

# ThermalRouter: Enabling Users to Design Thermally-Sound Devices

Alex Mazursky  
University of Chicago  
alexmazursky@uchicago.edu

Daria Shifrina  
University of Chicago  
dshifrina@uchicago.edu

Borui Li  
University of Chicago  
borui2000@gmail.com

Joyce E. Passananti  
University of Chicago  
joycep@uchicago.edu

Shan-Yuan Teng  
University of Chicago  
tengshanyuan@uchicago.edu

Svitlana Midianko  
University of Chicago  
s.midianko@gmail.com

Pedro Lopes  
University of Chicago  
pedrolopes@uchicago.edu



**Figure 1:** Three objects improved using ThermalRouter: (a) 3D-printed game console slowed due to CPU heat. (b,c) ThermalRouter added thermally-conductive channels to route heat away from the heat source. (d) 3D printed dragon lamp melted at PLA’s heat deflection temperature; ThermalRouter created a version that didn’t buckle. (e) ThermalRouter can also be applied to soft objects like a silicone phone case with embedded liquid metal channels.

## ABSTRACT

Users often 3D model enclosures that interact with significant heat sources, such as electronics or appliances that generate heat (e.g.,

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 the author(s) 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 [permissions@acm.org](mailto:permissions@acm.org).  
UIST '23, October 29–November 01, 2023, San Francisco, CA, USA  
© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.  
ACM ISBN 979-8-4007-0132-0/23/10...\$15.00  
<https://doi.org/10.1145/3586183.3606747>

CPU, motor, lamps, etc.). While parts made by users might function well aesthetically or structurally, *they are rarely thermally-sound*. This happens because heat transfer is non-intuitive; thus, engineering thermal solutions is not straightforward. To tackle this, we developed *ThermalRouter*, a CAD plugin that assists with improving the thermal performance of their models. ThermalRouter automatically converts regions of the model to be made from thermally-conductive materials (such as nylon or metallic-silicone). These regions act as heat channels, branching away from hotspots to dissipate heat. The key is that ThermalRouter automatically simulates the thermal performance of many possible heat channel configurations and presents the user with the most thermally-sound design

(e.g., lowest temperature). Furthermore, it allows users to customize by balancing costs, indicating non-modifiable geometry, etc. Most importantly, ThermalRouter achieves this without requiring manual labor to set up or parse the results of complex thermal simulations.

## CCS CONCEPTS

• **Human-centered computing** → Human computer interaction (HCI); Interaction devices; Haptic devices; • **Hardware** → Emerging technologies.

## KEYWORDS

thermal, heat, engineering, interactive assistant, CAD

### ACM Reference Format:

Alex Mazursky, Borui Li, Shan-Yuan Teng, Daria Shifrina, Joyce E. Passananti, Svitlana Midianko, and Pedro Lopes. 2023. ThermalRouter: Enabling Users to Design Thermally-Sound Devices. In *The 36th Annual ACM Symposium on User Interface Software and Technology (UIST '23), October 29–November 01, 2023, San Francisco, CA, USA*. ACM, New York, NY, USA, 14 pages. <https://doi.org/10.1145/3586183.3606747>

## 1 INTRODUCTION

Advances in fabrication and computer-aided design (CAD) have empowered users to create functional devices, from home media players (e.g., printed enclosures for microcomputers such as the Raspberry Pi), interactive clothing & wearables [10, 21], or even adding 3D printed parts that improve the functionality of existing appliances (e.g., [53]).

Unfortunately, while many 3D printed objects created by users are mechanically functional & aesthetically pleasing, they are often naïve to thermal issues (e.g., heat loosens fits, slows down computation, and even melts plastic) because *heat transfer is not an intuitive phenomenon*. Over the last decades, many learning scientists have researched how people (including children, young students, and college students) develop an intuition for heat transfer (e.g., [9, 16–18, 31, 34, 36, 56], to cite a few). Researchers have found that understanding the fundamentals of heat transfer can be difficult even for university-level engineering students, even after taking physics and thermodynamics courses [54]. One would not expect non-engineers engaged with 3D printing (e.g., some makers & hobbyists) to be an exception to these findings. We also take interest in the challenge of understanding heat transfer; however, our interest is not strictly pedagogical. Instead, we explore how a 3D editor may assist users, not only with creating shapes but also with incorporating thermally-conductive materials and structures to alleviate their thermal issues.

Let us exemplify how even the thermal performance of the 3D printed case shown in Figure 1 (a), which turns a Raspberry Pi into a gaming console, has stark consequences for the interactive device's performance. As revealed in our *Walkthrough*, the 3D printed case traps the CPU's heat and causes its clock speed to slow down after just *one minute* of intense computation (CPUs measure their internal temperature and adjust their clock speed to prevent thermal failures). Unfortunately, designing a custom 3D printed case with *thermal performance in mind* is non-trivial.

Moreover, thermal issues can have other consequences, such as an impact on structural stability or geometric fits. For instance,

heat buildup on regions of a user's model made from plastics will cause the plastic to deform, especially under loads. These are just a few examples of possible thermal issues, which can negatively impact objects that interact with heat sources. The challenge is that while devices engineered by professionals undergo dedicated engineering to address this, users do not have interactive tools that directly support them in authoring *thermally-sound devices*.

Similarly, typical maker-friendly materials exhibit poor thermal performance. The most accessible 3D printing material (PLA) is inherently insulating and exhibits a low melting point—this is normally seen an advantage, as it eases the printing process, but it becomes a severe issue if the final printed object is exposed to heat (e.g., melting). In fact, nearly all plastics have poor thermal conductivity, leading to heat buildup within prints that interact with heat sources. Recently, 3D-printable thermally-conductive plastics have come to market and have been used to develop heatsinks with custom geometries. However, making entire objects from these new materials is costly to users (i.e., typically 10x the cost of PLA). Therefore, there is a need for an approach that allows users to *balance* both cost and performance.

To tackle these issues, we engineered *ThermalRouter*, a CAD plugin that assists in creating thermally-sound devices, as depicted in Figure 1. We implemented ThermalRouter as a plugin to Autodesk's Fusion 360, a popular tool for designing 3D-printed devices. ThermalRouter improves the thermal performance of an object by converting regions of the user's model to be printed with thermally-conductive materials (such as thermally-conductive nylon or even thermally-metallic-silicone, which enables applying our technique to soft models, too). While recent advances in computational design have led to optimal heatsink geometries [38, 57], these require adding additional materials into users' designs which can require additional modeling to accommodate the design changes; in contrast, a benefit of our approach is that we *embed the heat sink directly in the user's model* so no additional material or 3D design is needed.

ThermalRouter's key principle is that it simulates the thermal performance of *many* possible design variations, which are generated not only from its heuristics but also *incorporating the user's input*, such as maximum allowed material costs, regions to avoid, location of heat source(s), and so forth. At each step, ThermalRouter not only presents the user with the most thermally-sound design, but also lets users explore any internally generated variations, which allows users to gain intuition from different thermal designs or explore different aesthetics.

## 2 RELATED WORK

The work presented in this paper builds on the challenges of understanding heat transfer, advances in personal fabrication, especially, 3D modeling software with embedded domain knowledge, and thermally-engineered materials.

### 2.1 Heat transfer is unintuitive

First, to understand why ThermalRouter addresses an important problem, one must establish that heat transfer is not an intuitive phenomenon to grasp. Over the last decades, learning experts have researched how people develop an understanding of heat transfer (see [9, 16–18, 31, 34, 36, 56] to cite a few of these studies) and found

it poses “very significant difficulties [to the learner] in acquiring the notion of temperature” [36]. Luce & Callanan studied conversations about heat and temperature between parents and children (N=51), suggesting that everyday language (words such as “hot” and “cold,” that tend to replace the complex nature of heat transfer, e.g., “heat” and “gradient”) might play a key role in limiting our understanding of heat from an early age [43].

Numerous studies found evidence that heat transfer is non-intuitive across ages. Lewis & Linn found that adolescents, adults, and scientists can display difficulties with heat transfer [36]. Robbins & Ardebel found that engineering students require years of college-level study to develop an intuition on thermal transfer concepts [54]. Their findings (N=101) show that 2<sup>nd</sup> year engineering college students (before taking a thermodynamics course) were only able to correctly express 24% of heat conduction concepts, compared to just 41% after taking their first thermodynamics course.

Altogether, these findings suggest that understanding heat transfer is difficult, even for university-level engineering students. As such, one expects heat transfer to be a challenging phenomenon for many more users that might cross paths with interactive fabrication, such as makers & hobbyists involved with 3D printing and making—a domain in which heat-related challenges naturally arise. To this end, Khounsary et al.’s analysis of heat transfer education denotes that the background required to solve heat-related problems in engineering is a tall order for most users: “a practicing engineer must have a reasonable understanding of design, analyses, materials, fabrication processes, etc., to be able to apply heat transfer know-how to the problem at hand” [31]. Finally, this study specifically warns that “addressing thermal (. . .) issues will become more difficult as the projects proceed along the design path” [31], suggesting that tackling heat should be done early in the process, as well as continuously if possible.

## 2.2 Personal fabrication and the democratization of do-it-yourself (DIY)

Personal fabrication enables end-users to make their own interactive devices [7, 23]. Personal fabrication rocketed with the availability of low-cost CNC tools (e.g., 3D printers, laser cutters, cutting plotters, etc.) aimed at makers & hobbyists. To accelerate this explosion in end-user tools, researchers often develop hardware and software methods for democratizing making and fabrication. Since our work is software-based, we will focus on tools that assist the user by directly embedding domain knowledge inside the computational-aided design software (CAD).

## 2.3 Embedding domain knowledge *inside* CAD Editors

Beyond speeding up the user’s iteration cycles, many researchers have focused on understanding how to lower the knowledge barrier required to manufacture objects or use CAD functionality. This is especially relevant for techniques that build on more advanced know-how and would otherwise require a degree in mechanical engineering or related fields [42, 51]. To this end, many researchers explore how to embed domain-specific knowledge *inside* the CAD environment.

