StrutModeling: A Low-Fidelity Construction Kit to Iteratively Model, Test, and Adapt 3D Objects

Danny Leen*
LUCA School of Arts, KULeuven
Genk, Belgium
danny.leen@luca-arts.be

Raf Ramakers*, Kris Luyten
UHasselt - tUL - imec
Expertise Centre for Digital Media
Diepenbeek, Belgium
firstname.lastname@uhasselt.be

ABSTRACT
We present StrutModeling, a computationally enhanced construction kit that enables users without a 3D modeling background to prototype 3D models by assembling struts and hub primitives in physical space. Physical 3D models are immediately captured in software and result in readily available models for 3D printing. Given the concrete physical format of StrutModels, modeled objects can be tested and fine tuned in the presence of existing objects and specific needs of users. StrutModeling avoids puzzling with pieces by contributing an adjustable strut and universal hub design. Struts can be adjusted in length and snap to magnetic hubs in any configuration. As such, arbitrarily complex models can be modeled, tested, and adjusted during the design phase. In addition, the embedded sensing capabilities allow struts to be used as measuring devices for lengths and angles, and tune physical mesh models according to existing physical objects.

ACM Classification Keywords
H.5.2 [Information interfaces and presentation]: User Interfaces

Author Keywords
Tangible Modeling; Construction Kit; 3D Modeling; Personal/Digital Fabrication

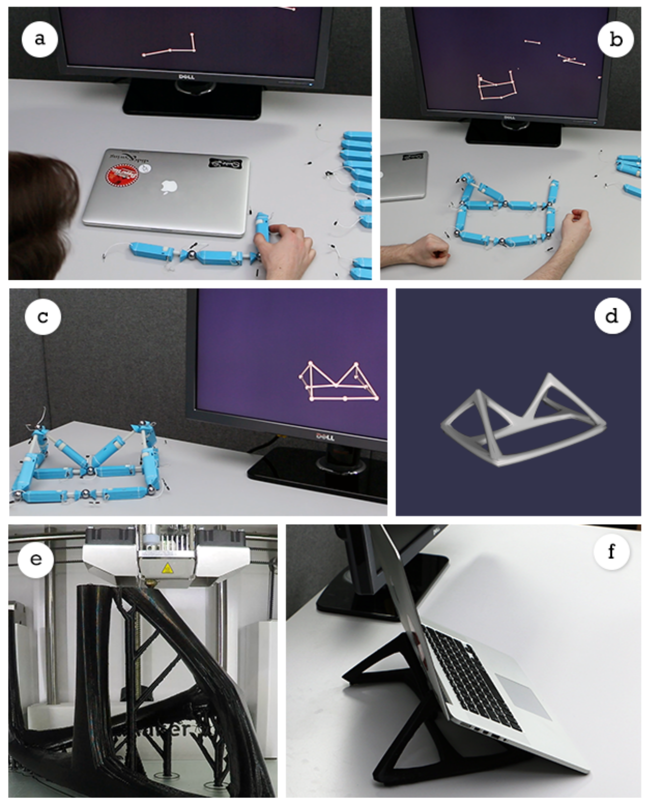

Figure 1. An ergonomic laptop stand modeled, tested, and adjusted with StrutModeling.

INTRODUCTION
Over the past decade, 3D printers became available and affordable for the mass public and enabled people without a 3D modeling or engineering background to produce 3D objects. While basic models can be downloaded from online libraries (e.g. Thingiverse[1]), the true premise of DIY 3D printing is the production highly customized and personal objects in low volumes at low costs. Designing a custom 3D object, however, requires modeling expertise. End-user CAD environments, such as Google Sketchup[2] and Autodesk 123D Design[3] lower the threshold for non-experts to design 3D models after following introductory online tutorials. However, designing models that can be adjusted over time requires additional expertise to organize and structure operations. For example, employing dynamic constraints to preserve an object's shape when its dimensions change. Furthermore, one can only test the functionality or ergonomy of a virtual model once it is produced. This disconnect between the virtual design environment and the physical target model oftentimes results in slow design iterations as objects have to be 3D printed before they can be tested.

*The first two authors contributed equally to this work
[1]https://www.thingiverse.com
[2]http://www.sketchup.com
[3]http://www.123dapp.com/design

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.

UIST 2017, October 22–25, 2017, Quebec City, QC, Canada

© 2017 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ISBN 978-1-4503-4981-9/17/10…$15.00
DOI: https://doi.org/10.1145/3126594.3126643

To bridge the gap between the software modeling environment and the produced artifact, researchers explored tangible modeling techniques using clay [28], blocks [16], LEGO bricks [2], or polyhedral shapes [12]. These physical props are digitized using optical scanning techniques [28] or by embedded sensors in the building primitives to reflect changes in the digital environment in real-time [12, 2]. While clay modeling allow for fine grained details and arbitrary shapes, they require extensive crafting skills. Clay does not offer any constraint, such unconstrained interactions require good dexterity to precisely shape materials. In contrast, construction kits offer primitives that are convenient to assemble. Composing precise structures, however, involves puzzling with many parts. For example, changing the angle between two LEGO brick walls requires repositioning or replacing many elements.

