Forte: User-Driven Generative Design

Xiang 'Anthony' Chen[†], Ye Tao[¶], Guanyun Wang[†], Runchang Kang[†]
Tovi Grossman[§], Stelian Coros[‡], Scott E. Hudson[†]
[†]Carnegie Mellon University [¶]Zhejiang University [‡]ETH Zurich [§]Autodesk Research
[†]{xiangche, scott.hudson}@cs.cmu.edu {guanyunw, runchank}@andrew.cmu.edu
[¶]taoye@zju.edu.cn [‡]scoros@inf.ethz.ch [§]tovi.grossman@autodesk.com

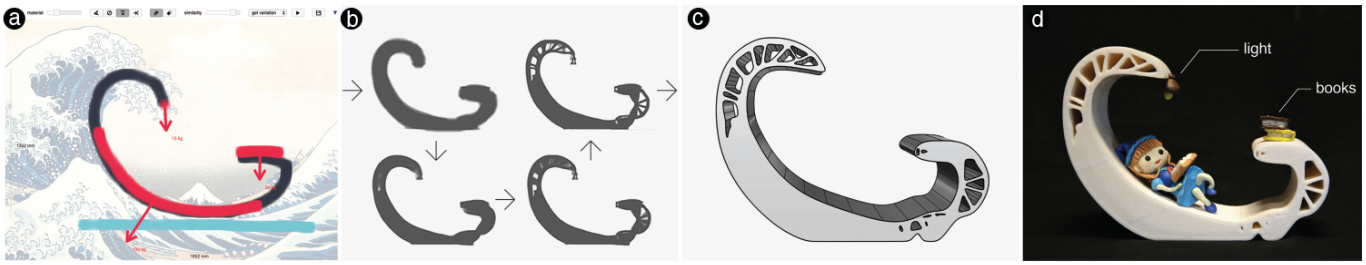

Figure 1. Forte is a 2D design tool that takes a user-driven approach of generative design: a user sketches a reading chair inspired by the painting 'The Great Wave off Kanagawa' (a) with specified loading scenario (red arrow forces, blue ground); Forte then generates structures (with real-time feedback) to support the loads while resembling the user's sketch (b), which can then be post-processed to create a 3D fabrication-ready model (cd).

ABSTRACT

Low-cost fabrication machines (e.g., 3D printers) offer the promise of creating custom-designed objects by a range of users. To maximize performance, generative design methods such as *topology optimization* can automatically optimize properties of a design based on high-level specifications. Though promising, such methods require people to map their design ideas–often unintuitively–to a small number of mathematical input parameters, and the relationship between those parameters and a generated design is often unclear, making it difficult to iterate a design. We present Forte, a sketch-based, real-time interactive tool for people to directly express and iterate on their designs via 2D topology optimization. Users can ask the system to add structures, provide a variation with better performance, or optimize internal material layouts. Users can globally control how much to 'deviate' from the initial sketch, or perform local suggestive editing, which interactively prompts the system to update based on the new information. Design sessions with 10 participants demonstrate that Forte empowers designers to create and explore a range of optimized designs with custom forms and styles.

ACM Classification Keywords

H.5.2. Information interfaces and presentation: User Interfaces-Input devices and strategies

Author Keywords

Generative design; topology optimization; fabrication

CHI 2018, April 21–26, 2018, Montreal, QC, Canada

 ISBN 978-1-4503-5620-6/18/04…$15.00

DOI: https://doi.org/10.1145/3173574.3174070

INTRODUCTION

Low-cost fabrication machines (e.g., 3D printers) offer the promise of enabling people beyond highly trained engineers and technicians to create a variety of objects with customized geometry and functionality. As these machines continue to develop, they no longer just serve as prototyping tools, but gradually become the means to manufacture the final products that will be deployed in real application scenarios [19, 26].

To fully realize the potential of these machines, users' designs need to address performance issues beyond geometric form alone. For example, to create a robotic leg, in addition to designing its shape, users also need to consider how much it weighs and how it can support the weight of the robot's body.

Traditionally, to solve such problems, generative design methods such as *topology optimization* will automatically optimize material distribution given the constraints of space, material amount and other functional requirements [25]. Though promising, such methods provide very few ways for users to directly express or manipulate their designs; oft-times users have to map their design ideas–often unintuitively–to mathematical input parameters. Further, it is often assumed that a CAD model has been designed prior to using such methods. As a result, there is a lack of support for users' ideation process, and they have little direct control of the optimization's outcome [30], or ways to effectively iterate their designs.

To address this problem, prior work and existing tools have been focusing on interactively specifying the input parameters [3, 12], adding template-based textures to the optimized results [20], or contextualizing them with CAD [24] or freeform sketching [18]. However, none of this work lets users directly control how a design should be generated and optimized.

To help address these issues, we present Forte, a sketch-based 2D design tool that takes a user-driven approach for people to

directly express and iterate on their ideas through generative design with real-time visual feedback. Our main contribution is a set of user-driven optimization techniques that allow people to express their design intents, resulting in a combination of structural performance and users' desired styles. Figure 1 shows an exemplar workflow: sketching a reading chair inspired by the famous painting 'The Great Wave off Kanagawa'[1] with a specified loading scenario; Forte then generates structures to support the loads while resembling the user's sketch, which can then be post-processed using external tools (e.g., Rhinoceros) to create a 3D fabrication-ready model.

Specifically, with Forte, users can ask the system to add structures to the original sketch, provide a variation with better performance, or generate optimized internal structures. Users can globally adjust a 'similarity' slider, which controls how much the system will 'deviate' from their initial input; they can also locally edit the design by adding or erasing parts of the sketch, which then interactively prompts the system to update based on this new information. Meanwhile, a visualization informs users of the performance trade-off as they iterate on the design. Building on a fast topology optimization engine [4], Forte enables rapid iteration and exploration, unlike most existing practices that run the process in a non-interactive, batch fashion (even for 2D designs).

We envision Forte will empower professionals who design structures, such as industrial designers, mechanical engineers and architects. To validate our approach, we held design sessions with 10 participants. Our study demonstrates that Forte empowers designers to create and explore a range of optimized designs with custom forms and styles. While our current focus is on a 2D design tool, we showcase three post-processing techniques using external tools to turn a 2D design to a 3D object: *extrusion*, *warping* and *combination*. We report stress analysis on these 3D models under varying loading conditions and finally demonstrate a series of fabricated examples.

RELATED WORK

Forte is a sketch-based interface that takes a user-driven approach to generatively suggest, modify and enhance users' designs. Below we review three areas of related prior work.

Sketch-based Design and Modeling

Despite its limitation to 2D, sketching has been shown as a powerful and expressive medium for design and modeling [22]. For example, Teddy is a sketching interface that allows users, with a few 2D strokes, to create free form 3D models [16]. ILoveSketch and EverybodyLovesSketch are sketching interfaces with a suite of interaction techniques to facilitate the creation of 3D curves, using techniques such as automatically rotating the camera to provide a better viewing angle for sketching, and widgets to select specific sketching surfaces in 3D [5, 6]. Sketching can create functional objects even without reconstructing a fully 3D shape: the SketchChair system demonstrates the capability of sketching a chair by drawing its cross section on a 2D canvas then extruding it to a full 3D model [23].