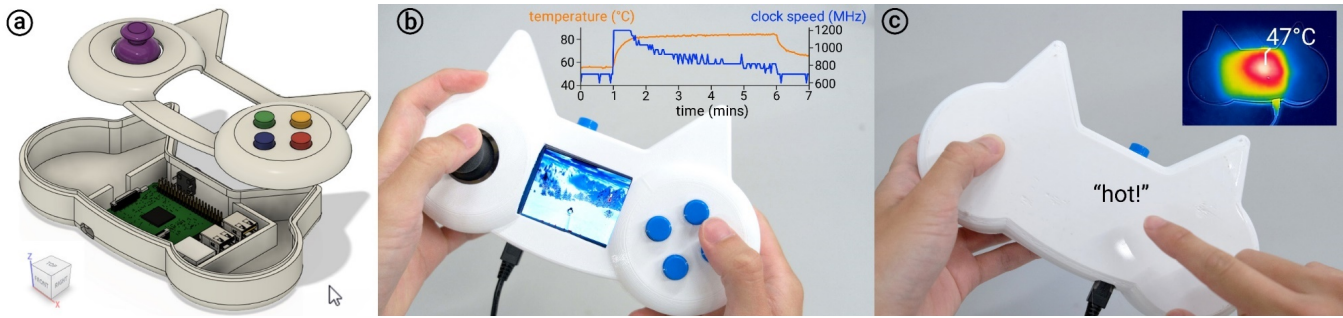
Several recent works have successfully embedded knowledge of structural mechanics into users’ designs to make them structurally sound, i.e., prevent the user’s resulting objects from collapsing under heavier loads. For instance, *ShapeStructuralizer* integrates a recommender system in a wireframe design tool to inform users of alternate designs that satisfy stress constraints [15]. *FastForce* automatically identifies laser-cut structures that would normally break and reinforce them in real-time [1]. *Forte* let users easily sketch out 2D structures, which then served as the basis for generative designs that reinforce and make them structurally viable [13]. *TrussFab* [32] and *TrussFormer* [33] integrated structural & dynamical analysis into an editor, allowing users to create stable large-scale and even animated structures. Moreover, others embed knowledge of materials into editors [26]. *Metamaterial Mechanisms* is an editor with metamaterial cells that can be assembled into machines [29], this is supported by a simulator that assists users while designing these otherwise unintuitive mechanisms. Along similar lines, users benefit from editors with domain knowledge of shape-changing materials [2]. In fact, embedding domain-knowledge directly *inside* the CAD editor is so paramount that many subfields beyond mechanics/3D-printing have been approaching this challenge too, such as magnetics [48], electronics [64], aerodynamics [61], acoustics [60], animation [30], paper crafting [49], and so forth.

While several domains have benefited from embedding domain knowledge into editors, to the best of our knowledge, there is no exploration on how to assist users with their model’s thermal performance. Simultaneously, problems arising from heat sources or poor thermal performance have plagued both students [31] and researchers when designing, for instance, wearable devices [58] or professional machinery with custom heatsinks [57].

## 2.4 Thermally-engineered materials and devices

Materials with high thermal performance have long been studied in material science for their ability to regulate heat flow. While, metals traditionally shined for thermal conductivity (e.g., typical heatsinks added externally to most machines [4, 8]), polymer composites with thermally-conductive fillers have recently gained attention for their low cost and electrical insulation [57]. While the plastics, used in most personal fabrication machines (e.g., laser cutters or 3D printers), have poor thermal conductivities ( $\sim 0.1\text{--}0.5$  W/m-K) [3], they can be improved by blending conductive filler particles in the filament (such as carbon fiber, ceramics, conductive nano-materials, and more [12, 14, 24]). These composites have thermal conductivities orders of magnitude greater than the base plastic ( $\sim 2\text{--}30$  W/m-K). Recently, composites have been tuned specifically for fused filament 3D printers [35, 52]. The recent commercialization of these thermally-enhanced filaments (e.g., the *ICE9* thermal filament [66], which we use to demonstrate some of the tool’s functionality) has brought these materials into maker spaces [57]. However, CAD tools are not directly supporting users with the required domain knowledge to make use of the thermal performance of these thermal filaments.

While 3D printing is still primarily used to fabricate rigid objects, other techniques (e.g., casting or molding) are used to make soft objects (e.g., covers for smartphones or wearables), which must conform to the target shape. To this end, materials like silicone



**Figure 2:** (a) A user models a custom handheld gaming case for their Raspberry Pi. (b) However, after printing and playing a game for a few minutes, they notice their framerate reduce (performance curve generated with *stressberry*, a standard test used by Raspberry Pi community to measure CPU temperature [47]) and (c) that the back of the case has become hot to the touch.

are used for their ability to conform/stretch. Unfortunately, like rigid polymers, silicones have poor thermal conductivity [6]. As with plastics, researchers have doped these soft materials with thermally-conductive particles to improve thermal conductivity [46]. However, rigid fillers result in a material that is less conformable than undoped silicone. To develop a material that is both soft and thermally-conductive, researchers mix silicone with eutectic gallium indium (EGaIn), a liquid metal alloy at room temperature. This enhances the thermal conductivity of silicones by up to 25x [6, 25, 28]. Inspired by these, we use metallic silicone to enable our users to produce soft objects with increased thermal performance (which we call *soft heat channels*).

Finally, in addition to thermally-conductive materials, recent research has shown the benefits of alternative structures for heat transfer, such as computationally optimized and biomimetic/origami heatsinks [22, 38, 41, 44, 55, 62]. We leverage this research on the effectiveness of branching patterns in our own algorithm [37, 39, 63]. However, these methods are not user interfaces like ThermalRouter. These methods are algorithms applied to specific engineering domains. These do not support users and only cater to engineers and researchers that know how to implement them to specific scenarios. Instead, ThermalRouter is the first to assist users with their design’s thermal performance. Therefore, it offers interactive support relevant to makers’ needs (e.g., balance cost of materials, simple fabrication process, etc.).

### 3 WALKTHROUGH

To give the reader a complete picture of how ThermalRouter enables users to design thermally-sound devices, we present an example of a user designing a handheld gaming case for their Raspberry Pi.

We chose this object as our first example for three reasons: (1) it depicts a popular use of personal fabrication (thousands of hobbyists make custom cases like this); yet illustrates how even a simple object can hide serious thermal issues (so drastic these affect the actual CPU performance and comfort); (2) the geometry of this game console is not overly complex, which aids in understanding ThermalRouter’s output—note we show the contrast of ThermalRouter’s output (black thermal filament) with the unmodified (printed in white filament); yet, if the user wishes for a uniform appearance, these could be printed entirely in the same color as the thermal

filament, painted over, or the thermal filament may be hidden inside the object’s shell; and (3) finally, the game console provides an example of an object that lends itself well to iterations, illustrating that ThermalRouter is useful through different phases of a prototype.

At the start, our user creates a version of their cat-shaped gaming case by modeling around existing Raspberry Pi and electronics models in Autodesk Fusion 360, as depicted in Figure 2 (a). However, after printing and assembling their design, they notice that the back of their case is hot to the touch, and the game’s framerate is reduced dramatically after gaming for a few minutes, as shown in Figure 2 (b, c).

**Selecting a problematic heat source.** As the user turns off the console and touches the Raspberry Pi, they realize the heat source is the CPU, which overheats when enclosed or used without a heatsink. Similarly, they could also inspect for heat sources using a thermal camera, as shown in Figure 2 (c) (all thermal images taken with a FLIR C3). In the thermal image, this hotspot is 47°C. To resolve this, our user runs the ThermalRouter plugin within Fusion 360 and indicates that this CPU is heating up to 80°C (note there are several ways a user can obtain a temperature, such as estimates from the Raspberry Pi’s datasheet, with a thermal camera or thermometer, or the CPU’s internal log). To select a heat source in ThermalRouter, users select a 3D part and indicate its temperature or wattage, as depicted in Figure 3.

**Configuring and running ThermalRouter.** Next, ThermalRouter needs to know which of the model’s regions it is allowed to modify by embedding heat channels, which will be printed with thermally-conductive filament. The user selects both the top and bottom of the case to be modified. This freedom is critical as it allows users to (1) run ThermalRouter on subsets of a model (e.g., useful for iterations, large models, etc.) and (2) provide users with control of where ThermalRouter places thermal material (e.g., for aesthetic purposes or to avoid specific areas). Next, the user selects the outer surfaces of the case (those that are in contact with outside air); these inform the simulation of surfaces that naturally convect heat into the environment.

**By selecting “advanced controls,”** users can input (optional) parameters to achieve better solutions, shown in Figure 4. For example, they can: (1) **Adjust the number of variations** that

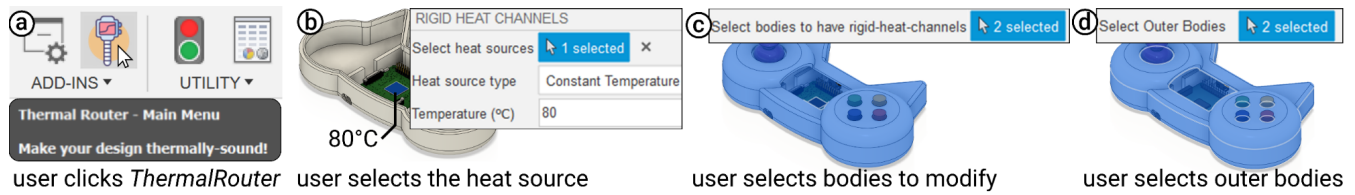


Figure 3: User input pipeline for improving the thermal issues on this Raspberry Pi case using ThermalRouter.

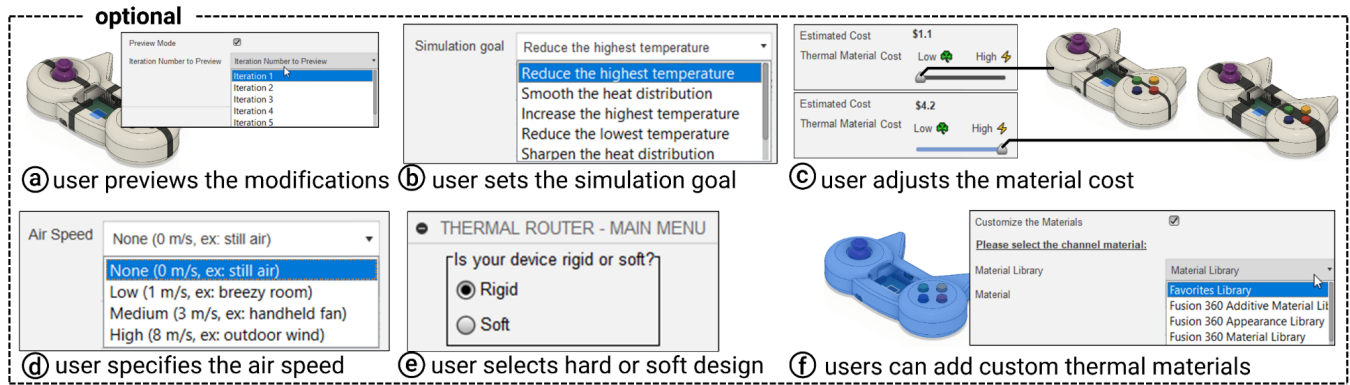


Figure 4: ThermalRouter allows users to provide (optional) input to generate better solutions, such as (a) number of iterations, (b) specific temperature goals, (c) cost of materials, (d) air speed, (e) rigid vs. soft output, and (f) customize the thermal materials.