In this paper, we present StrutModeling, an electronically augmented construction kit with a strut and hub design. By connecting struts and hubs, end-users physically prototype 3D objects of which the geometry is immediately translated to a virtual model. Unlike existing truss systems, where struts come in different lengths and hubs have predefined anchor points, our system uses a one size fits all approach and consists of adjustable struts and a universal hub design. Struts are adjustable in size and hubs support a variable number of anchor points at any angle. StrutModeling therefore supports arbitrary complex geometric models and does not require interchanging struts and hubs while fine-tuning the design. Instead, users simply change the angle and/or length of a strut to transition to a vast set of shapes that are topological equivalent (homeomorphic) or add extra struts and hubs to further scale the object. StrutModeling encourages testing, experimentation, and iterative design changes during the modeling process. These changes are updated in real-time in the rendering environment. Given the concrete physical format of StrutModels, design decisions are made in the presence of existing physical objects and capture specific needs and feedback of users. As such, parameters can be fine-tuned and tested before 3D printing the final model.

WALKTHROUGH: AN ERGONOMIC LAPTOP STAND

Designing a custom ergonomic laptop stand using a traditional CAD software environment requires several considerations. Besides measuring the dimensions of the laptop, one needs to determine the optimal angle of the stand to ensure the screen is positioned at eye level and well-aligned with the height of an external monitor (Figure 1c). This design process could even involve adjusting the position of the external monitor or the user's chair. As a new high-fidelity design needs to be produced before the design can be tested in the context of other objects, realizing an optimal laptop stand could be a tedious process requiring 3D printing improved versions over multiple nights.

Figure 1 illustrates the integrated design and testing process, supported by StrutModeling, to make an ergonomic laptop stand: (a-b) The user starts by connecting struts using magnetic hubs. He adjusts the length of each strut by turning the threaded rod mechanism clockwise (extending) or anticlockwise (shortening). The StrutModeling rendering environment responds in real-time by updating the virtual model in correspondence with the topology of the physical struts. (c) The user positions the prototyped stand next to the external monitor, positions the laptop on top, and verifies the shape and size of the stand. As such, he can verify and adjust its ergonomy in the environment in which the final laptop stand will be used. The laptop is not at eye level and the user adjusts the length of the struts to increase the height and angle of the stand. Using incremental adjustments and tests, the user finalizes his design. (e,f) When the design is finished, the StrutModeling rendering environment converts the StrutModel to a 3D printable design by automatically applying post-processing steps. (g) The final design is ergonomically sound from the first 3D printing iteration and can be deployed.

CONTRIBUTIONS

The main contribution of this paper is a computationally enhanced construction kit that enable users without a 3D modeling background to prototype arbitrarily complex virtual models by assembling strut and hub primitives in physical space. As StrutModeling allows to physically render detailed wireframe meshes in physical space, modeled objects can be tested and fine-tuned in the presence of existing objects and specific needs of users. StrutModeling avoids puzzling with many pieces by offering an adjustable strut and a universal hub design that connects to struts in any configuration. StrutModels are therefore convenient to adjust and encourage fast design iterations before the fabrication process. In addition, the embedded sensing capabilities allow struts to be used as measuring devices for lengths and angles. To demonstrate the utility and versatility of StrutModeling, we present several example designs and use-cases.

RELATED WORK

This work draws from, and builds upon prior work in end-user 3D modeling software, construction and modeling toolkits, and tools and techniques to accelerate the fabrication and design process.

End-User CAD Software

Researchers have looked into CAD environments to lower the barrier of 3D content authoring. With Sketch [35] novices design and alter 3D shapes using gestures and simple line drawings. Teddy [13] builds on top of this work and offers an interface that generates 3D shapes based from 2D silhouette drawings. The system inflates the region surrounding the silhouette which makes this approach particularly suitable for designing characters. In a similar vein, SketchChair [27] automatically generates sturdy chairs from 2D silhouettes. ModelCraft [30] shares inspiration with these approaches but allows for making 3D models by sketching and gesturing directly on a paper version of the model using the Anoto pen.

Instead of manually designing models in CAD environments, recently there is also an increasing interest in automatically generating 3D shapes to adapt and patch 3D scanned models [23, 4] or to reserve space for inserting electronic sensors and devices [19]. While heuristics and physics simulations allow for basic optimizations and testing of virtual models [27, 4], extensive tests in the presence of existing objects, the environment, and users are only possible after producing the object.

Construction and Modeling Toolkits

Leveraging humans' tacit knowledge for manipulating and shaping physical objects, researchers used 3D scanning to digitize clay models [2, 28]. To capture geometries during the tangible modeling process, advanced computer vision algorithms have been developed to track incremental changes to LEGO Duplo constructions [11, 20]. While these approaches are robust to occlusions from hands, vision-based approaches are limited to capturing the outside shape visible to sensors.