[1] https://en.wikipedia.org/wiki/The_Great_Wave_off_Kanagawa

Sketching is an appealing alternative to traditional CAD tools for several reasons. Not only can it serve to create 3D models based on a formulated design, it also supports the imagination, ideation and exploration process. As pointed out by Gross and Do, sketching "supports ambiguity, imprecision, and incremental formalization of ideas as well as rapid exploration of alternatives." [13] Thus sketching by its nature embraces a wide range of possibilities, alternatives and suggestions, which has inspired us to take a user-driven approach to co-create designs with users through sketching and optimization. Below we review literature on user-driven suggestive interfaces that informs our approach.

User-Driven Suggestive Interfaces

In a broader context, user-driven suggestive interfaces date back to research in mixed-initiative interaction—an approach of designing intelligent systems that allows users and systems (e.g., an interface agent) to complementarily take on different roles in a task, enabling an optimal collaboration with each being able to "do what they do best" [11]. For example, TRAINS-95 is a dialog-based planning system where a railway manager describes the high-level routing goals while the system detects potential problems (e.g., traffic, weather) [11]. Horvitz summarizes principles of building such mixed-initiative user interfaces, as illustrated in the LookOut system that suggestively prompts and helps users to schedule events from their emails [14]. More recently, mixed-initiative interaction has been demonstrated in tools that help analysts 'wrangle' data into processable format by proactively suggesting plausible transform operation based on user input [28].

In the domain of design tools, user-driven suggestive interfaces allow users to take the lead and drive the design task, while the system observes, analyzes and suggests problems or improvements to enhance the design. For example, Chateau is a 3D sketching tool that predicts a user's next drawings or provide suggestions to complete drawings by observing user input [15]. Tsang et al. use images similar to a target design to guide the creation of 3D curves, and suggest relevant geometry based on users' input strokes [27]. Umetani et al. present a tool for guided furniture design: as users edit their design, the tool indicates structures that are non-durable and/or non-stable, and further offers suggestions for solving these problem [28].

Compared to this prior work, Forte goes beyond producing suggestions based on user input. Specifically, our tool enables users to explore multiple ways of generating designs while being able to customize how these designs are made by specifying a global similarity value or performing local suggestive editing. Further, Forte leverages a generative approach to produce a large space of design suggestions from very simple initial user input. Below we review related work in this generative design domain.

Generative Design

Generative design typically considers designing an object as an optimization process that searches a space of design configurations to find one which best meets an objective function. For example, topology optimization–one of the most popular generative design approaches–iteratively computes

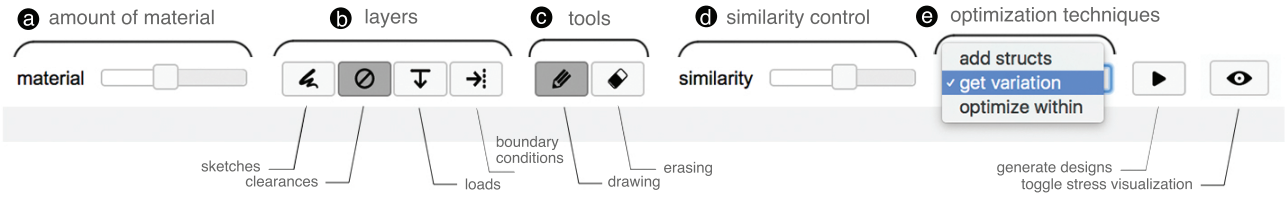

Figure 2. Overview of Forte's default tool bar: adjusting the amount of material for generating the structures (a); different layers for sketching and specifying the loading scenario (b); drawing and erasing tools (c); controlling how the generated results are similar to the original sketch (d); and selecting different design optimization methods (e).

material distribution that seeks to optimize performance, such as minimizing the overall compliance of the object, given the constraints of space, material amount, loads and boundary conditions [7]. There have been research several professional tools that support generative design, e.g., ANSYS [1], Shape Generator in Autodesk Inventor [17], and Grasshopper scripts in Rhinoceros [12].

Most of these tools only provide simple interactive widgets for specifying basic inputs, such as pointing at where the boundaries are and dragging to determine the amount of loads. However, it is often difficult to understand how changing these inputs will affect the final design, and hence hard to iterate using these techniques. To achieve more interactivity, one common obstacle is speed: users often have to wait for hours or even days before coming back to see the results. To overcome the speed limit, Andreassen et al. accelerate the original 99-line program [25] with a speed improvement of over two orders of magnitude [4]. This makes it possible for topology optimization to run on mobile devices that allow users to watch the optimization evolve in real time [3].

However, even with fast iteration, topology optimization still remains as a 'black box', with users often having only very indirect control of the optimization's outcome [30]. To address this issue, eifForm is a generative design tool that combines a traditional modeling system with generation of structures within a model [24]. Martinez et al. allow users to supply exemplar visual patterns and obtain a design that is optimized to be both structurally sound and aesthetically similar to the exemplar [20]. DreamSketch enables users to sketch their ideas (e.g., the seating of a glider) while leaving structural components (e.g., connectors between seats, handles and wings) to be generated by an optimization module [18].

While promising, most of this research, however, does not permit users to directly manipulate or customize the optimization process. Our work demonstrates how a user-driven approach to generative design can give users direct control over the optimization outcomes, and create results that uniquely combine their ideas on aspects such as appearance with the system's perspectives on structures.

FORTE: USER-DRIVEN GENERATIVE DESIGN

In this section, we describe users' interaction with Forte and its key features for user-driven generative design, while leaving the technical details for later in the next section. Specifically, we will walk through a usage scenario: creating a 2D design of a leg for a quadruped robot (Figure 3). The design goal is three-fold: *(i)* in contact with the ground (Figure 3b), the leg needs to be optimized for supporting the weight of the robot's body (Figure 3a), *(ii)* it needs to be lightweight to ensure the overall efficiency of the robot, and *(iii)* its form also needs to be customized with respect to other user-defined goals such as aesthetics. Below we first go through some basic concepts and terminologies and then demonstrate how Forte allows users to jointly achieve these goals.

Basic Concepts and Terminologies

Generative design via topology optimization considers an object as a distribution of a certain amount of material in order to meet certain performance criteria. This paper focuses on structural performance, which is optimizing structures—with only a limited amount of material available—to support a given load. For instance, a robot leg should be designed to support the weight of the robot's body with a lightweight structure. In Forte, a design consists of a sketch (considered below) and a **loading scenario**, which include *(i)* where the expected **loads** are and how large they are, and *(ii)* **boundary conditions**, which is where the object is in contact with its environment (e.g., a robot leg will touch the ground, as shown in Figure 3). Based on this input, Forte will then generate structures via different types of **design optimization**, i.e., distributing allocated material in a structurally optimized way. Each type of optimization is also constrained by a given **amount of material**, which is typically measured by the percentage of the entire **design domain**–the bounding box of the design object (Figure 3c). For example, 30% material means the material used for the robot leg amounts to 30% the size of its bounding box[2].

#1 Sketching Designs and Loading Scenarios

With Forte, a user sketches a design idea as well as the loading scenario. Each type of sketch is drawn in a different color on its own layer (Figure 2b). This allows users to select and focus on one particular layer at a time. Once a layer is

[2]We discuss later in the design sessions that participants found this to be a problematic way of measuring material, as it seems hard to get an intuitive estimate based on bounding boxes.