ThermalRouter generates and preview each one—this saves time by letting the user recognize if they would like to make further changes before running ThermalRouter’s simulation. (2) **Specify ThermalRouter’s goal**, such as reducing the maximum temperature or smoothing the thermal gradient (see *Implementation* for details). (3) **Adjust the expected cost** of the thermal material used via a slider—this adjusts the volume of the thermal material devoted to the heat source—this gives users control over their budget and strikes an appropriate balance between material cost and thermal performance. Moreover, to improve the accuracy of their solutions, users can indicate (4) **whether their objects will interact with circulating air**, which allows ThermalRouter to simulate convection—in this game console example, our user chose “no air” since it is only exposed to still air (later we will depict examples that make use airspeed). Also, the user can choose whether they are (5) **printing a rigid model or creating a mold for a soft shape**—our user chose “rigid” for the console (we depict later how ThermalRouter generates molds automatically for silicone-based models). Finally, if the user plans to use a thermally-conductive material other than our default materials, they can (6) **add custom thermal materials**.

**ThermalRouter generates variations.** After choosing the heat source (and any optional settings), ThermalRouter automatically generates *many variations* (see Figure 5) of this case with regions of the geometry converted to print in a commodity filament with high thermal conductivity. These channels, depicted in a darker color, extend outwards from the heat source to *spread heat*, like conventional heat pipes. Finally, ThermalRouter shows the user the model that performed best in simulation, e.g., the model with the

lowest maximum temperature (see *Implementation* for details). Importantly, ThermalRouter balances cost and thermal performance by only printing specific regions from thermally-conductive filament, rather than the entire object (as previously described, users can tune this balance via our GUI). After simulating, users can select the variation to view from a drop-down list.

**Iterating upon the output of ThermalRouter.** The user inspects the result of ThermalRouter’s simulation of the design that most reduced the CPU’s temperature and is satisfied by the improved thermal performance: *the hot spot is gone*. However, the heat channels also direct heat toward the pause button at the top of the device, which heats to 33°C (as shown in Figure 6 (a)) and will be undesirably warm to the touch. As such, the user navigates backward in their timeline (a feature of modern parametric CAD software like Fusion 360 that captures the step-by-step edit history, allowing to modify or undo operations) and splits the case by sketching a circular boundary around the button. Now, when they navigate forward in the timeline to update the design with ThermalRouter’s changes, these regions are left unmodified by our channel algorithm, thus avoiding the button (as shown in Figure 6, it is now at 27°C). In 3D modeling, there are often alternative ways to achieve the same goal; the same is true for ThermalRouter. In this situation, the user could have also removed the heat channels around the button from the extrusion profiles used by ThermalRouter—this is possible because all geometry generated by ThermalRouter is added to the model’s timeline and is *completely editable*.

**Gaining insights by stepping through ThermalRouter.** The user can review all the solutions created by ThermalRouter (rather

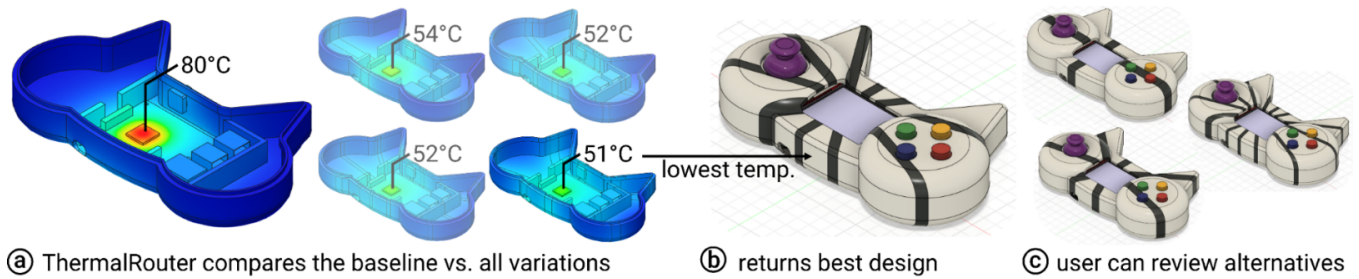


Figure 5: ThermalRouter compares simulations on the baseline and many variations to find the best thermal solution.

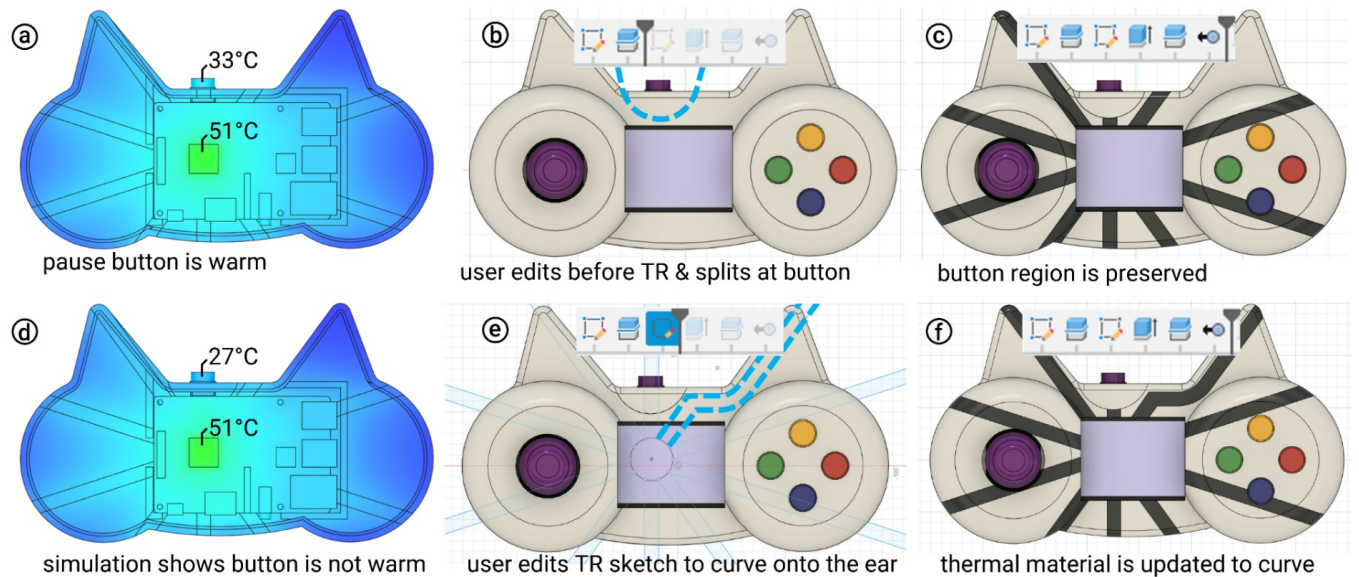
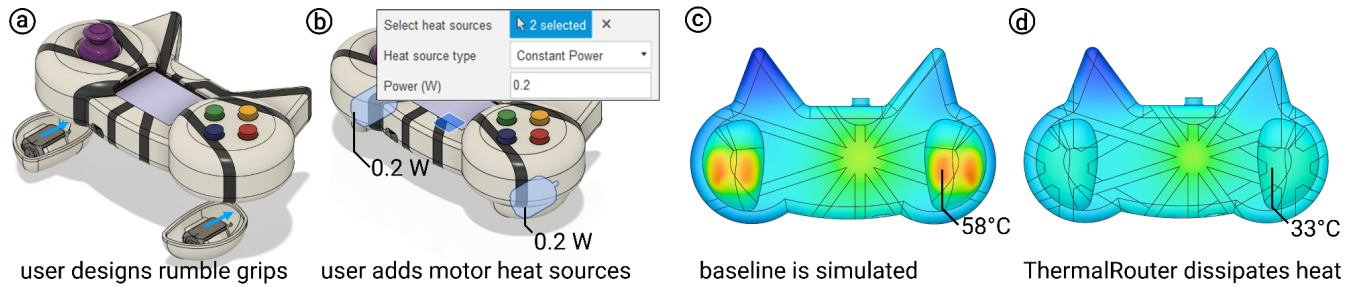


Figure 6: The user keeps iterating on their case to perfect its thermal behavior, this time preventing heating the top button, as well as editing ThermalRouter’s paths directly to influence the design’s aesthetics (ThermalRouter shortened to “TR” for brevity).

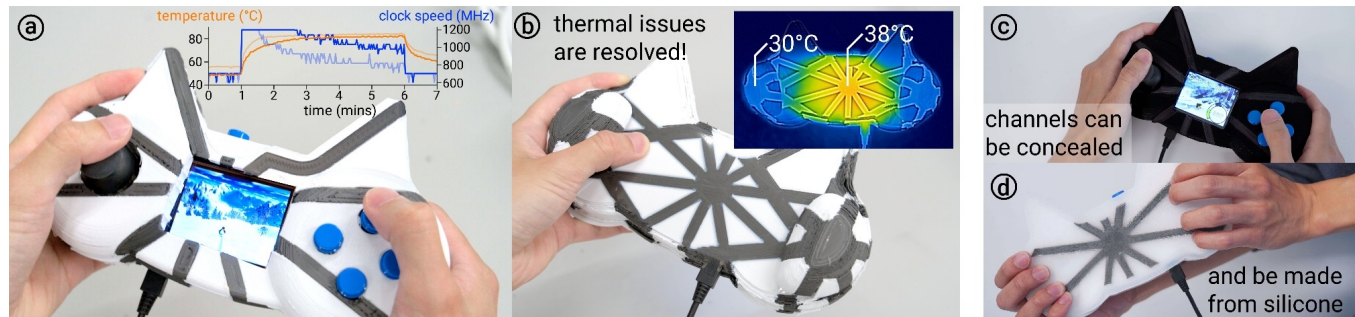
than letting it automatically pick one), which allows users to prioritize other qualities, such as aesthetics. The user can also recover the operations that generated the model and edit the channels’ geometry by accessing ThermalRouter’s sketch. For example, they can adjust the channels so that *both* cat ears have a stripe. To do this, they can simply modify the sketch, and the model is automatically updated, as shown in Figure 6 (e, f).