As an alternative approach, electronic sensors have been integrated in building blocks to recognize their topology. The ActiveCube [16] recognizes structures created with cubical blocks that are placed on top or next to each other. Blocks communicate their topology using an embedded real-time communication network. Ikegawa et al. [14] present a block system with a simpler electronic circuit, consisting of only a capacitor integrated in every block. By measuring the total capacitance, the base plate infers the total number of stacked blocks. Unlike the ActiveCube system, however, blocks cannot be stacked sideways. To automatically capture the topology of LEGO structures, Anderson et al. [2] augment LEGO-like bricks with electronic sensors. As every pair of knobs and tubes in a brick have a power signal and communication channel, sturdy designs can be created by placing bricks in a staggered joint layout. Unlike strut and hub models, changes to brick constructions cascade rapidly as many stacked bricks need to be repositioned when altering the design.

In contrast to blocks, several projects presented electronically augmented planar surfaces. Using a base plate with electronic slots for recognizing inserted vertical panels, the Segal Model [26] presented a construction kit to experiment with layout configurations of traditional single floor Segal-style houses. To support a wider variety of shapes, the Triangle [9] system offers triangular shapes to form 2D and 3D shapes. Triangles attach to each other and communicate their IDs and thus topology using magnetic connectors. EasiGami [12] takes a similar approach but also provides pentagons, hexagons, and square shapes to allow for constructing complex shapes by connecting only a few pieces. StrutModeling takes inspiration from these approaches but offers a universal strut design that is adjustable in length and connect to other struts in any angle, using hubs. As such, StrutModeling avoids puzzling with pieces, common in planar, origami-like kits.

Closest to our work, a number of construction kits have a hub and strut layout. FlexM [6] proposes a kit consisting of hubs with three adjustable anchor points. To capture the topology, the work proposes and explores implementations in which hubs communicate by transmitting light signals through struts and the angle between struts is measured using potentiometers. Building on top of these concepts, Posey [34] is a tangible modeling kit that senses the topology as well as orientation of struts using infrared LEDs and sensors in every hub. Jacobson et al. [15] present a construction kit that uses a hall effect sensor to track the rotation of joints and communicates states through an embedded bus system. As hubs have a limited number of anchor points and require a minimum of 90 degrees between their angles, the kit is particularly suitable for modeling and animating skeletons. Glauser et al. [8] further optimized the design to eliminate gimbal lock effects. To allow for designing more organic forms, the Senspectra tool [18] uses malleable struts that bend in any direction. Bend angles are estimated by analyzing the reflectance of IR light transmitted through struts. Gluss [24] and Topobo [22] present similar construction kits but focus on rendering and animating forms in the physical world using actuators instead of digital modeling. Our work is different in that hubs do not have a fixed number of anchor points and allow for small angles between adjacent struts. As such, StrutModeling is optimized for modeling detailed outer-surface of objects that are ready for fabrication. In contrast, previous approaches target skeleton modeling and focus on learning and play [34] or animation [8].

Also related to our work are tools that facilitate modeling using physical props or novel measurement tools during modeling operations. ShapeTape [10] uses a strip, embedding optical sensors, to measure the curvature of objects. Alternatively, an array of strain gauges can be used, as shown in [5]. SPATA [31] presents novel electronic versions of calipers and protractors to automatically input distance and angular measurements in digital models. To use a complete object during modeling operations (e.g. add or subtract operation), researchers used 3D scanners to immediately capture objects [33] or their imprints in clay [7]. Similar to these approaches, our struts and hubs can also be used as measurement tools: one or multiple struts can be aligned with an existing object to digitize its length. Similarly, adjusting the angle between two adjacent struts digitizes angular measurements.

Accelerating the Fabrication and Design Process

StrutModeling accelerates the fabrication process as models are tested and fine-tuned during the design phase and only the final design is fabricated. Previous experiments, for example, accelerate the 3D printing process by printing only the model's wireframe [21]. Reform [32] presents a system for bi-directional fabrication. This system facilitates and accelerates prototyping iterations by supporting iterative adaptations of clay models by the user as well as the system. Similar to StrutModeling, Protopiper [1] offers a low-fidelity prototyping tool for physically rendering structures in the presence of other objects. The TrussFab [17] tool, in contrast, starts from a virtual model which is decomposed into truss geometries that can be assembled with bottles. As these low-fidelity assemblies can be tested immediately, they speed up design iterations.

SYSTEM ARCHITECTURE AND IMPLEMENTATION

StrutModeling offers an adjustable strut design and universal magnetic hubs that connect to struts in any configuration. We created extensible struts which embed computational elements for both sensing and communicating their orientation, size, and topology information. Struts are self-contained and constitute a modular toolkit. As precise absolute positioning is not feasible without external tracking systems, our approach realizes accurate digital reconstructions by combining the orientation data of all struts with topology information, describing the

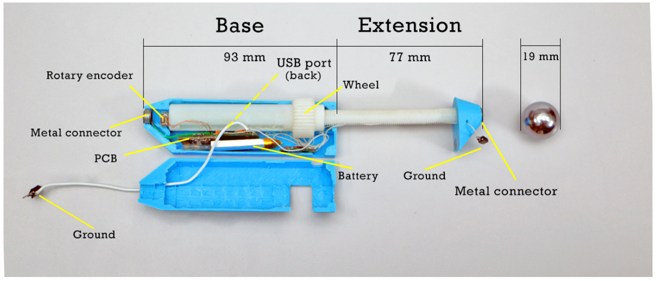

Figure 2. The inside mechanism of a strut.