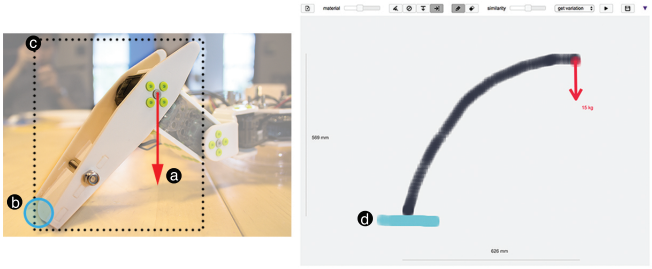

Figure 3. From users' sketch, Forte (right) generates lightweight structures to support loads, such as a robot leg (left) that needs to support the robot's weight using a specified amount of material.

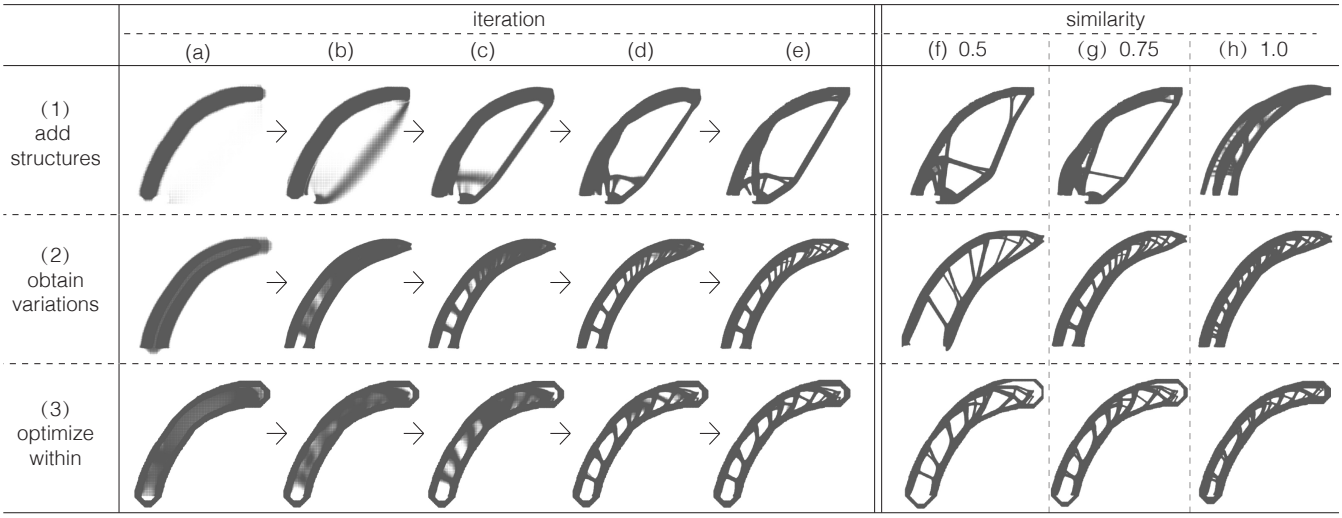

Figure 4. Forte's three types of design optimization iteratively generate structures based on user's sketch while providing them with real-time visual feedback akin to an animation (a-e), the result of which can be controlled by a global 'similarity' value (f-h).

selected, it is pushed to the top with the rest of the layers set to semi-transparent.

To start, a user simply selects the design layer and sketches a curve representing the robot's leg. With Forte, it suffices to provide such a simple representation that succinctly expresses the user's high-level design intent, i.e., "I want the robot's leg to curve like this". As the user sketches the robot leg, Forte also shows the dimensions of the current sketch's bounding box to inform users of the size of their design. The default scale is 20 mm per pixel, which can also be adjusted by the user in the preference settings.

As a next step, the user would consider the loading scenario. First, to create a load, the user switches to the load layer (Figure 2), They can click on the sketch to indicate a single loading point, or draw to specify a sequence of loading points. As shown in Figure 5, as soon as these points are drawn (e.g., mouse released), an arrow is rendered, following the cursor (e.g., where the mouse moves), and indicating the direction as well as the amount of load, which is also shown at the end of the arrow. A click confirms and pins down this load vector. To specify a boundary condition, the user simply draws it on the corresponding layer (Figure 3d).

In addition, Forte also supports a fourth type of layer that allows users to lasso-select a **clearance** region that needs to be kept clear, e.g., the space above a chair's seat or around the tip of a hook (Figure 6). This information proactively prevents generating structures at places that will compromise the usage of an object.

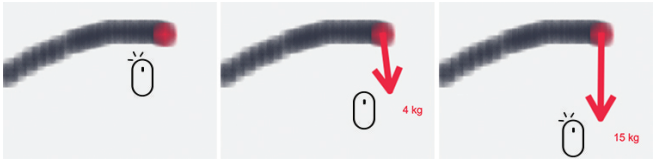

Figure 5. Interaction to specify a load.

#2 Design Optimization with Global Similarity Control

Once the user creates a sketch and specifies the loading scenario, they can now explore three different ways the system can generate structures by design optimization. Forte considers the generation of structures as a spectrum between users' original idea and the system's solution to the defined problem. Users can navigate on this spectrum by adjusting a *similarity* value from 0 to 1 (Figure 2d): conceptually, a low similarity would allow the system to 'deviate' more from the user's initial sketch while a higher value will keep the generated structures closer to what the user originally draws. Further, Forte's optimization algorithms run at an interactive rate, providing users with real-time visual feedback after each iteration, as demonstrated in the accompanying video figure. Below we illustrate Forte's three new generative design optimizations and how similarity allows users to have direct control of the generated results.

Add structures To start, Forte can generate additional structures to strengthen the user's original sketch. For example, as shown in Figure 4(1a-1e), a major truss structure is added parallel to the user-sketched robot leg as reinforcement.

The similarity value effectively controls how far away from the original sketch these structures can be added. As the user increases similarity, added structures will appear increasingly closer to the original sketch and eventually become part of it, equivalent to a thickening operation (Figure 4(1h)). However, note that the added structures' position does not change smoothly with the similarity value; rather it tends to 'snap' to positions along the way that are structurally meaningful.

Figure 6. Lasso-selecting an area of clearance around the tip of the hook (no structure generated here).

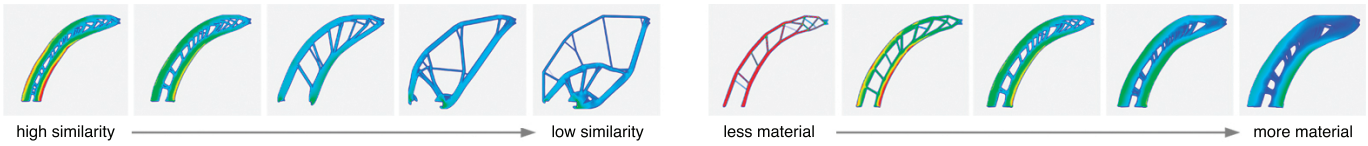

Figure 7. Stress visualization informs performance trade-off between different similarity values (left), and the amount of material (right).

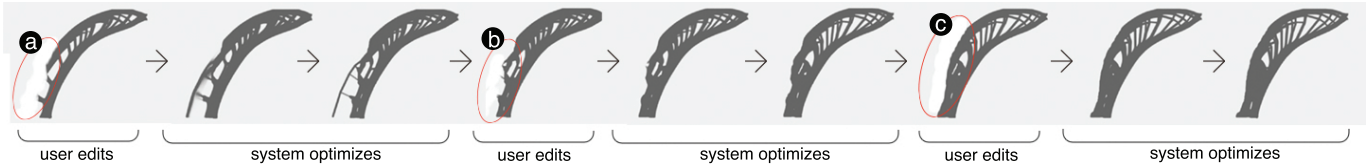

Figure 8. Users can suggestively erase part of a generated result (ab), which prompts the system to update the optimization addressing the intent of removing material; with this tool, users can also refine or smoothen a generated result (c).