**Using ThermalRouter at different prototyping phases.** Solving a single thermal problem does not have to be the end of interacting with ThermalRouter. Here, our user continues to build on top of the design already modified by ThermalRouter. They now model grips at the back of the case that contain rumble motors for haptic feedback, as shown in Figure 7. They notice that the motors are also a heat source and use ThermalRouter to address it. They select the heat sources and parts to modify (the motor grips and the back of the case), and ThermalRouter embeds additional heat channels into the model. As always, ThermalRouter simulates the case’s performance, including the existing channels, as well as the new channel variations to cool the motors.

**Fabrication.** Satisfied, the user prints their final design by printing the case from PLA with embedded heat channels from thermally-conductive nylon. As shown in Figure 8, the thermal performance of the case has improved as the CPU idles at lower temperatures and does not throttle as rapidly under extreme stress. Moreover, the motor grips do not overheat on the user’s hands. While we used white PLA to visually contrast the heat channels from standard 3D printing materials, the user has several options to make channels less visible: (1) print the PLA with a color that matches the thermally-conductive filament’s color; (2) hide the channels within the design by thickening the shell surrounding the entire part; (3) post-process the print by painting it. In addition to the rigid design highlighted in this *Walkthrough*, ThermalRouter also supports making thermally-sound **soft and stretchable devices**. For example, the user can choose to add a silicone sleeve to the back of their case that makes it comfortable to grip, but also aids in heat dissipation.



**Figure 7:** (a) The user continues to edit their handheld console design by drawing and attaching grips with vibration motors to attach to the back of the case. (b) The user then runs ThermalRouter again, selects the motors as heat sources, and indicates that the grips and bottom of the case can be modified by the tool. (c) ThermalRouter simulates the baseline performance, which reveals a hotspot forming on the grip. (d) ThermalRouter then presents the user with the design that best reduces the maximum temperature.



**Figure 8:** (a) The user 3D prints their new case designed with ThermalRouter. Compared to the naïve PLA design (transparent curves), the new case yields lower CPU temperatures and higher clock speeds. (b) Similarly, ThermalRouter addressed the issues of heat buildup on the back of the case, as well as on the motor grips. (c) Users can also print their case with black PLA to better hide the thermal channels for aesthetic reasons. (d) ThermalRouter also supports designing thermally-sound soft objects.

#### 4 CONTRIBUTION AND LIMITATIONS

Our key contribution is that we proposed and developed a CAD plugin to assist users with producing thermally-sound devices. We implemented our concept as *ThermalRouter*—which we believe is a first step exploring embedding thermal knowledge in a CAD editor. Using ThermalRouter results in several benefits for users, such as time and money saved through less physical iteration, not to mention more thermally-sound solutions. To further refine the generated solutions, ThermalRouter considers user input, such as the location and temperature of heat sources, the user’s goal (e.g., reduce temperature, smooth heat distribution, etc.), regions that should not be modified, faces exposed to outside air, speed of air, and even expected material cost. Then, ThermalRouter automatically simulates many configurations of their design with embedded heat channels and returns the design that best meets their goals without the need to manually set up complex FEA simulations. Notably, users can also directly edit ThermalRouter’s output because it uses existing CAD operations, unlike prior work that did not center around the user and their UI.

ThermalRouter is not without limitations. While our solutions might outperform a naïve baseline in a range of scenarios, our

solutions are not necessarily optimal (our primary goal was *assisting* users with producing thermally-sound devices, rather than optimality), since these are based on both our heuristics (e.g., creating channels outwards from heat sources) and simulation (e.g., Fusion 360’s FEA-based heat simulation), which merely approximates the real world. Combining our user-centered approach with iterative, goal-seeking algorithms or machine learning may push ThermalRouter into wider territory, such as via the methods used in [19, 38, 41]. However, we want to emphasize that our goal was not to develop optimal solutions; this has been handled exceptionally well in fields like numerical methods and optimization. However, these methods are not in-editor assistants. We explored a *CAD editor* that assists users with designing thermally-sound objects. Moreover, other techniques that drastically alter the object’s form (e.g., generative approaches for custom heatsinks) may also be effective at addressing thermal issues but come with their own challenges of having to remodel around the added heatsink geometry. Instead, we designed ThermalRouter to preserve objects’ geometry, i.e., we only *convert* existing regions to conductive materials, but do not add new geometry. This is beneficial because the original design is virtually unchanged, which minimizes the negative impact on desired form factor (i.e., users do not need to drastically remodel to account for new geometry, and our thermal channels can be

visually hidden by, for instance, burying the channels inside the design by expanding the object’s shell after ThermalRouter). Finally, ThermalRouter addresses heat with passive cooling techniques (i.e., dissipates heat without requiring additional energy) because these lend themselves well to the maker community (no need for additional circuits, coolant fluids, or fans). However, there is a range of more demanding heat systems, such as model engines or gaming rigs, which rely on active cooling and are therefore not addressed by ThermalRouter.

## 5 IMPLEMENTATION

We implemented ThermalRouter as a plugin for Autodesk Fusion 360. This provides ample opportunities for distributing our tool to users engaged with 3D modeling physical prototypes, including other researchers and the maker community—to this end, we provide all our code, test models, and scripts as open source<sup>1</sup>.

**Overview.** To implement ThermalRouter, we used two programming environments: (1) Fusion 360’s Python interpreter, which we used to implement all the algorithms that create the heat channel geometry, run iterative simulations and parse their results; and (2) Fusion 360 API [5], which provided access to the GUI functionality we used to implement ThermalRouter’s user-facing dialogs and allowed access to scripting Fusion 360’s Finite Element Analysis.

**Finite Element Analysis.** Fusion 360 provides Finite Element Analysis (FEA) tools that simulate structural mechanics, as well as thermal simulation. Unfortunately, Fusion 360’s thermal simulation is intended for manual use by experts, i.e., before a user can simulate the thermal performance, they are required to define each part’s material, generate contacts, define thermal loads, and finally mesh the model before they can simulate. Thus, setting up an accurate simulation requires domain knowledge of heat transfer’s non-intuitive mechanisms (assigning appropriate values for conduction, convection, and radiation). Every step introduces opportunities for errors, such as inaccurate boundary conditions, incorrect contacts, or flawed meshes, any of which can lead to unrealistic results, such as infinitely cold/hot parts and even crash the simulation. Debugging these processes can take hours, even for experts.

**ThermalRouter automatically sets up FEA for the user’s model.** Instead, ThermalRouter simplifies this manual work and automatically sets up thermal simulations on the user’s model. ThermalRouter only requires the user to select which parts are heating up and indicate their temperature (°C) or power (W), as well as indicate the faces of parts in contact with the outer air. Then, ThermalRouter executes the remaining setup for simulations in the background, allowing it to perform many simulations to find the best performing design. The user can determine the temperature and power of the heat sources by examining component datasheets. Or, if the user has access to a temperature probe, infrared thermometer, or thermal camera, they can record the heat source temperature during operation.

Now we describe how we implemented ThermalRouter. For clarity, we describe in detail only the key components, while the more trivial components, can be found directly in our open-source code<sup>1</sup>.

### 5.1 Selecting regions of the model that will be processed by ThermalRouter

Now, the user selects the regions in which ThermalRouter will operate, by selecting parts with their cursor. Internally, ThermalRouter uses Fusion 360 API to create this collection of parts. We do so via the `args.command.commandsInput` API, retrieving mouse selections via `addSelectionInput()`, and filtering the selected faces with `addSelectionFilter()`. Using this process, ThermalRouter holds an internal representation of the regions that it will use for thermal improvements.

### 5.2 Routing heat channels

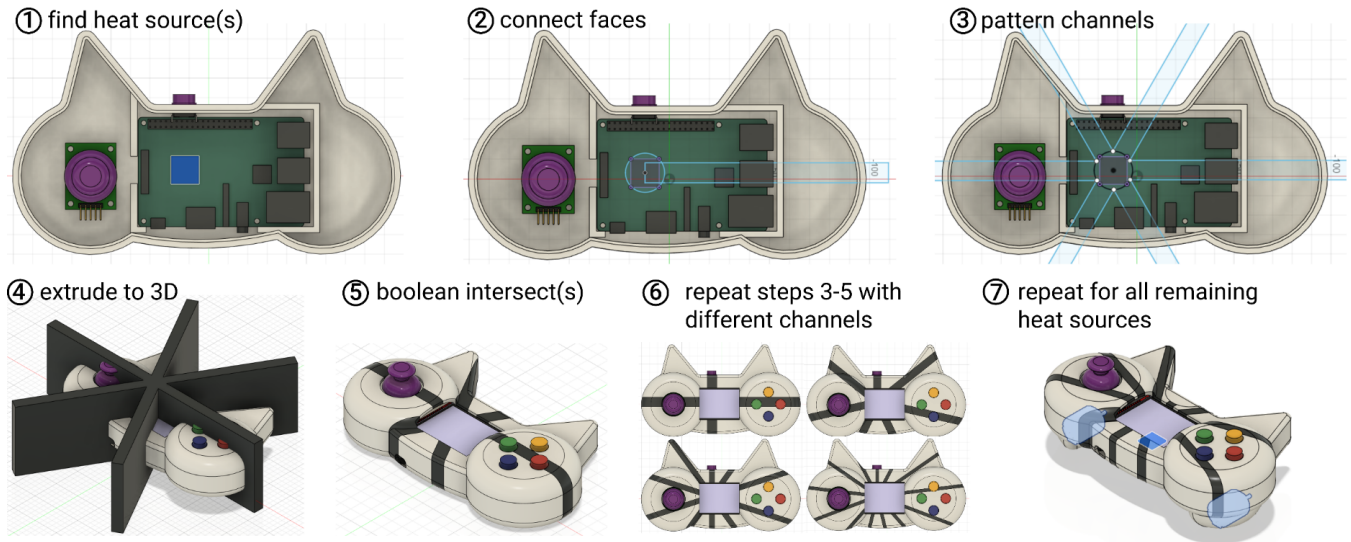
To route heat away from a target source, ThermalRouter uses a straightforward geometrical hub-and-spokes approach (inspired by modern & effective branched heatsinks [37, 39, 50, 63]), which we depict in Figure 9 it directly connects the faces of the user’s model with a 3D shape, representing the shortest path from the heat source to the environment. To realize this, ThermalRouter creates arbitrarily long 3D spokes that branch away from this 3D connection between heat sources and the faces of the model. These spokes are then used as the *splittingTool* to split the parts selected using *splitBodyFeature*. This results in partitioned regions that are converted to thermally-conductive material in both the design and simulations. While this approach that branches heat channels out from source to drain is simple and just one of many possibilities, it is effective for heat transfer and is popularly used to spread heat in electronics [11] and next-generation heat sinks [41].