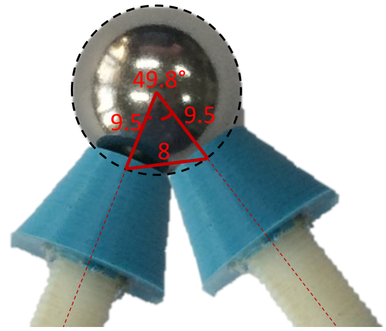

Figure 3. Minimum angle between struts.

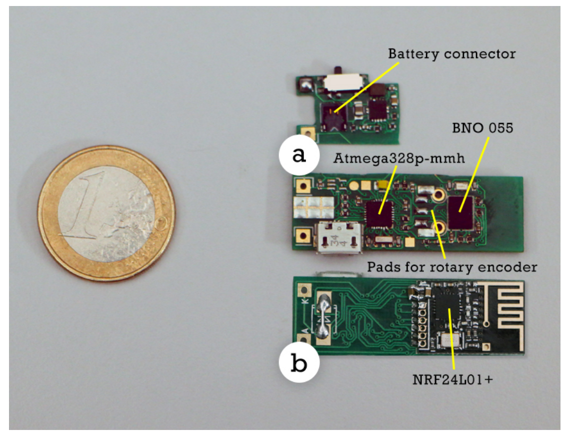

Figure 4. Custom PCB embedded in struts: (a) front, (b) back.

struts which are directly connected through a hub. Every strut is responsible for tracking its orientation and identifying adjacent struts. Unlike hubs with fixed anchor points, magnetic hubs snap to struts in any configuration. On the flipside however, connected struts only have a single contact point which makes it impossible to both power the system as well as transmit data. As such, every strut is powered with a battery and communicates wirelessly with a desktop computer that runs our virtual reconstruction environment. This environments continuously captures and digitizes the full topology of all connected struts. In contrast to the struts, the magnetic hubs are entirely passive. Besides being used as structural support, they also offer a conductive route to identify adjacent struts. Although this approach scales to any number of struts and hubs, the nrf wireless communication protocol, used in our implementation, supports up to 780 nodes with multicast disabled.

Mechanical Design

Figure 2 shows the inside of a strut which consists of a base (9.3cm / 3.66inches) and an extension part (7.7cm / 3inches). The extension part has a threaded rod design and connects to a nut in the base. The wheel shown in Figure 2 turns the nut and actuates the threaded rod. A notch at the end of the base fits in the sleeve of the threaded rod and ensures that the rotary movement of the wheel is translated in a linear movement of the extension part. This mechanical transfer system offers precise control over the length of a strut as the threaded rod has a pitch of 1mm.

The hub shown in Figure 2 consists of a magnetic bullet. The hub has a diameter of 19mm, weights 27grams and has a holding strength of 5.6kg/197ounces and approximately 0.84kg/29.6ounces shear strength to ensure stability of StrutModels even without strutting hubs vertically. Both sides of a strut end in a nozzle of 8mm in diameter. Given the current design of the struts, the minimum angle between adjacent struts is 49.8° (Figure 3). This significantly lowers the constraints for modeling freeform objects compared to minimum angles of approximately 90° by previous toolkits [15, 34]. Future versions of struts can be reduced in thickness to further lower the angles between struts. Smaller angles could also be achieved using larger hubs, yet this would result in higher magnetic forces that makes disconnecting struts from hubs uncomfortable.

Electronic Design

Figure 4 shows our custom PCB that fits in a strut (Figure 2) and integrates all components required to track the orientation and length of a strut, identify adjacent struts, communicate wirelessly, and power and charge a strut. Struts run on a 8-bit AVR Atmega328p-mmh microcontroller (Figure 4) that are deployed with the Arduino platform. To track the absolute orientation of a strut, previous experiments [15] show that accelerometer and magnetometer data is unstable for absolute orientation as the magnetic field in office environments are unstable and lead to errors above 40°. Especially in the presence of magnetic hubs, the magnetic field diverges even further which makes geomagnetic sensors highly unstable. Therefore, our approach fuses data that comes from an accelerometer with the gravity vector from a gyroscope to calculate the absolute orientation of every strut. All this data is delivered by ARM an Cortex M0+ microcontroller embedded in the Bosch BNO055 integrated circuit, which packs 3 different sensors: a triaxial 16-bit gyroscope, a triaxial 14-bit accelerometer, and a geomagnetic sensor, of which the latter one is disabled. As orientation information of accelerometers is relative to their starting configuration, the user must align all struts before the system sets the reference orientation.

Extensibility of the struts is implemented using a mechanical wheel connected with a screw-thread rod. Rotations of the mechanical wheel are tracked with a rotary encoder, which is

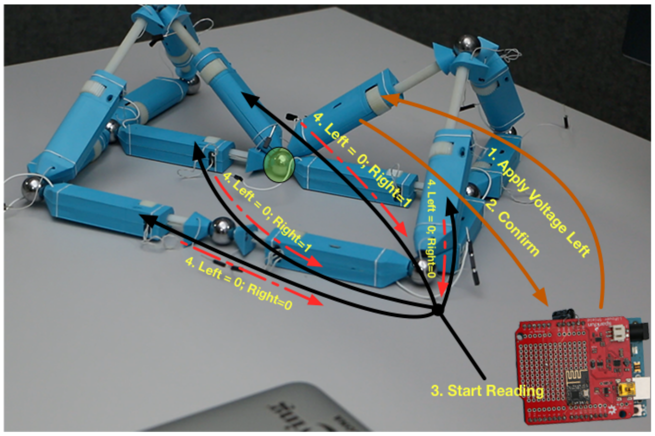

Figure 5. Protocol to identify adjacent struts and reconstruct the topology.