Overall, 'add structures' is targeted at situations where users only have a partial or incomplete design idea in mind and are unsure whether and how they should add extra structures to provide support for the loads.

Obtain variation Forte provides a second optimization algorithm that can generate variations of the initial sketch. For example, as shown in Figure 4(2a-2e), Forte generates two parallel interconnected trusses instead of one user-sketched structure; however, the variation still bears resemblance to the original sketch as it globally follows the curve path specified by the user. As the user increases the similarity value, the generated structures will follow the user's sketch more closely (Figure 7). 'Obtain variation' works best when users only have a concept (e.g., a skeleton) of what the design should look like without any structural details, in which case the system will take the initial designs as 'seeds' and vary it with generated structures.

Optimize within Forte can also consider the user's sketch as solid shapes (rather than skeletons) and optimize the internal structures within. For example, as shown in Figure 4(3a-3e), Forte reproduces the contour of the user-sketched robot leg while filling in generated structures. By controlling similarity, the user can effectively shrink or expand the contour, e.g., decreasing similarity results in a robot leg of the same curvy shape but much thicker than the original sketch, given the system more space to generate structures. 'Optimize within' serves well for cases where users already have a fully-designed solid shape but want to re-design the interior structures.

For any of these design optimizations, users can adjust how much material can be used in generating the design. For example, as shown in Figure 7, reducing the material amount often creates hollow structures with trusses, while a higher value tends to thicken or fill up certain parts of the design.

As the user runs one of the three optimizations, Forte creates and records a trial, and together displays a history of results from all these trials. Selecting a trial not only shows the resulted structures, but also updates the menu so that users can see which of the three optimizations was used, how much material was allocated and what similarity value was applied.

#3 Refining Optimization with Local Suggestive Edits

The similarity value allows users to control the optimization outcome at a global level with respect to the entire design. Complementarily, once a result is generated, users can also perform fine local editing, which will prompt the system to re-run the optimization, taking into account users' underlying intent of the edit. For example, if a user wants a thinner tip of the robot leg, they can use the eraser tool (Figure 2c) to scrub that area (Figure 8a, shadowed part). The system immediately starts a new optimization and returns an updated result with less material on the tip of the leg. Increasing the erasing 'power' and applying it again finally sharpen the leg, but results in a rough outline (Figure 8b). We can also use this tool to smooth the optimized result (Figure 8c).

Similarly, the user can also use the pen tool (Figure 2c) to draw on an optimization result, such as adding a truss to the leg (Figure 9ab). The system will also re-run the optimization and will attempt to generate some structures where the new truss is drawn (Figure 9).

This technique is called suggestive editing, as users' action do not directly remove or add to the design; rather, it serves to suggest their intent to the system, e.g., "I want less material here", or "I would like to add something there". The next round of structure generation will take into account such intent and attempt to realize it through the optimization process. However, intent cannot be realized if the removal or addition of material compromises the structural integrity of the design (e.g., removing the entire bottom part of the robot leg, making it unable to stand on the ground).

#4 Informing Performance Trade-offs with Visualization

As users exploratively customize the optimization process, it is important to inform them of the performance trade-offs. Forte

Figure 9. Users can suggestively add material to a generated result (circled strokes), which prompts the system to re-run the optimization and attempt to add structures accordingly.

allows users to toggle a stress visualization (Figure 2) overlaying each optimized result (Figure 7) . The heat map indicates which part of a design is the weakest; it also allows users to compare different designs' performance and see how the stress situations evolve as the design evolves. Users can also adjust the safety factor: a higher value is equivalent to assuming more weight, thus revealing more structurally problematic areas in the design.

IMPLEMENTATION

In this section we describe technical details of Forte's implementation.

#1 Sketching Designs and Loading Scenarios

Recall that users' design input consists of a sketch within a design domain as well as loads, boundary conditions (Figure 3), and/or clearance (Figure 6).

Given these input parameters, first the design domain is discretized into $N = W \times H$ square **finite elements** [31], each of which (denoted as e) is assigned a **density** value $x_e \in [0,1]$. The user's intial sketch is discretized into a binary density map on this domain with 1's for elements overlapping with the sketch and 0's for the others. Similarly, loads, boundary conditions and clearance are also converted to such density maps. In addition, for loads each density=1 element is also associated with a load vector indicating the amount of the load as well as the direction.

#2 Design Optimizations with Global Similarity Control

The implementations of the three design optimization techniques presented here are developed based on the 88-line program [4], which uses a SIMP (Solid Isotropic Microstructure with Penalization) method for topology optimization. For readers to better understand our approach, we first provide a brief of overview and refer the reader to Andreassen et al. for further details [4].

Overall, the process solves the following compliance optimization problem:

$$\min_x : c(\mathbf{x}) = \mathbf{U}^T\mathbf{K}\mathbf{U} = \sum_{e=1}^{N} E_e(x_e)\mathbf{u}_e^T\mathbf{k}_0\mathbf{u}_e, \qquad (1)$$

subject to:

$$V(\mathbf{x})/V_0 \leq f;\ \ \mathbf{K}\mathbf{U} = \mathbf{F};\ \ 0 \leq x_e \leq 1$$

where c is the compliance of the design, which is computed with $\mathbf{U}$ – the global displacement matrix, and $\mathbf{K}$ – the global stiffness matrix determined by the stiffness of the material as well as its distribution (or layout). The objective function is further broken down to the summation of compliances across all elements: E_e is the Young's modulus of e ($E_e = E_{min} + x_e^p(E_0 - E_{min})$), $\mathbf{k_0}$ is the unit stiffness matrix which together with E_e determines the stiffness of element e; u_e is the displacement vector of e. Finally, $\mathbf{U}$ can be computed from $\mathbf{K}\mathbf{U} = \mathbf{F}$, where $\mathbf{F}$ is vector of forces (loads).

At each iteration, once c is computed, x_e is updated as follows:

$$x_e^{new} = \begin{cases} max(0, x_e - m), & \text{if } x_e B_e^{\eta} \leq max(0, x_e - m) \\ min(1, x_e + m), & \text{if } x_e B_e^{\eta} \geq min(1, x_e - m) \\ x_e B_e^{\eta} & \text{otherwise} \end{cases} \qquad (2)$$

where m is a pre-set step for density changes, $\eta = 1/2$ is a numerical damping coefficient, and $B_e = (-\frac{\partial c}{\partial x_e})/(\lambda \frac{\partial V}{\partial x_e})$, with λ being a constraint factor to ensure the total volume does not exceed $V_0 f$—the allocated amount of material where f is the percentage and V_0 is the volume of the design domain.

This process represents the classic topology optimization approach that automates the generation of structures in a 'black box'. Below we describe our methods of extending it to a user-driven approach, allowing the control of the optimization results via a similarity value.

Add structures We initialize the design domain by assigning each element the value of f (e.g., 15% of the bounding box area). Through iterations the optimization will increase or decrease this value for a given element, thus forming a global structure. After each iteration, we add the intermediate generated structures to the user's sketch, and take it as the input for the next iteration. Specifically, now each element density is computed as

$$\widehat{x_e^{new}} = \begin{cases} 1, & \text{if } e \in user\ sketch \\ x_e^{new} & \text{otherwise} \end{cases} \qquad (3)$$

To control similarity, we perform a *mass transport* (cf. [10]) after each iteration before adding the intermediate results to the user's sketch. A mass transport considers two designs (e.g., user sketch vs. optimized result) as two ways of distributing the same amount of mass (computed by summing up elements' density), and transports the mass in between the two distributions to obtain an interpolation. We use mass transport to interpolate between the optimized result and the user sketch, taking similarity as the weight of interpolation (0: user sketch — 1: optimized result). This allows us to control how much the design will be similar to the original sketch after adding the optimized result.