**Generating variations.** Again, ThermalRouter is not attempting to find a final solution in one shot since we found early on you cannot easily generalize to all models. Instead, it uses straightforward heuristics to create many variations of these spokes, simulates their thermal performance, and judges them against each other. In fact, the width and number of these spokes are the key parameters that ThermalRouter varies across the different iterations as well as the cost that the user is willing to spend (cost corresponds to the channels’ maximum volume). The channel generation process takes only a few seconds per iteration, though simulating each thermal solution is more time consuming (depending on the model’s complexity, ~1 minute for the gaming case shown in *Walkthrough*). Therefore, we allow users to adjust the maximum number of iterations manually (or they can leave it at ThermalRouter’s default of 10 iterations).

The range of channel widths was chosen based on computational & empirical tests. We printed test objects (simple geometries and generalizable heat sources, such as in Figure 10) that let us determine that 2–20mm were the most useful widths—the lower bound was set by how reliably channels could be printed with adequate material, while the upper bound was where we found the area near heat sources (of typical size for a 3D printed solution to work, i.e., up to 30mm) to start to be made entirely from Nylon, which would be both costly & trivial. Therefore, we balance the number of spokes with the width of the channel: as the number of spokes increases, the channel width decreases. This keeps the amount of material and costs across iterations similar. Adjusting the cost parameter also adjusts the baseline channel width.

<sup>1</sup>Code, models, print settings, & guides: <https://lab.plopes.org/#ThermalRouter>





**Figure 9: Main steps involved in our heat channel generation process. For simplicity and visual clarity, we depict it on the example of a single heat source in our Raspberry Pi case model.**

**Assigning material properties.** Before each of ThermalRouter’s variations are simulated, ThermalRouter uses the *TargetBody.material* property to assign different material properties to the regular 3D printed or silicone regions and to the thermally-conductive regions (dark regions in Figure 9). If ThermalRouter is creating rigid heat channels, it assigns the material properties of the ICE9 material; conversely, if ThermalRouter is creating soft heat channels, it assigns the metallic-silicone material properties. These are our defaults, but ThermalRouter can run any other material the user has in their material database if indicated to ThermalRouter in “Advanced options”.

**Estimating material costs.** At this step, ThermalRouter also calculates the cost of the materials by retrieving the volume of the thermal regions and multiplying by their cost per volume. Thus, users are given an estimate of the cost of each solution not only before they spend time fabricating, but also before they spend time running simulations.

### 5.3 Simulating heat transfer

We implemented ThermalRouter to execute as much of the heat simulation as possible without the need for user interaction. We designed ThermalRouter this way since setting up a finite element analysis (FEA) simulation that usefully approximates a real-world heat transfer is not an easy task for users. At the end of any ThermalRouter process users can still open Fusion 360’s simulation panel and browse through the results of each iteration simulated by ThermalRouter to gain insights.

**Heat sources.** The first step of our automation for setting up a thermal simulation is defining the heat sources: (1) ThermalRouter iterates over all objects in the CAD scene: if a part was marked by the user as a heat source, it is converted to a *thermal load* in the simulation; (2) for the volume of each heat source, ThermalRouter also adds its relative power by converting the temperature provided

by the user (e.g., 80°C in our Raspberry Pi CPU in the *Walkthrough*) to watts using the specific heat formula; and (3) if the user specified the room temperature in “advanced options,” ThermalRouter sets this as the simulation’s *environmental temperature*, otherwise it assumes 20°C as a default room temperature.

**Non-heat sources.** ThermalRouter also automates the setup for all non-heat source objects in the CAD scene, as follows: (1) iterates over all objects and sets the *material properties* they have been assigned—materials might be specified by the user or from defaults (e.g., PLA is default) or by the output of ThermalRouter itself (e.g., rigid heat channels default to ICE9 Nylon, while soft heat channels default to a 50-50 volume mixture of silicone and EGAIn; both of these materials which we detail in our *Material Evaluation* can also be specified by the user to any other material if present in the material library); (2) assigns to each object’s surface a *radiation thermal load* that matches the environmental temperature, i.e., in our simulations, non-heated objects start at thermal equilibrium with their surroundings. (3) ThermalRouter sets the *emissivity coefficient* for all its custom materials (ICE9 Nylon, silicone, and metallic-silicone) at 0.9, as it is typical for plastics and rubbers [27, 45]; (4) now to properly simulate a more realistic heat transfer where the object is also surrounded by air, ThermalRouter adds *natural convection thermal loads* (i.e., heat transfer into still air at previous room temperature) to all surfaces with a convection coefficients at 2.50 W/m<sup>2</sup>K, an approximation of convection in still air [40]; (5) finally, after assigning these thermal loads to all objects, ThermalRouter ensures contact between all surfaces separated by less than 0.10 mm, enabling conduction between material regions that are touching.

**Materials settings.** We introduced all the material parameters needed to simulate ThermalRouter’s techniques. Much like all our code and tools, we provide these as an open-source material library<sup>1</sup>.

**Table 1: Material properties of our common and thermally-conductive materials used in ThermalRouter’s simulations.**

Material	Density (kg/mm <sup>3</sup> )	Thermal conductivity (w/m•K)	Specific heat (J/kg•K)
PLA (polylactic acid) [59]	$1.25 \times 10^{-06}$	0.13	1800
ICE9 (nylon-based) [66]	$1.45 \times 10^{-06}$	4	1100
silicone [20]	$1.07 \times 10^{-06}$	0.2	1200
metallic-silicone [6]	$3.66 \times 10^{-06}$	4.7	750

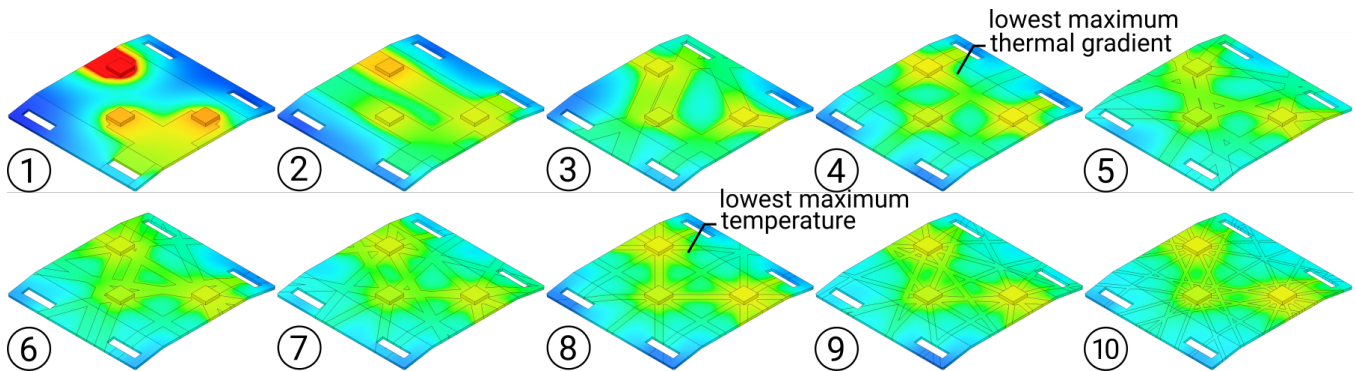

**Figure 10: ThermalRouter generates and simulates many candidate versions of designs for each solution, then presents the user with the design that best matches their chosen goal (e.g., lowest maximum temperature or smoothest thermal gradient).**

Table 1 summarizes the key material settings we used in this paper, obtained from manufacturer’s datasheets and the literature.

#### 5.4 Picking the best result of a simulation

Now that Thermal Router generated many solutions for the user’s model and has simulated each solution’s thermal performance, it picks the best of these depending on the user’s goal. We provide users with five methods to pick thermal simulations: **(1) reduce the highest temperature** picks the solution with the lowest maximum temperature—this is our **default**, unless the user specifies differently in the “advanced” settings; **(2) smooth the heat distribution** picks the simulation result that has the lowest thermal gradient, i.e., where there is low temperature change from one region to another—this is useful, for instance, for smoothing hot spots for wearables (e.g., Figure 10); **(3) increase the highest temperature**—useful in designs that try to maximize heat at a spot, e.g., a heating device using resistive heaters; **(4) reduce the minimum temperature**—e.g., a cooling device using Peltiers; and **(5) sharpen the heat distribution**, which maximizes gradients—e.g., an insulated cup. To compare the simulation results, ThermalRouter parses each vertex of the FEA simulation output file to extract the temperature and magnitude of the thermal gradient at that vertex. It randomly samples 10,000 temperature points (for speed, since thermal simulation data can be extremely large, up to a million samples). Then, since Fusion 360’s FEA simulation operates on coarse meshes to optimize for computation speed, it also often produces some outliers, i.e., points where the temperature is extremely high or low (usually physically impossible temperatures). To filter these outliers, ThermalRouter discards the hottest and the coolest 1000 points. The remainder 8,000 points are then sorted by increasing

value. Finally, it ranks solutions by comparing the highest/lowest temperatures/gradients, as specified by the user. To illustrate this, Figure 10 shows ten simulations on the baseplate of a wearable device with three heat sources. As the user chose “smooth heat distribution”, ThermalRouter picked *variation #4*, which presents the most homogenous heat distribution, often desirable in wearables that touch the skin. Instead, had the user stuck to the default of “reducing the highest temperature” ThermalRouter would have picked *variation #8*, which presents the lowest peak temperature—this might be ideal for most cases, but perhaps not for this wearable device.