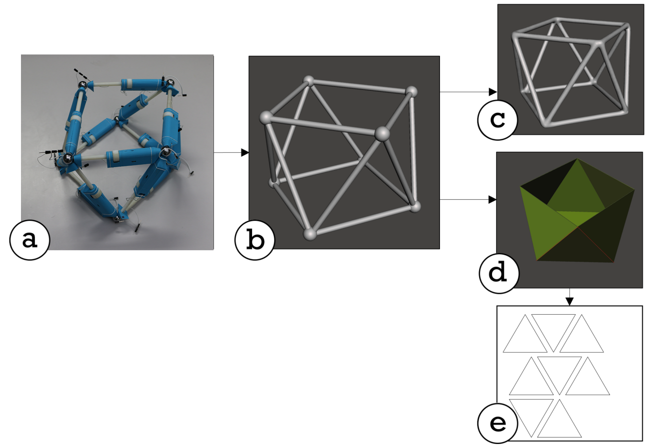

Figure 6. (a) StrutModel of a cookie jar, (b) Digitized strutmodel. Export options: (c) Reinforced wireframe 3D model (3D printing), (d) A single shell convex hull (3D printing), (e) 2D vector graphics (lasercutting).

mounted at the end of the strut (Figure 2). The encoder produces 12 pulses per second for millimeter precision tracking. Each metal end of a strut (Figure 2) is connected to a GPIO pin to identify adjacent struts, as explained in the next section. All data is transmitted wirelessly using the Nordic nRF24L01+ 2.4Ghz radio frequency transceiver (Figure 4) to a master Arduino Uno module that uses the same RF transceiver. This master module automatically discovers new struts in the mesh network and communicates with the rendering environment using serial communication. At its peak, our PCB consumes 50mAh of power and thus runs for over 2.5hours on a 130mAh LiPo battery. The battery is small enough to fit in our current design. The electronic components for a single strut cost around \$17 or €16 for prototyping quantities.

Topology Tracking

StrutModeling calculates the relative position of all struts and hubs by combining the absolute rotations of the struts, from fusing the gyroscope and accelerometer data, with topology information. Topology data describes which struts are directly connected to each other and thus share the same hub. Adjacent struts identify when one of them applies a voltage on the shared hub. This voltage is however floating as the circuits are completely separated. For other struts to read and recognize this voltage, a common ground is required. To realize this, both ends of a strut integrate a stackable header pin that is connected with a short wire to ground. Although a single connection to ground would be sufficient to realize a common ground, we chose to embed a connector in both ends to allow for using struts omnidirectional.

Figure 5 shows how the master module wirelessly communicates with all struts and reconstructs the topology: The master module starts by instructing one strut to apply a voltage on one end of the strut. When the strut confirms, the master module instructs all other struts to measure the voltage on connected hubs and transmit the readings to the master module. The master module processes all data and and registers struts to be connected that measured a high voltage. The master module then starts a new cycle in which another strut is instructed to apply a voltage on a hub. When traversing all struts, the ends of struts that already measured a high voltage are excluded. Afterwards, a new iteration starts. As our mesh network has a bandwidth of 2Mbit, reconstructing the topology of 20 struts takes less than one second. When a connection is dropped, a timeout is triggered and the master module recovers and continues. In an alternative implementation, all struts in parallel can transmit a unique square wave over the hubs. However, adjacent struts should first perform a handshake to ensure only one strut is broadcasting over the hub.

We also implemented a topology tracking technique that eliminates the common ground and is instead based on capacitive sensing for identifying the mesh topology. This requires a user to touch every hub, which results in a change in capacitance. This change is detected by the struts that are connected, identifying them as connected to the same hub. While this reconstruction technique eliminates the need for a common ground, it is not completely fail safe as different hubs might be touched simultaneously. The technique is therefore most suited for models that do not require real-time updates in the rendering environment and instead render the final model once the user goes through a procedure during which he is instructed to touch every hub one by one.

Rendering Environment and Post-Processing

The topology information, absolute orientation, and length of struts are collected on the master module and streamed over serial communication to the rendering environment running on a desktop computer. The software environment fuses all geometric information and calculates the relative position of every strut and hub. For every closed polygon, the absolute orientation of one strut is redundant as the start and end position can be calculated from the geometry information of adjacent struts. We use this information to compensate for potential drift caused by the gyroscope and accelerometer. The final geometry is rendered in real-time using the Babylon.js[4] 3D engine and consists of cylindrical and spherical shapes (Figure 6b).

