include/sequential-line-search/\*.hpp and src/\*.cpp. This repository also contains the following example demos

 $a(\cdot)$ 

- **bayesian\_optimization\_1d**: A simple demo of the standard Bayesian optimization applied to a one-dimensional test function.
- **sequential\_line\_search\_nd**: A simple demo of the sequential line search https://github.com/yuki-koyama/sequential-line-search/blob/master/docs/concept.jpg



## Questions / Comments



### (URL REMOVED)





### **Acknowledgment:** I thank all the collaborators!

- Nobuyuki Umetani
- Ryan Schmidt •
- Takeo Igarashi
- Morihiro Nakamura
- Masa Ogata •
- Eisuke Fujinawa •

- Shigeo Yoshida
- Takuji Narumi
- Tomohiro Tanikawa
- Michitaka Hirose
- Shinjiro Sueda
- Emma Steinhardt •

- Ariel Shamir
- Wojciech Matusik
- Daisuke Sakamoto •
- Issei Sato
- Masataka Goto •



## Introduction to Computational Design

Presenter: Yuki Koyama (AIST, Japan)

2021-05-12 | CHI 2021 Courses | Online