#### 5.5 Automatically generating silicone molds (for soft objects)

If the user is fabricating a soft object (e.g., from silicone), ThermalRouter will not only add these conductive regions to the model (same process as for the rigid channels) but also automatically generate and output the 3D print files for *molds*. As depicted in Figure 11, users fill these molds with silicone and metallic-silicone.

**Mold generation.** For each part that requires a mold, the following is performed to generate the mold automatically. First, the user is prompted to select the open face of the mold. This region will be left open so that silicone may be cast directly inside. Then, ThermalRouter creates a copy of each silicone part; this copy is then made 0.5 mm thinner by applying *ExtrudeFeature* with the open face on the copied part as input, followed by a *CutOperation*. This step is necessary to produce a 0.5 mm layer of silicone onto which the metallic-silicone may be cast, as depicted in Figure 11. Next, the copy is split using the same method described in Section 5.2; however, after performing the split, the regions which are not

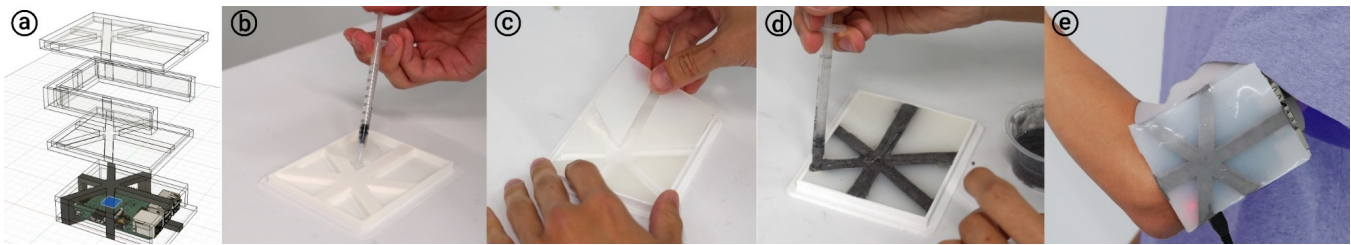


Figure 11: (a) When working with 3D models to be cast in soft materials, such as silicone, ThermalRouter generates all the molds. (b-e) Users may then print the molds, cast with silicone, and glue into soft assemblies.

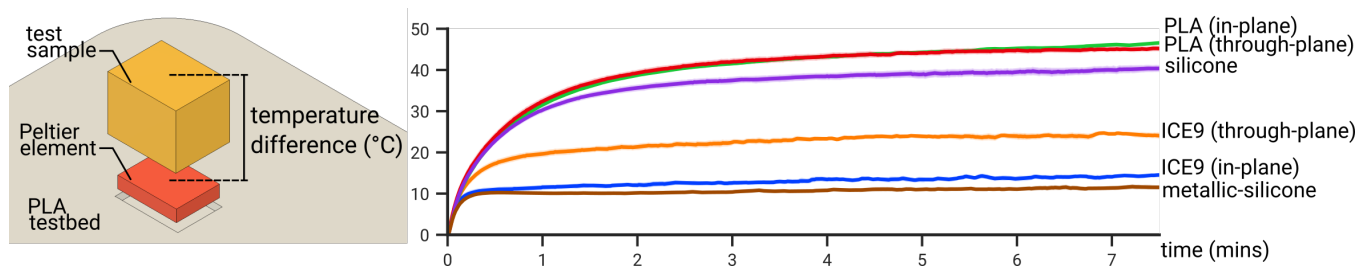


Figure 12: We characterized the thermal performance of the materials used in all examples of ThermalRouter.

made from metallic-silicone are removed. Then, a shell is created from the part’s geometry by calling *ShellFeatures* with the open face as input. The shell’s thickness is set to 2 mm, which is the mold’s wall thickness. Finally, the shell is combined with the copied part to form the mold, which is then moved away from the model itself using *MoveFeatures*.

## 6 MATERIAL EVALUATION

**Objective.** Before we test the effectiveness of ThermalRouter’s solutions, we first characterize the materials employed by our tool. To do so, we measured the relative thermal conductivities of each material.

**Apparatus.** To realize this, we built a hot-bed apparatus, depicted in Figure 12, using a Peltier element (CM23-1.9-05AC) and two temperature probes (100 kOhm NTC 3950). This apparatus allows us to place a material sample in contact with the heating element and measure how quickly the heat conducts through it. This yields a proxy of the *thermal conductivity* of each sample. We fabricated a 9.5 x 7.5 x 8 mm block of each material: PLA (in-plane & through-plane), silicone (Dragonskin FX pro), metallic-silicone (50 vol% EGaIn), and ICE9 Nylon (in-plane & through-plane).

**Procedure.** The Peltier was heated at constant voltage (2.6 V) for 7.5 minutes while measuring the temperature difference between the Peltier surface and the top of the sample (two trials per sample, averaged). Temperature difference started at 0°C as the experiment began at room temperature (both faces were at the same temperature). Over time, a temperature difference arose corresponding to the material’s conductivity as one face was heated.

**Results.** Figure 12 depicts our findings. We found that the base materials, PLA, and silicone, have low thermal conductivities (i.e., higher temperature difference in the plot, as they do not conduct heat well through the sample), which validates their thermal issues.

Conversely, materials such as ICE9 Nylon and metallic-silicone, have greater thermal conductivities (i.e., lower temperature difference in the plot, as they conduct heat better through the sample). This characterization validated the material assumptions upon which we later built our tool and tuned our simulations: simulating the hot-bed apparatus gave **temperature differences within 5%** for each material (note that Fusion 360 is limited to isotropic materials, thus our ICE9 material used the average thermal conductivity of the in-plane and through-plane conductivities).

## 7 ADDITIONAL EXEMPLARY USE CASES FOR THERMALROUTER

After depicting ThermalRouter with the example of the handheld game console, we now provide a broader illustration of its applicability. For each example object, we depict its baseline and the best solution generated and picked by ThermalRouter, alongside each object under our thermal camera.

**Structural stability affected by heat sources: a 3D printed lamp.** In Figure 13, we show a 3D-printed lamp featuring a fire-breathing dragon (a popular Thingiverse model [65]), with its 30W incandescent bulb placed within the flame, giving it a glowing aura. Unfortunately, after just 5 minutes, the heat produced by the bulb causes the lamp’s plastic to melt and crumple under the dragon’s weight. Instead, in the version generated by ThermalRouter, the channels help to dissipate heat into the environment and provide additional resistance to heat-induced deflection, allowing the lamp to withstand high temperatures for longer than a 24-hour stress test.

**Precise geometric fits are affected by heat: a 3D printed motor holder.** In Figure 14, we depict a 3D printed fixture designed to hold a DC motor for a custom fan. Unfortunately, the heat buildup as the motor spins causes the plastic to soften, undermining

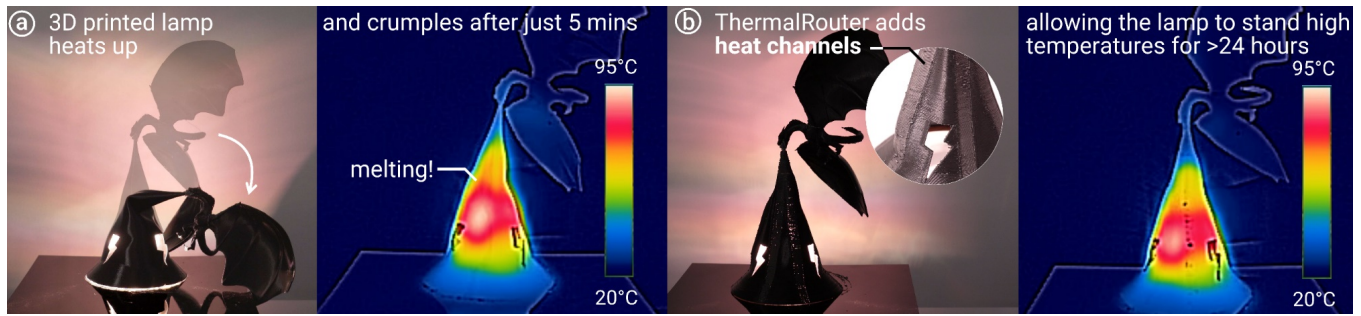


Figure 13: (a) A 3D printed lamp (lightbulb inside) features a dragon breathing fire; however, the heat generated by the lightbulb causes plastic to deform and crumple under the weight of the dragon after just 5 minutes of being on. (b) ThermalRouter generates a solution that enables trapped heat to escape to the environment via heat channels, allowing the lamp to operate without failures. By printing with black PLA, the channels can be concealed to preserve the lamp’s aesthetics.

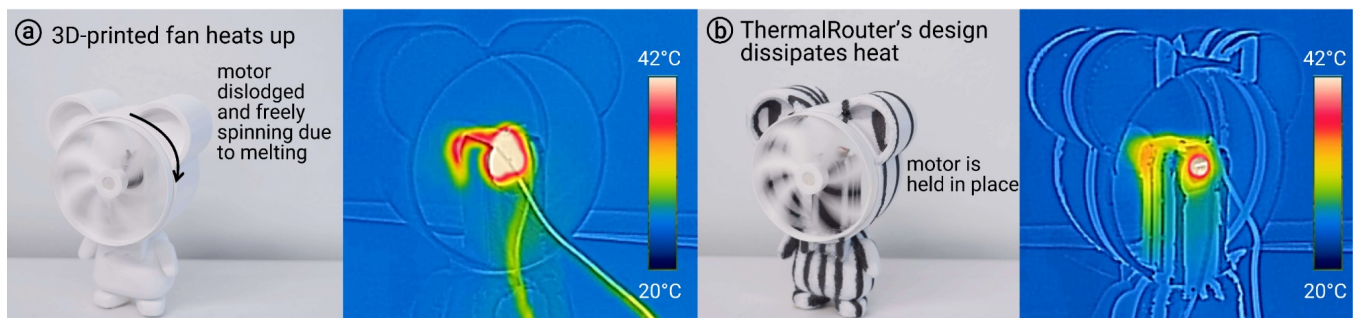


Figure 14: (a) As the fan’s motor turns, its excess heat builds up at the fixture, causing the plastic to soften and loosen the fit. (b) ThermalRouter produces a solution that addresses excess heat by routing it away from the source via heat channels.