[4]https://www.babylonjs.com

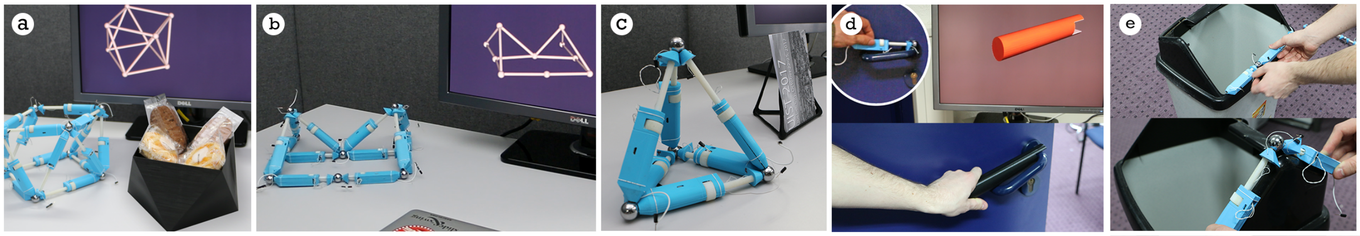

Figure 7. Example designs and use cases: (a) A cookie jar, (b) an ergonomic laptop stand, (c) a photo holder, (d) An extension for a door handle, (e) Repairing a broken trash can.

Our rendering environment runs Meshmixer and OpenSCAD in the background to prepare the virtual rendering for fabrication. Three post-processing options are supported (Figure 6): (1) 3D printing a reinforced wireframe model as shown in Figure 6c. In this configuration, the rendering environment starts by replacing the struts with extended cylinder shaped meshes that replace the hubs. To reinforce the model, these cylinder meshes are converted to a single organic shape with a uniform triangle distribution using the smoothing operation available in the Meshmixer API [29] (Figure 6c). The resulting 3D model (STL-file) has improved shape properties and can be fabricated using a 3D printer. (2) 3D printing the convex hull to realize an enclosed model (Figure 6d). In this export option, the relative positions of the hubs are automatically imported in OpenSCAD to produce a mesh of the convex hull with OpenSCAD[5]. The resulting mesh (STL-file) can be 3D printed in a single shell (Figure 6d) or after thickening the object by extruding all faces in the direction of the normal vector using the Meshmixer API. Converting a StrutModel in a convex hull might not always produce the desired shape. Therefore, users can select additional or discard existing planes e.g. discarding the top lid of a box. (3) Alternatively, the convex hull can be produced using a lasercutter to speed up the fabrication process (Figure 6e). In this configuration, the vertices of the convex hull are projected onto a 2D plane and exported to vector graphics (SVG-file). Instead of gluing the resulting panels together, future versions could automatically generate joints as shown in Platener [3].

USAGE AND EXAMPLE DESIGNS

To validate our approach, we created several example designs that demonstrate the utility and versatility of StrutModeling in different settings (Figure 7). All of the models took less than 15 minutes to be created once the idea was raised, without any prior upfront design. Conversion into a format that is suitable for 3D printing or lasercutting is virtually instantaneous.

StrutModeling supports modeling of mundane objects, such as the cookie jar, the ergonomic laptop stand, and the photo holder shown in Figure 7a-c. While our struts and hubs enable non-experts to make highly personalized 3d objects, key to our approach is that StrutModels can be modeled and tested in during the design phase in the presence of existing objects as demonstrated in the walkthrough. Example design (a) is exported as a convex hull and 3D printed as a single shell object. Designs (b-c) feature the production of sturdy organic wireframe objects from StrutModels.

[5] http://www.openscad.org

As struts and hubs can also be used as measuring tools for lengths and angles, StrutModeling facilitates adapting parameters of existing 3D models. Users can use struts to adjust pre-defined parameters of existing parametric models designed by experts. For example, adapting the length of an extension for a door handle, makes it possible to fine-tune the lever and thus the force required to operate the door. By attaching a strut to the door handle and adjusting its length, caregivers are empowered to test various sizes for door handles with impaired users (Figure 7d). Changes to the handle are immediately reflected in the personalized virtual 3D model which is already tested and ready for production. Figure 7e shows how StrutModeling can be used as measuring tools when repairing a broken lid of a trash can. By extending and connecting multiple struts, the user measures the required width of the lid. The angle between the two panels of the lid can be measured precisely by aligning two adjacent struts with the curve at the top of the trash can.

Customizing existing 3D models, not created with StrutModeling, requires a parametric design specifying the mapping of parameters to changes in the virtual model, such as models designed in Thingiverse Customize[6] with OpenSCAD. With easy-to-use modeling kits, such as StrutModeling, we expect parametric 3D designs to become increasingly popular as these tools bridge the gap between professional designers, making the parametric models, and non-experts who adjust and personalize parameters.

LIMITATIONS AND FUTURE WORK

StrutModeling has three limitations we feel are important to mention:

First, features smaller than a single strut cannot be modeled. However, the modular design of struts and their length adjustment mechanism ensures that models scale to arbitrarily complex geometries larger than one strut. Although it is possible to model very large structures, such as couches or closets, a large number of struts would be required as struts only extend approximately one length. To reduce the number of struts in larger structures, future versions of struts could embed a telescoping mechanism to extend multiple lengths. Such a mechanism would also enable faster and coarser interactions as compared to the current rotary wheel mechanism. Larger structures, such as a chair, are sometimes subject to extensive loads. Future versions of the rendering environment could hint users on the structural rigidity of the designed model

[6] https://www.thingiverse.com/customizer

by analyzing the topology of the underlying graph structure, as described in [25]. In the mean-time, structures subject to gravity and average loads (e.g. laptop stand), are supported by increasing the thickness of beams or the wall thickness of convex hulls.