We implement the mass transport function by computing and interpolating the barycenters of the two input designs as distributions. We compute the distance field of the user sketch and use it as the input distribution to the mass transport function. This provides more fine-grained information at each element (what is the distance of this element to the user sketch) beyond a binary value (whether or not it is part of the user sketch).

Obtain variation Instead of f, we initialize the design domain with the user's sketch, and then run topology optimization, which will effectively 'shift' the material away from user-specified locations to achieve a variation with optimized material layout.

To support user control of similarity, we map the distance field of the user's sketch to the input design domain as

$$x_e^{initial} = \frac{1}{\sum_{i=1}^{N} cos(\frac{\pi d(i)}{2})^s} cos(\frac{\pi d(e)}{2})^s f \qquad (4)$$

where $x_e^{initial}$ is the initial material density at element e, d is the normalized distance field function, s is the user specified similarity request, and the cos function raised to s produces value close to 1 when $d(e)$ is small (close to the user's sketch) and drops rapidly to 0 when shifting away. s controls how

quickly such a drop occurs. Thus a larger similarity value will initialize the optimization (hence the result as well) more closely to the user's sketch, while a smaller value will loosen the constraint by 'diluting' the user's sketch, and produces a result with wilder variation.

Optimize within We consider the user's sketch as a 'skeleton'. As a first step, we expand it using the distance field by setting a new contour around the original sketch. We then run the optimization only within the contour, as follows, at each iteration,

$$\widehat{x_e^{new}} = \begin{cases} x_e^{new}, & \text{if } d(e) < d_c \\ 1, & \text{if } d_c \leqslant d(e) < d_c + t \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

d_c is a distance field value used to define the new contour, and t controls the thickness of the contour. We control similarity by shifting c towards or away from the user's original sketch.

#3 Refining Optimization with Local Suggestive Edits

Forte enables users to interactively suggest adding to, or removing from, parts of the optimization result, which then prompts the system to re-run the optimization to produce a version that matches the users' intent, including the new edits. Importantly, in such re-run, the system is initialized based on the result from the previous run, rather than starting from the original sketch.

As a user draws on, or erases, parts of a design on a $W \times H$ domain, we create a mask M of the same dimensions with each pixel's value $M_e \in [1-\Delta_{remove}, 1+\Delta_{add}], \Delta_{remove} \in (0,1), \Delta_{add} > 0$. Then in the optimization process, we update element e as

$$\widehat{x_e^{new}} = M_e x_e^{new} \quad (6)$$

In essence, M_e indicates whether and how much a user would want to add or remove material at this point. If a user performs an erasing action, M_e is set to be in $[1-\Delta_{remove}, 1)$, which will dampen the to-erase elements' density at each iteration; if a user performs additional drawing, we integrate the addition to the result as input for the re-run. As M_e is now set to be in $(1, 1+\Delta_{add}]$, it will amplify the to-add elements' density at each iteration. All unedited M_e's will be set to 1.

#4 Informing Performance Trade-offs with Visualization

We compute the von Mises stress of each optimized result using Biyikli and To's method [9], which is implemented as an integral component of the optimization process. The values in the stress distribution are then normalized by a material-specific yield stress σ_y and visualized as a heat map. All our examples and demonstrations in the design tool use a material model based on an Extrusion Grade PLA [21]. The yield stress is also divided by the user-specified safety factor, e.g., a $2\times$ safety factor will result in the use of $\sigma_y/2$, thus visually revealing more potentially problematic area in the design.

Software and Hardware

The front end of Forte is written in JavaScript using jQuery[3] for UI development. The back end consists of two components: *(i)* the optimization techniques were written in MATLAB, which is then compiled into a standalone service; and *(ii)* a phython-based server that relays optimization requests and input parameters from the front end to the MATLAB service, and sends back the results. Everything runs on a MacBook Pro (Retina, 15-inch, Early 2013) with a 2.4 GHz Intel Core i7 and 8 GB 1600 MHz DDR3 memory. In our demonstrations, the front end runs on a Google Chrome web browser.

[3]jQuery: `https://jquery.com/`

DESIGN SESSIONS

To validate Forte, we conducted informal, qualitative design sessions with 10 participants. The objective of the study is to let participants create their own designs using Forte's user-driven generative approach, and elicit their initial reaction and feedback to the system to reveal existing problems and to inform future improvements.

Participants We recruited 10 participants from our university (1 female, aged 18-33). Two majored in interaction design, three in architecture and the others in mechanical engineering. All participants reported to have sketching experience and all had experience using CAD tools. Two reported to be knowledgeable about topology optimization, four had passing knowledge and the others had none.

Tasks & Procedure The design sessions took place in our research lab and lasted between 1 and 1.5 hours for each participant. To begin, participants were given a tutorial by the experimenter on some basic understanding of topology optimization, and how to use Forte by walking through a concrete example of designing a bracket. Next, they were asked to use Forte to create objects of their own design, as well as coming up with loading scenarios. For each design, they were free to choose any of the three optimization techniques and run as many trials as they would like. Participants created 2-4 designs during the course of the study.

Throughout the design session, participants were asked to think aloud as the experimenter took notes. they completed a questionnaire to summarize their experience with Forte, e.g., what they like about the tool, what problems they encountered, and what new features they would like to have in the future.

Feedback and Discussion

We analyzed the qualitative data using a method akin to the Affinity Diagram approach [8]. We organized notes of participants' comments and survey response to iteratively develop meaningful and coherent themes.

Overall, participants responded positively to Forte's user-driven generative design approach: "I like how it can produce interesting shapes, i.e., organic, asymmetric shapes which are difficult/time-consuming to hand-draw" (P1); "... pretty nice about making the idea of topology optimization accessible to people" (P5); "computer generating structures from my sketch is interesting; it might be the way computers can design something like humans do in the future" (P10).

Optimization techniques Participants were able to learn and recognize the difference between the three optimization techniques. Further, some developed their own understanding and preferences of the techniques. For example, P1 mentioned

that "'add structures' lead to traditional shapes; 'obtain variation' is more creative." P4 liked to use 'add structures' as he wanted "the system to add something different from the original drawing". P9 extensively used 'optimize within' as he wanted to keep the exact shape of the drawing and only optimize what is inside.

Similarity control Participants had no trouble understanding or using the similarity control. P1 commented that "... it allows me to control how much imagination I need". P4 liked that he could control how different the result will look from the original design. P7 summarized the experience of controlling similarity as having "design freedom", and felt he "can design anything and [Forte] will keep it to some extent [after optimizations]".

Real-time feedback Participants did not report any perceived latency; they found it pleasant and useful to see how the design evolved in real time: "[I] enjoy watching the shape evolving" (P1), "Watching it grow and iterate is really nice–it helps you visualize stresses and know where the strength needs to be" (P2), "It helps a lot to see how things are branching out, happening in the background" (P7).

Local suggestive editing We observed only three participants frequently used local suggestive editing. They all appreciated this capability of refining the optimized design: "... very easy to make quick changes on the fly to see how overall structures change" (P2); "... add and subtract material dynamically which was very helpful" (P7). P10 would continuously use this feature in a sequence of trials to add more reinforcement based on the stress visualization.