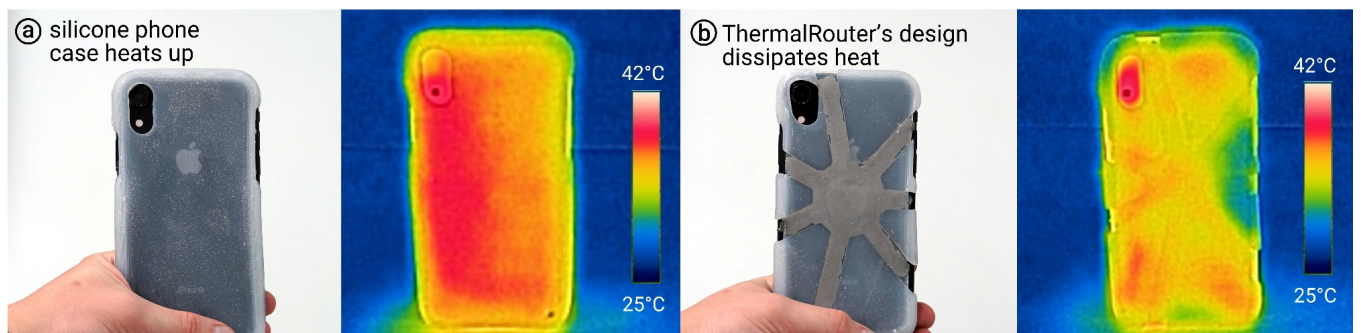
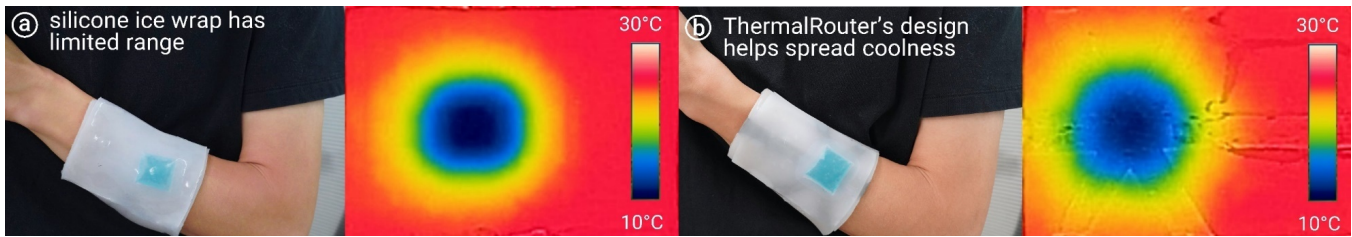


Figure 15: (a) A user notices that when gaming, their phone is warm to the touch because of its silicone case. (b) They use ThermalRouter’s soft channels to let their case act as a heat sink that dissipates excess heat to the air, thus cooling down the phone.

the structural stability of the fit and causing the motor to dislodge after just 7 minutes. Instead, the version made with ThermalRouter can dissipate more heat away from the motor via conduction, thus preventing plastic deformation. Moreover, in this example, ThermalRouter considered the user-defined speed of the fan’s air, allowing it to use even less thermal material for a thermally-sound performance.

**Soft objects also interact with heat sources: a silicone phone cover.** In Figure 15, we depict a situation where the *soft heat channels* proved useful. The user custom-made a silicone sleeve for their smartphone, which heats up at its center while gaming or using GPS navigation. As depicted in Figure 15, ThermalRouter’s soft heat channels route heat away from the hot spot, allowing the user’s phone to feel less hot to the touch.



**Figure 16:** (a) A user makes a silicone sleeve for their ice pack. However, using silicone as a buffer insulates the user's arm from cooling down well. (b) A user instead uses ThermalRouter to generate soft heat channels that better distribute the cool sensation.

**Cold can also be a thermal design goal: a wearable ice pack.** Figure 16 depicts a wearable patch designed to help the user cool down their arm with an ice pack. For comfort, users typically add a buffer between their skin and the ice pack (e.g., a towel). Instead, this user uses ThermalRouter to design a silicone patch that spreads out the cooling from the ice pack to a larger area. This unique example makes use of ThermalRouter to route *cold* (via the absence of heat at the pack creating a thermal gradient along the channels). The thermal images in Figure 16 confirm that the ice pack's cold spot spreads more across the user's skin after being applied for 10 minutes.

## 8 CONCLUSION

In this paper, we presented ThermalRouter, an interactive design tool that assists users with making thermally-sound devices. ThermalRouter embeds some of the required heat transfer domain-knowledge directly into the CAD software, unlocking thermal design for users. We designed ThermalRouter to tackle thermal issues and confirmed the improved thermal performance of objects created with this technique. Our approach covers a wide range of use cases, from traditional 3D printing to casting soft objects made from silicone. We hope this variety will foster creativity and use by both researchers & maker community, who we tend to think as some of the prospective users of our tool.

For future work, we believe that researchers might extend ThermalRouter, which we made entirely open-source, by integrating more advanced heuristics that generate more complex variations, or even more advanced algorithms to allow users to design for more complex thermal situations.

## ACKNOWLEDGMENTS

This work was partially supported by the University of Chicago Materials Research Science and Engineering Center, which is funded by the National Science Foundation under award number DMR-2011854. Moreover, we are grateful for the constructive feedback from the reviewers.

## REFERENCES

- [1] Abdullah, M. *et al.* 2021. FastForce: Real-Time Reinforcement of Laser-Cut Structures. *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama Japan, May 2021), 1–12.
- [2] An, B. *et al.* 2018. Thermorph: Democratizing 4D Printing of Self-Folding Materials and Interfaces. *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18* (Montreal QC, Canada, 2018), 1–12.
- [3] Anderson, D.R. 1966. Thermal Conductivity of Polymers. *Chemical Reviews*. 66, 6 (Oct. 1966), 677–690. DOI:<https://doi.org/10.1021/cr60244a004>.
- [4] Armstrong, J. and Winstanley, D. 1988. A Review of Staggered Array Pin Fin Heat Transfer for Turbine Cooling Applications. *Journal of Turbomachinery*. 110, 1 (Jan. 1988), 94–103. DOI:<https://doi.org/10.1115/1.3262173>.
- [5] Autodesk Fusion 360 API: <https://autodeskfusion360.github.io/>. Accessed: 2021-09-08.
- [6] Bartlett, M.D. *et al.* 2017. High thermal conductivity in soft elastomers with elongated liquid metal inclusions. *Proceedings of the National Academy of Sciences*. 114, 9 (Feb. 2017), 2143–2148. DOI:<https://doi.org/10.1073/pnas.1616377114>.
- [7] Baudisch, P. and Mueller, S. 2017. Personal Fabrication. *Foundations and Trends® in Human-Computer Interaction*. 10, 3–4 (2017), 165–293. DOI:<https://doi.org/10.1561/1100000055>.
- [8] Bejan, A. *et al.* 1996. *Thermal design and optimization*. Wiley.
- [9] Brookes, D.T. and Etkina, E. 2015. The Importance of Language in Students' Reasoning About Heat in Thermodynamic Processes. *International Journal of Science Education*. 37, 5–6 (Apr. 2015), 759–779. DOI:<https://doi.org/10.1080/09500693.2015.1025246>.
- [10] Buechley, L. and Eisenberg, M. 2009. Fabric PCBs, electronic sequins, and socket buttons: techniques for e-textile craft. *Personal and Ubiquitous Computing*. 13, 2 (Feb. 2009), 133–150. DOI:<https://doi.org/10.1007/s00779-007-0181-0>.
- [11] Cai, Y. *et al.* 2017. Experimental investigation on a novel multi-branch heat pipe for multi-heat source electronics. *International Journal of Heat and Mass Transfer*. 104, (Jan. 2017), 467–477. DOI:<https://doi.org/10.1016/j.ijheatmasstransfer.2016.08.080>.
- [12] Chen, H. *et al.* 2016. Thermal conductivity of polymer-based composites: Fundamentals and applications. *Progress in Polymer Science*. 59, (Aug. 2016), 41–85. DOI:<https://doi.org/10.1016/j.progpolymsci.2016.03.001>.
- [13] Chen, X. "Anthony" *et al.* 2018. Forte: User-Driven Generative Design. *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC Canada, Apr. 2018), 1–12.
- [14] Chen, Y.-M. and Ting, J.-M. 2002. Ultra high thermal conductivity polymer composites. *Carbon*. 40, 3 (Mar. 2002), 359–362. DOI:[https://doi.org/10.1016/S0008-6223\(01\)00112-9](https://doi.org/10.1016/S0008-6223(01)00112-9).
- [15] Chidambaram, S. *et al.* 2019. Shape Structuralizer: Design, Fabrication, and User-driven Iterative Refinement of 3D Mesh Models. *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow Scotland Uk, May 2019), 1–12.
- [16] Clark, D. and Jorde, D. 2004. Helping students revise disruptive experientially supported ideas about thermodynamics: Computer visualizations and tactile models. *Journal of Research in Science Teaching*. 41, 1 (Jan. 2004), 1–23. DOI:<https://doi.org/10.1002/tea.10097>.
- [17] Clark, D.B. 2006. Longitudinal Conceptual Change in Students' Understanding of Thermal Equilibrium: An Examination of the Process of Conceptual Restructuring. *Cognition and Instruction*. 24, 4 (Aug. 2006), 467–563. DOI:[https://doi.org/10.1207/s1532690xci2404\\_3](https://doi.org/10.1207/s1532690xci2404_3).
- [18] Clough, E.E. and Driver, R. 1985. Secondary students' conceptions of the conduction of heat: bringing together scientific and personal views. *Physics Education*. 20, 4 (Jul. 1985), 176–182. DOI:<https://doi.org/10.1088/0031-9120/20/4/309>.
- [19] Dede, E.M. 2009. Multiphysics Topology Optimization of Heat Transfer and Fluid Flow Systems. *Excerpt from the Proceedings of the COMSOL Conference 2009 Boston* (2009), 7.
- [20] Dragon Skin™ FX-Pro™ Product Information: <https://www.smooth-on.com/products/dragon-skin-fx-pro/>. Accessed: 2021-09-08.
- [21] Edwards, C. 2013. Not-so-humble Raspberry Pi gets big ideas. *Engineering & Technology*.
- [22] Felber, R.A. *et al.* 2016. Design and Modeling of 3D-Printed Air-Cooled Heat Exchangers. (2016), 9.
- [23] Gershenfeld, N.A. 2007. *Fab: the coming revolution on your desktop - from personal computers to personal fabrication*. Basic Books.