Existing problems and suggested features Participants also pointed out existing problems and suggested new features in the tool. One main confusion is the amount of material: P2, P4 and P5 thought it was relative to the sketch, whereas it was actually relative to the sketch's bounding box. P5 also suggested visualizing the actual size of allocated material to give users a more concrete idea. Four participants (P2, P4, P7 and P10) asked how they can 'transfer' an optimized result to the original drawing, or simply just be able run technique B on technique A's result. Such request points to an interesting problem for future work, which is enabling users to manage how an initial design branches out to a complex tree structure. Finally, participants proposed other controls akin to the similarity variable, e.g., keeping the design balanced (P1), controlling the global maximum thickness of the generated trusses (P5), and specifying symmetric structures (P6, P7), all of which suggest fertile opportunities for future work.

Designs Created by Participants Figure 10 shows a series of designs created by the participants. Overall, we observe two recurring approaches of creating these designs. First, some users would take a 'form-first' approach, whereby they started by drawing their imagined form and then drove the system to generate structures similar to the form. For example, 'public bench' by P1 (Figure 10a) started with a single curve as a shared sitting space, while the system generated structures underneath to support it. 'Cross-legged book case' by P6 (Figure 10e) started with a form that almost looked like a Small Seal Script[4] character in ancient Chinese. In contrast, other users would take a 'structure-first' approach, starting with a basic structure based on the object's functionality; they explored which of the system's generated structures can also create interesting forms to the design. For example, 'eagle crane' by P2 (Figure 10f) was initially drawn as a simple polygon that provides the functionality of a crane. However, as the system 'morphed' the input structure, it evolved into an eagle head shaped design. 'House' by P8 also started with simple geometry representing a two-story house, a loft and a patio, which then evolved into two distinct design variations (Figure 10 g and h, using 'obtain variation' and 'add structures', respectively).

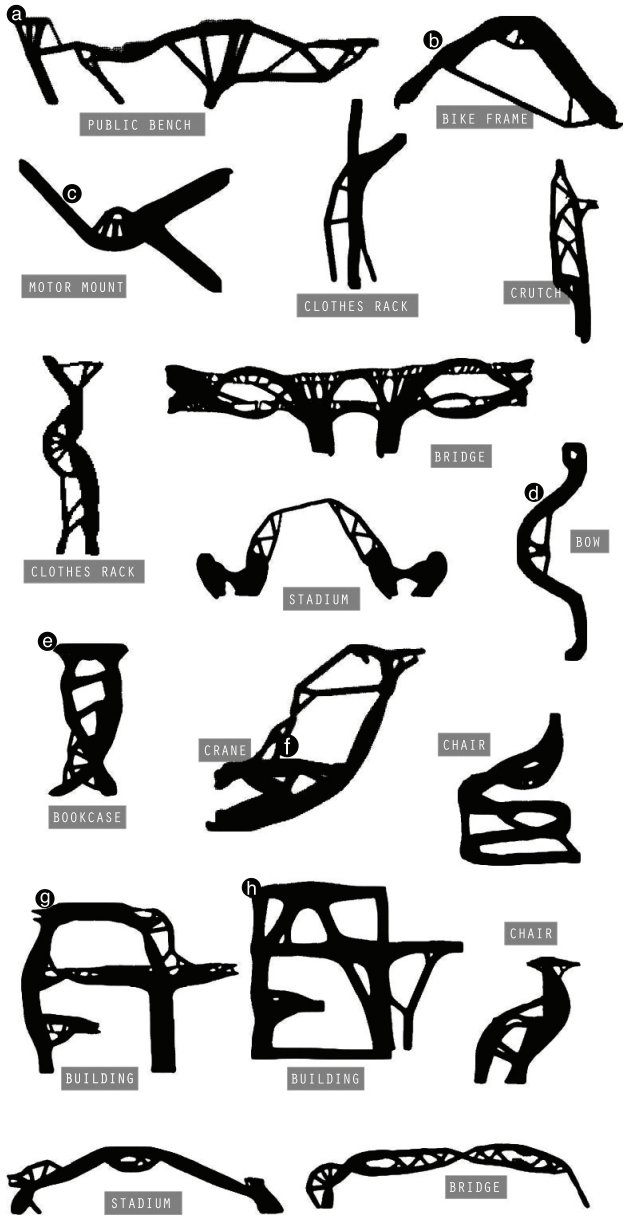

Figure 10. Designs created by 10 participants using Forte.

[4]Small Seal Script. **https://en.wikipedia.org/wiki/Small_Seal_Script**

We also observed different iteration strategies. Most users iterated a design by playing with different similarity values and amounts of material. For example, P3 used Forte to design (the cross-section of) the GE Aircraft Engine Bracket (Figure 10c). His first three trials were simply trying out different material-similarity combination to gauge variety of the underlying design space. Finally, quite a few users modified their initial design based on the system's generated structures. For example, in designing a bike frame (Figure 10b), P2 noticed that the system did not keep one of the trusses he drew but instead created a different one. P2 then modified the initial design to a 'imitate' the system's approach. Participants also tried to modify the system's design. As mentioned above, three participants were observed actively using the local editing techniques. For example, In designing a bow P7 used the eraser tool to iteratively refine the shape of the bow updated by the system ((Figure 10d).

Figure 10 shows other designs created by our participants.

POST-PROCESSING AND FABRICATED RESULTS

Forte enables users to formulate, express and optimize their design in a 2D environment. To fabricate their designs into 3D artifacts, we describe three post-processing techniques using external tools (e.g., the Rhinoceros[5] CAD system) for fabrication and showcase some of the fabricated results.

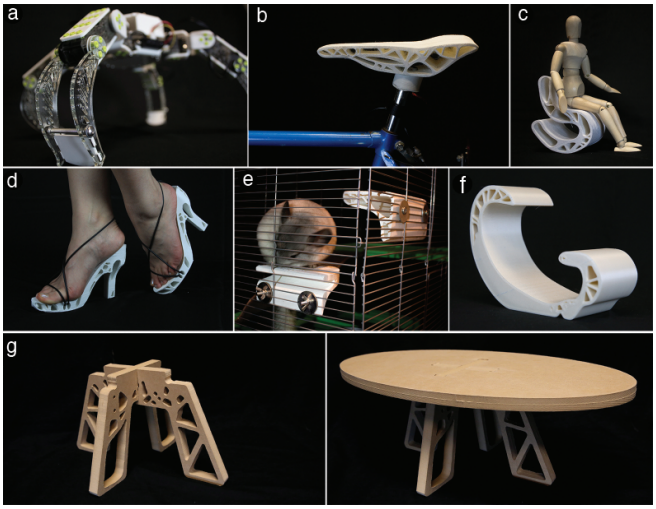

Figure 11. Fabricated examples designed from Forte's generated structures: legs for a quadruped robot (a); a bike seat (b); an S-shaped chair (c); high heels (d); pet jumping platforms (e); a reading chair (f); and a tea table (g).

Extrusion We can extrude a 2D design along a path to create a 3D object. First we convert a generated design from a bitmap to a vector graphic, which can already be used in a laser cutter to create planar objects, such as the legs for the quadruped robot (Figure 11a). Further we can also import the vector graphics to a 3D modeling tool for creating a 2D surface and extruding it to a 3D object. We use this technique to create a multi-functional reading chair that consists of an overhead light, a reclined sitting area, and bookshelves (Figure 11f). Extrusion is not limited to a perpendicular path: for example, it is possible to extrude the 2D chair profile in Figure 11c along a circular path to create a bench.

Warping We can also warp a 2D design into a 3D surface. We used this technique to design a pair of high heel shoes using generated structures as the base (Figure 11d). We designed a jumping platform for chinchillas by warping the cross-section to form a rounder top (Figure 11e). We designed a bike seat by warping a 2D generated pattern to the shape of a cushion (Figure 11b).

Combination We can join multiple planar components made from the 2D designs (e.g., fabricated using laser or CNC cutting) to create 3D structures. For example, we designed a tea table supported by two joined structures (Figure 11g).

To test the structural integrity of the 3D models produced in the post-processing step, we performed Finite Element Analysis (FEA) using Autodesk Inventor [17]. As shown in Figure 12, we tested each model on a range of increasing loads—from normal to overloading—and found consistent stress distribution across all the cases with high structural performance[6]. As our current focus is on the 2D design tool, this test is only based on our current examples (as well as specific types of material chosen for running FEA). Further experiments are required in order to validate the general process of converting 2D structural patterns to 3D objects manufactured in a wide range of materials, which we leave as future work.

LIMITATIONS, TRADE-OFFS AND FUTURE WORK

Forte presents algorithmic approaches that add interactivity to otherwise 'batch' optimizations. While these are focused on a specific type of optimization, we believe that they do provide exemplars and starting points for development of similar algorithms in other optimizations in the future.

Going from 2D to 3D The implementation of Forte is extensible to 3D: on the front end, prior work has demonstrated numerous techniques for enabling sketching in 3D [16, 5, 6]; on the back end, the optimization can also handle 3D cases by adding a third dimension to the input parameters (design domain, loads, boundary conditions, etc.) as well as the material stiffness matrix. Perhaps the real challenge of going 3D is maintaining real-time feedback and interactive iteration. Past work has explored engineering solutions for handling 3D generative designs. For example, Aage et al. provide a topology optimization implementation based on PETSc (Portable and Extendable Toolkit for Scientific Computing), which enables incredibly fine discretization, e.g., a (3D) design with 27.6 million design elements running on 24 CPUs (144 cores) only requires 30-60s per iteration [2]. Building upon all this prior work, our next step will develop and extend Forte to a fully 3D design tool.

Ambiguity and imprecision of sketching As mentioned in Gross and Do's work, sketching is a particularly appealing medium for early ideation and design formation, as it embraces ambiguity and imagination. As discussed in the design session,

[5] `https://www.rhino3d.com/`

[6] We define the range of loads based on the usage scenarios of each design. The heatmap visualization is based on the maximum stress amongst all five loads.

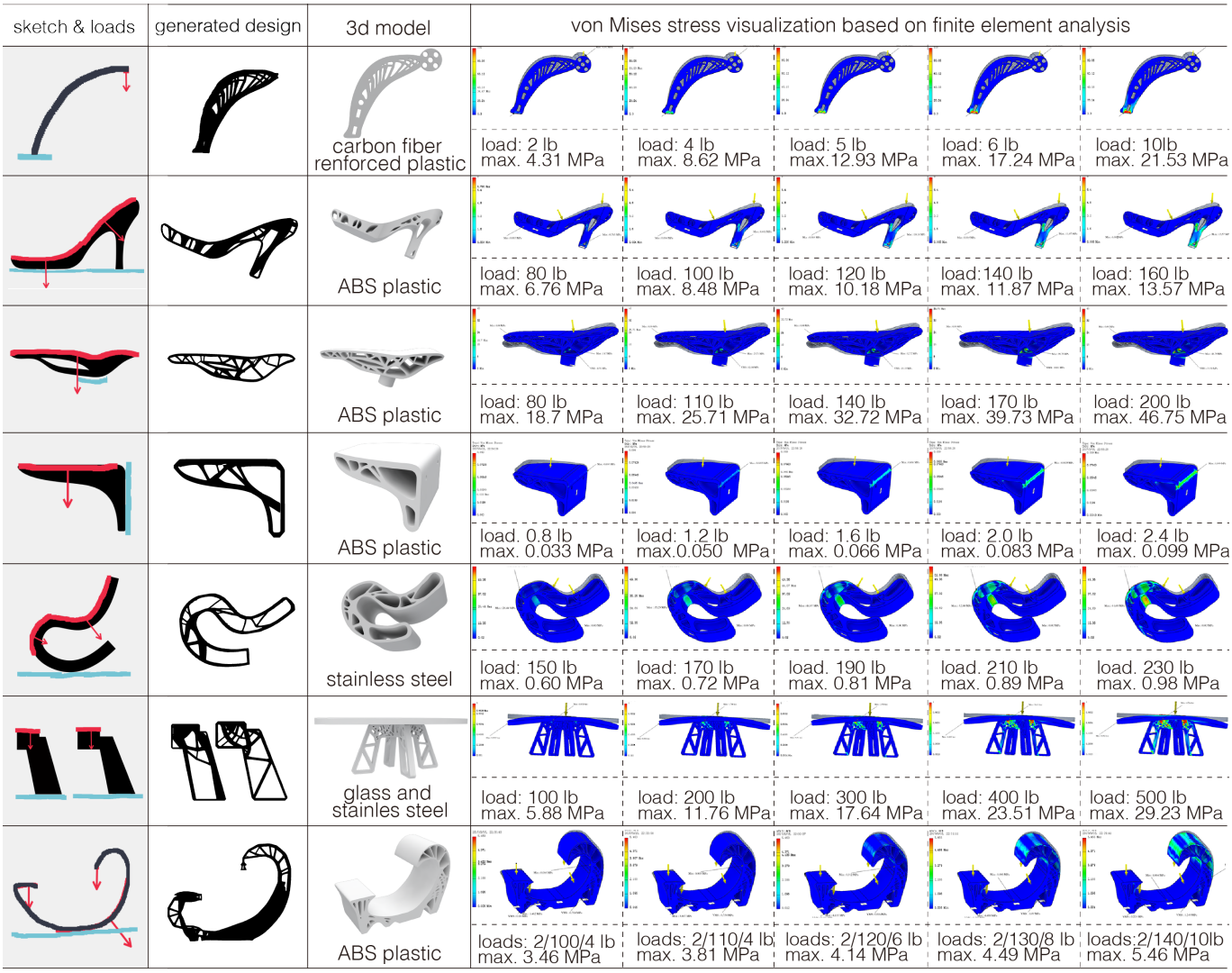

Figure 12. All our fabricated examples from the sketches in Forte, to generated 2D designs, and to post-processed 3D models; we performed a Finite Element Analysis on each 3D model based on the input loading scenario, which shows consistent stress distribution across a range of increasing loads.

this is also a strength of Forte, as it allowed participants to use unique hand-drawn forms to generate structures. However, as pointed out by a few participants, Forte lacks support for more precise design, e.g., straight lines, smooth curves, and symmetry. In the future versions of Forte, we would like to add more controlled ways of specifying designs.

Generating and exploring a large space of examples Currently each of Forte's optimized result is initiated and driven by the users. While this gives users freedom to explore and customize each optimization trial, it becomes a somewhat tedious process as the number of trial increases. In the future we plan to explore ways to automatically generate a large design space of structures from a single user input. To realize this, the challenge is two-fold: *(i)* how to infer and sample more variables from one single user input in order to generate more than one result; *(ii)* how to enable users to efficiently navigate and filter a large number of results. Our future work will explore solutions to tackle these two challenges.