- [24] Ha, S.M. et al. 2013. Thermal conductivity of graphite filled liquid crystal polymer composites and theoretical predictions. *Composites Science and Technology*. 88, (Nov. 2013), 113–119. DOI:<https://doi.org/10.1016/j.compscitech.2013.08.022>.
- [25] Haque, A.B.M.T. et al. 2020. Programmable Liquid Metal Microstructures for Multifunctional Soft Thermal Composites. *Advanced Functional Materials*. (Apr. 2020), 2000832. DOI:<https://doi.org/10.1002/adfm.202000832>.
- [26] He, L. et al. 2019. Ondulé: Designing and Controlling 3D Printable Springs. *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans LA USA, Oct. 2019), 739–750.
- [27] Hong, S. et al. 2020. An Adaptive and Wearable Thermal Camouflage Device. *Advanced Functional Materials*. 30, 11 (Mar. 2020), 1909788. DOI:<https://doi.org/10.1002/adfm.201909788>.
- [28] Huang, X. et al. 2020. Soft Thermal Actuators with Embedded Liquid Metal Microdroplets for Improved Heat Management. *2020 3rd IEEE International Conference on Soft Robotics (RoboSoft)* (New Haven, CT, USA, May 2020), 367–372.
- [29] Ion, A. et al. 2016. Metamaterial Mechanisms. *Proceedings of the 29th Annual Symposium on User Interface Software and Technology* (Tokyo Japan, Oct. 2016), 529–539.
- [30] Kazi, R.H. et al. 2016. ChronoFab: Fabricating Motion. *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose California USA, May 2016), 908–918.
- [31] Khounsary, A.M. et al. 1998. Heat Transfer Education: Keeping it Relevant and Vibrant. *Heat Transfer: Volume 3 – Application of Heat Transfer in Equipment, Systems, and Education* (Anaheim, California, USA, Nov. 1998), 17–23.
- [32] Kovacs, R. et al. 2017. TrussFab: Fabricating Sturdy Large-Scale Structures on Desktop 3D Printers. *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver Colorado USA, May 2017), 2606–2616.
- [33] Kovacs, R. et al. 2018. TrussFormer: 3D Printing Large Kinetic Structures. *The 31st Annual ACM Symposium on User Interface Software and Technology - UIST '18* (Berlin, Germany, 2018), 113–125.
- [34] Lautrey, J. and Mazens, K. 2004. Is children's naive knowledge consistent? A comparison of the concepts of sound and heat. *Learning and Instruction*. 14, 4 (Aug. 2004), 399–423. DOI:<https://doi.org/10.1016/j.learninstruc.2004.06.011>.
- [35] Lazarov, B.S. et al. 2018. Experimental validation of additively manufactured optimized shapes for passive cooling. *Applied Energy*. 226, (Sep. 2018), 330–339. DOI:<https://doi.org/10.1016/j.apenergy.2018.05.106>.
- [36] Lewis, E.L. and Linn, M.C. 1994. Heat energy and temperature concepts of adolescents, adults, and experts: Implications for curricular improvements. *Journal of Research in Science Teaching*. 31, 6 (Aug. 1994), 657–677. DOI:<https://doi.org/10.1002/tea.3660310607>.
- [37] Li, B. et al. 2017. Constructal design of internal cooling geometries in heat conduction system using the optimality of natural branching structures. *International Journal of Thermal Sciences*. 115, (May 2017), 16–28. DOI:<https://doi.org/10.1016/j.ijthermalsci.2017.01.007>.
- [38] Li, B. et al. 2016. Generating optimal topologies for heat conduction by heat flow paths identification. *International Communications in Heat and Mass Transfer*. 75, (Jul. 2016), 177–182. DOI:<https://doi.org/10.1016/j.icheatmasstransfer.2016.04.016>.
- [39] Li, Y. and Feng, M. 2021. Optimal design of conductive natural branched pathways for cooling a heat-generating volume. *Heat Transfer*. 50, 3 (May 2021), 2571–2591. DOI:<https://doi.org/10.1002/htj.21993>.
- [40] Lienhard, J.H. and Lienhard, J.H. 2019. *A heat transfer textbook*. Dover Publications, Inc.
- [41] Lohan, D.J. et al. 2017. Topology optimization for heat conduction using generative design algorithms. *Structural and Multidisciplinary Optimization*. 55, 3 (Mar. 2017), 1063–1077. DOI:<https://doi.org/10.1007/s00158-016-1563-6>.
- [42] Lu, L. et al. 2014. Build-to-last: strength to weight 3D printed objects. *ACM Transactions on Graphics*. 33, 4 (Jul. 2014), 1–10. DOI:<https://doi.org/10.1145/2601097.2601168>.
- [43] Luce, M.R. and Callanan, M.A. 2020. Family Conversations About Heat and Temperature: Implications for Children's Learning. *Frontiers in Psychology*. 11, (Aug. 2020), 1718. DOI:<https://doi.org/10.3389/fpsyg.2020.01718>.
- [44] Mishra, A.K. et al. 2020. Autonomic perspiration in 3D-printed hydrogel actuators. *Science Robotics*. 5, 38 (Jan. 2020), eaz3918. DOI:<https://doi.org/10.1126/scirobotics.aaz3918>.
- [45] Morgan, R.V. et al. 2017. *Emissivity Measurements of Additively Manufactured Materials*. Technical Report #LA-UR-7-20513, 1341825.
- [46] Mu, Q. et al. 2007. Thermal conductivity of silicone rubber filled with ZnO. *Polymer Composites*. 28, 2 (Apr. 2007), 125–130. DOI:<https://doi.org/10.1002/pc.20276>.
- [47] nschloe/stressberry: Stress tests for the Raspberry Pi: <https://github.com/nschloe/stressberry>. Accessed: 2021-09-08.
- [48] Ogata, M. and Koyama, Y. 2021. A Computational Approach to Magnetic Force Feedback Design. *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama Japan, May 2021), 1–12.
- [49] Oh, H. et al. 2018. PEP (3D Printed Electronic Papercrafts): An Integrated Approach for 3D Sculpting Paper-Based Electronic Devices. *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC Canada, Apr. 2018), 1–12.
- [50] Park, K.T. et al. 2014. Experimental study of natural convection from vertical cylinders with branched fins. *Experimental Thermal and Fluid Science*. 54, (Apr. 2014), 29–37. DOI:<https://doi.org/10.1016/j.expthermflusci.2014.01.018>.
- [51] Prévost, R. et al. 2013. Make it stand: balancing shapes for 3D fabrication. *ACM Transactions on Graphics*. 32, 4 (Jul. 2013), 1–10. DOI:<https://doi.org/10.1145/2461912.2461957>.
- [52] Quill, T.J. et al. 2018. Thermal and mechanical properties of 3D printed boron nitride – ABS composites. *Applied Composite Materials*. 25, 5 (Oct. 2018), 1205–1217. DOI:<https://doi.org/10.1007/s10443-017-9661-1>.
- [53] Ramakers, R. et al. 2016. RetroFab: A Design Tool for Retrofitting Physical Interfaces using Actuators, Sensors and 3D Printing. *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose California USA, May 2016), 409–419.
- [54] Robbins, D. and Ardebili, M. 2006. Engineering Students' Conceptions Of Heat And Temperature Pre And Post Thermodynamics Course. *2006 Annual Conference & Exposition Proceedings* (Chicago, Illinois, Jun. 2006), 11.578.1-11.578.9.
- [55] Rösch, A.G. et al. 2021. Fully printed origami thermoelectric generators for energy-harvesting. *npj Flexible Electronics*. 5, 1 (Dec. 2021), 1. DOI:<https://doi.org/10.1038/s41528-020-00098-1>.
- [56] Shayer, M. and Wylam, H. 1981. The development of the concepts of heat and temperature in 10-13 year-olds. *Journal of Research in Science Teaching*. 18, 5 (Sep. 1981), 419–434. DOI:<https://doi.org/10.1002/tea.3660180506>.
- [57] Smith, M. et al. 2019. Maximizing the Performance of a 3D Printed Heat Sink by Accounting for Anisotropic Thermal Conductivity During Filament Deposition. *2019 18th IEEE Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems (ITherm)* (Las Vegas, NV, USA, May 2019), 626–632.
- [58] Starner, T. and Maguire, Y. 1999. Heat dissipation in wearable computers aided by thermal coupling with the user. *Mobile Networks and Applications*. 4, 1 (1999), 3–13. DOI:<https://doi.org/10.1023/A:1019113924178>.
- [59] Ultimaker PLA material: Highly versatile, easy to print: <https://ultimaker.com/materials/pla>. Accessed: 2021-09-08.
- [60] Umetani, N. et al. 2016. Printone: interactive resonance simulation for free-form print-wind instrument design. *ACM Transactions on Graphics*. 35, 6 (Nov. 2016), 1–14. DOI:<https://doi.org/10.1145/2980179.2980250>.
- [61] Umetani, N. et al. 2014. Pteromys: interactive design and optimization of free-formed free-flight model airplanes. *ACM Transactions on Graphics*. 33, 4 (Jul. 2014), 1–10. DOI:<https://doi.org/10.1145/2601097.2601129>.
- [62] Wang, X. and Jiang, H. 2018. Design of origami fin for heat dissipation enhancement. *Applied Thermal Engineering*. 145, (Dec. 2018), 674–684. DOI:<https://doi.org/10.1016/j.applthermaleng.2018.09.079>.
- [63] Xu, P. et al. 2006. Heat conduction in fractal tree-like branched networks. *International Journal of Heat and Mass Transfer*. 49, 19–20 (Sep. 2006), 3746–3751. DOI:<https://doi.org/10.1016/j.ijheatmasstransfer.2006.01.033>.
- [64] Zhu, J. et al. 2020. MorphSensor: A 3D Electronic Design Tool for Reforming Sensor Modules. *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology* (Virtual Event USA, Oct. 2020), 541–553.
- [65] Game of Thrones - Drogon fire dragon lamp. <https://www.thingiverse.com/thing:3648004>.
- [66] Ice9 Nylon Thermally Conductive and Electrically Non-Conducting Filament. <https://tcpoly.com/purchase-ice9-materials/>. TCPoly.