Enabling more semantic controls of design optimization On the input side, we are interested in enabling more semantic controls of design optimization beyond the similarity value. For example, in Yumer et al.'s work [29] they developed a number of semantic metrics for 3D models by having the crowd perform pair-wise comparison on a large data set. In the future, we want to let users label and quantify each optimization result, e.g., 'how organic is this bike frame' and 'how thin or muscular is this chair'. By collecting these quantified labels, we hope to map the space of the input parameters (e.g., amount of material, similarity, clearance) to the semantic space of the optimization outcome. This can potentially enable the 'transfer' of design, e.g., applying a set of input parameters from an organic-looking bike frame design to turn this chair into a similar style.

ACKNOWLEDGMENTS

Thanks to reviewers and colleagues in CMU HCII. This work was funded in part by the National Science Foundation under grant IIS-1718651 and a Google Faculty Research Award.

REFERENCES

1. ANSYS 18. 2017. Topology Optimization: Applications. http://www.ansys.com/products/structures/topology-optimization. (2017). (Accessed on 08/31/2017).
2. Niels Aage, Erik Andreassen, and Boyan Stefanov Lazarov. 2014. Topology optimization using petsc. *Struct Multidiscip Optim* (2014).
3. Niels Aage, Morten Nobel-Jørgensen, Casper Schousboe Andreasen, and Ole Sigmund. 2013. Interactive topology optimization on hand-held devices. *Structural and Multidisciplinary Optimization* 47, 1 (2013), 1–6.
4. Erik Andreassen, Anders Clausen, Mattias Schevenels, Boyan S Lazarov, and Ole Sigmund. 2011. Efficient topology optimization in MATLAB using 88 lines of code. *Structural and Multidisciplinary Optimization* 43, 1 (2011), 1–16.
5. Seok-Hyung Bae, Ravin Balakrishnan, and Karan Singh. 2008. ILoveSketch: as-natural-as-possible sketching system for creating 3d curve models. In *Proceedings of the 21st annual ACM symposium on User interface software and technology*. ACM, 151–160.
6. Seok-Hyung Bae, Ravin Balakrishnan, and Karan Singh. 2009. EverybodyLovesSketch: 3D sketching for a broader audience. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology*. ACM, 59–68.
7. Martin P Bendsøe, Ole Sigmund, Martin P Bendsøe, and Ole Sigmund. 2004. *Topology optimization by distribution of isotropic material*. Springer.
8. Hugh Beyer and Karen Holtzblatt. 1997. *Contextual design: defining customer-centered systems*. Elsevier.
9. Emre Biyikli and Albert C To. 2015. Proportional Topology Optimization: A New Non-Sensitivity Method for Solving Stress Constrained and Minimum Compliance Problems and Its Implementation in MATLAB. *PloS one* 10, 12 (2015), e0145041.
10. Nicolas Bonneel, Michiel Van De Panne, Sylvain Paris, and Wolfgang Heidrich. 2011. Displacement interpolation using Lagrangian mass transport. In *ACM Transactions on Graphics (TOG)*, Vol. 30. ACM, 158.
11. George Ferguson, James F Allen, Bradford W Miller, and others. 1996. TRAINS-95: Towards a Mixed-Initiative Planning Assistant.. In *AIPS*. 70–77.
12. Grasshopper. 2017. Stress topology optimization with Millipede. http://www.grasshopper3d.com/forum/topics/stress-topology-optimization-with-millipede. (2017). (Accessed on 08/23/2017).
13. Mark D Gross and Ellen Yi-Luen Do. 1996. Ambiguous intentions: a paper-like interface for creative design. In *Proceedings of the 9th annual ACM symposium on User interface software and technology*. ACM, 183–192.
14. Eric Horvitz. 1999. Principles of mixed-initiative user interfaces. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. ACM, 159–166.
15. Takeo Igarashi and John F Hughes. 2001. A suggestive interface for 3D drawing. In *Proceedings of the 14th annual ACM symposium on User interface software and technology*. ACM, 173–181.
16. Takeo Igarashi, Satoshi Matsuoka, and Hidehiko Tanaka. 2007. Teddy: a sketching interface for 3D freeform design. In *ACM SIGGRAPH 2007 Courses*. ACM, 21.
17. Autodesk Inventor. 2017. About Shape Generator. https://knowledge.autodesk.com/support/inventor-products/learn-explore/caas/CloudHelp/cloudhelp/2016/ENU/Inventor-Help/files/GUID-D74F47F3-FE22-44EF-85BE-7C6B1F56DCF9-htm.html. (2017). (Accessed on 08/31/2017).
18. Rubaiat Habib Kazi, Tovi Grossman, Hyunmin Cheong, Ali Hashemi, and George Fitzmaurice. 2017. DreamSketch: Early Stage 3D Design Explorations with Sketching and Generative Design. To appear at the 30th Annual ACM Symposium on User Interface Software and Technology.
19. MarkForged. 2017. Metal X. https://markforged.com/metal-x/. (2017). (Accessed on 08/23/2017).
20. Jonàs Martínez, Jérémie Dumas, Sylvain Lefebvre, and Li-Yi Wei. 2015. Structure and appearance optimization for controllable shape design. *ACM Transactions on Graphics (TOG)* 34, 6 (2015), 229.
21. MatWeb. 2017. NatureWorks Ingeo^{TM} 2002D Extrusion Grade PLA. http://www.matweb.com/search/DataSheet.aspx?MatGUID=1e288619764846d2b794bd077e7f1bba&ckck=1. (2017). (Accessed on 09/05/2017).
22. Luke Olsen, Faramarz F Samavati, Mario Costa Sousa, and Joaquim A Jorge. 2009. Sketch-based modeling: A survey. *Computers & Graphics* 33, 1 (2009), 85–103.
23. Greg Saul, Manfred Lau, Jun Mitani, and Takeo Igarashi. 2011. SketchChair: an all-in-one chair design system for end users. In *Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction*. ACM, 73–80.
24. Kristina Shea, Robert Aish, and Marina Gourtovaia. 2003. Towards integrated performance-based generative design tools. *Digital Design, ECAADE* (2003), 253–264.
25. Ole Sigmund. 2001. A 99 line topology optimization code written in Matlab. *Structural and multidisciplinary optimization* 21, 2 (2001), 120–127.
26. HP Official Site. 2017. HP 3D Printers and Printing Solution. http://www8.hp.com/us/en/printers/3d-printers.html. (2017). (Accessed on 08/23/2017).
27. Steve Tsang, Ravin Balakrishnan, Karan Singh, and Abhishek Ranjan. 2004. A suggestive interface for image guided 3D sketching. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. ACM, 591–598.

28. Nobuyuki Umetani, Takeo Igarashi, and Niloy J Mitra. 2012. Guided exploration of physically valid shapes for furniture design. *ACM Trans. Graph.* 31, 4 (2012), 86–1.

29. Mehmet Ersin Yumer, Siddhartha Chaudhuri, Jessica K Hodgins, and Levent Burak Kara. 2015. Semantic shape editing using deformation handles. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 86.

30. Yahan Zhou, Evangelos Kalogerakis, Rui Wang, and Ian R Grosse. 2016. Direct shape optimization for strengthening 3D printable objects. In *Computer Graphics Forum*, Vol. 35. Wiley Online Library, 333–342.

31. Olgierd Cecil Zienkiewicz, Robert Leroy Taylor, Olgierd Cecil Zienkiewicz, and Robert Lee Taylor. 1977. *The finite element method*. Vol. 3. McGraw-hill London.