Second, the topology tracking technique requires interconnecting the grounds of all struts. As the conductive magnetic hubs are used for transmitting signals to identify adjacent struts, a common ground needs to be established using an additional connector to every strut. Multiple conductive channels could be integrated in magnetic hubs at the expense of fixed anchor points for struts and limited degrees of freedom for moving struts as shown in [15]. To avoid wiring struts, in the future, we would like to explore mechanical approaches for identifying adjacent struts, including, transmitting vibrations and identifying color-coded hubs with struts.

Last, data from the accelerometer and gyroscope, used for tracking the absolute orientation of struts are subject to drift over time. However, by fusing both data sources and automatic recalibration strategies integrated in the IMU sensor, these drifts are minimized. We further mitigate inaccuracies in sensor readings using topology information. For every closed polygon in the model, our system optimizes position and orientation information to ensure the geometry remains connected.

CONCLUSION

In this paper we presented StrutModeling, a modular toolkit that enables non-expert users to prototype physical objects using a strut and hub construction kit. We created struts that embed computational components to sense, identify, and share their relative orientation, neighbouring struts, and length. Using magnetic bullets as connecting hubs ensures maximal freedom in what can be modeled: hubs can be connected with an arbitrary number of struts, and pairwise struts can have a wide variation of angles between them. We showed that our toolkit allows for creating arbitrarily complex prototypes. Given the concrete physical format of StrutModels, users can test and evaluate their design directly in the context of existing objects. StrutModeling automatically captures the 3D model and converts it into formats that can be 3D printed or lasercut.

ACKNOWLEDGMENTS

We thank Steven Nagels and Peter Quax for providing advice on techniques for tracking topologies. We also thank Fablab Genk and Fablab Leuven for 3D printing models and Jan Van den Bergh, Florian Heller, and Kashyap Todi for proofreading this paper. This research was supported by the Research Foundation - Flanders (FWO), project G0E7317N End-User Development of Intelligible Internet-of-Things Objects and Applications.

REFERENCES

1. Harshit Agrawal, Udayan Umapathi, Robert Kovacs, Johannes Frohnhofen, Hsiang-Ting Chen, Stefanie Mueller, and Patrick Baudisch. Protopiper: Physically Sketching Room-Sized Objects at Actual Scale. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology (UIST '15)*. ACM, New York, NY, USA, 427–436. DOI: http://dx.doi.org/10.1145/2807442.2807505
2. David Anderson, James L. Frankel, Joe Marks, Aseem Agarwala, Paul Beardsley, Jessica Hodgins, Darren Leigh, Kathy Ryall, Eddie Sullivan, and Jonathan S. Yedidia. Tangible Interaction + Graphical Interpretation: A New Approach to 3D Modeling. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '00)*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 393–402. DOI: http://dx.doi.org/10.1145/344779.344960
3. Dustin Beyer, Serafima Gurevich, Stefanie Mueller, Hsiang-Ting Chen, and Patrick Baudisch. Platener: Low-Fidelity Fabrication of 3D Objects by Substituting 3D Print with Laser-Cut Plates. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 1799–1806. DOI: http://dx.doi.org/10.1145/2702123.2702225
4. Xiang 'Anthony' Chen, Jeeeun Kim, Jennifer Mankoff, Tovi Grossman, Stelian Coros, and Scott E. Hudson. Reprise: A Design Tool for Specifying, Generating, and Customizing 3D Printable Adaptations on Everyday Objects. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST '16)*. ACM, New York, NY, USA, 29–39. DOI: http://dx.doi.org/10.1145/2984511.2984512
5. Chin-yu Chien, Rong-Hao Liang, Long-Fei Lin, Liwei Chan, and Bing-Yu Chen. FlexiBend: Enabling Interactivity of Multi-Part, Deformable Fabrications Using Single Shape-Sensing Strip. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology (UIST '15)*. ACM, New York, NY, USA, 659–663. DOI: http://dx.doi.org/10.1145/2807442.2807456
6. Markus Eng, Ken Camarata, Ellen Yi-Luen Do, and Mark D Gross. 2006. Flexm: Designing a physical construction kit for 3d modeling. *International Journal of Architectural Computing* 4, 2 (2006), 27–47.
7. Sean Follmer and Hiroshi Ishii. KidCAD: Digitally Remixing Toys Through Tangible Tools. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, New York, NY, USA, 2401–2410. DOI: http://dx.doi.org/10.1145/2207676.2208403
8. Oliver Glauser, Benedek Vartok, Wan-Chun Ma, Daniele Panozzo, Alec Jacobson, Cédric Pradalier, Otmar Hilliges, and Olga Sorkine-Hornung. 2017. Rig Animation with a Tangible and Modular Input Device. *interactions* 24, 2 (Feb. 2017), 16–17. DOI: http://dx.doi.org/10.1145/3041959
9. Matthew G. Gorbet, Maggie Orth, and Hiroshi Ishii. Triangles: Tangible Interface for Manipulation and Exploration of Digital Information Topography. In

Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '98). ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 49–56. DOI: http://dx.doi.org/10.1145/274644.274652

10. Tovi Grossman, Ravin Balakrishnan, and Karan Singh. An Interface for Creating and Manipulating Curves Using a High Degree-of-freedom Curve Input Device. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '03)*. ACM, New York, NY, USA, 185–192. DOI: http://dx.doi.org/10.1145/642611.642645

11. Ankit Gupta, Dieter Fox, Brian Curless, and Michael Cohen. DuploTrack: A Real-time System for Authoring and Guiding Duplo Block Assembly. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology (UIST '12)*. ACM, New York, NY, USA, 389–402. DOI: http://dx.doi.org/10.1145/2380116.2380167

12. Yingdan Huang and Michael Eisenberg. Easigami: Virtual Creation by Physical Folding. In *Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction (TEI '12)*. ACM, New York, NY, USA, 41–48. DOI: http://dx.doi.org/10.1145/2148131.2148143

13. Takeo Igarashi, Satoshi Matsuoka, and Hidehiko Tanaka. Teddy: A Sketching Interface for 3D Freeform Design. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '99)*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 409–416. DOI: http://dx.doi.org/10.1145/311535.311602

14. Koshi Ikegawa, Masaya Tsuruta, Tetsuya Abe, Arika Yoshida, Buntarou Shizuki, and Shin Takahashi. Lightweight Capacitance-based Block System for 3D Space Interaction. In *Proceedings of the 2016 ACM on Interactive Surfaces and Spaces (ISS '16)*. ACM, New York, NY, USA, 307–312. DOI: http://dx.doi.org/10.1145/2992154.2996772

15. Alec Jacobson, Daniele Panozzo, Oliver Glauser, Cédric Pradalier, Otmar Hilliges, and Olga Sorkine-Hornung. Tangible and Modular Input Device for Character Articulation. In *ACM SIGGRAPH 2014 Emerging Technologies (SIGGRAPH '14)*. ACM, New York, NY, USA, Article 24, 1 pages. DOI: http://dx.doi.org/10.1145/2614066.2614072

16. Yoshifumi Kitamura, Yuichi Itoh, and Fumio Kishino. Real-time 3D Interaction with ActiveCube. In *CHI '01 Extended Abstracts on Human Factors in Computing Systems (CHI EA '01)*. ACM, New York, NY, USA, 355–356. DOI:http://dx.doi.org/10.1145/634067.634277

17. Robert Kovacs, Anna Seufert, Ludwig Wall, Hsiang-Ting Chen, Florian Meinel, Willi Müller, Sijing You, Maximilian Brehm, Jonathan Striebel, Yannis Kommana, Alexander Popiak, Thomas Bläsius, and Patrick Baudisch. TrussFab: Fabricating Sturdy Large-Scale Structures on Desktop 3D Printers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 2606–2616. DOI: http://dx.doi.org/10.1145/3025453.3026016

18. Vincent LeClerc, Amanda Parkes, and Hiroshi Ishii. Senspectra: A Computationally Augmented Physical Modeling Toolkit for Sensing and Visualization of Structural Strain. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '07)*. ACM, New York, NY, USA, 801–804. DOI: http://dx.doi.org/10.1145/1240624.1240744

19. David Ledo, Fraser Anderson, Ryan Schmidt, Lora Oehlberg, Saul Greenberg, and Tovi Grossman. Pineal: Bringing Passive Objects to Life with Embedded Mobile Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 2583–2593. DOI: http://dx.doi.org/10.1145/3025453.3025652

20. Andrew Miller, Brandyn White, Emiko Charbonneau, Zach Kanzler, and Joseph J LaViola Jr. 2012. Interactive 3D model acquisition and tracking of building block structures. *IEEE transactions on visualization and computer graphics* 18, 4 (2012), 651–659. DOI: http://dx.doi.org/10.1109/TVCG.2012.48

21. Stefanie Mueller, Sangha Im, Serafima Gurevich, Alexander Teibrich, Lisa Pfisterer, François Guimbretière, and Patrick Baudisch. WirePrint: 3D Printed Previews for Fast Prototyping. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. ACM, New York, NY, USA, 273–280. DOI: http://dx.doi.org/10.1145/2642918.2647359

22. Hayes Solos Raffle, Amanda J. Parkes, and Hiroshi Ishii. Topobo: A Constructive Assembly System with Kinetic Memory. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '04)*. ACM, New York, NY, USA, 647–654. DOI: http://dx.doi.org/10.1145/985692.985774

23. Raf Ramakers, Fraser Anderson, Tovi Grossman, and George Fitzmaurice. RetroFab: A Design Tool for Retrofitting Physical Interfaces Using Actuators, Sensors and 3D Printing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 409–419. DOI: http://dx.doi.org/10.1145/2858036.2858485

24. Robert L. Read. 2016. The Gluss Project. Article. (16 September 2016). Retrieved March 27, 2016 from https://hackaday.io/project/13277-the-gluss-project.

25. Ben Roth. 1981. Rigid and flexible frameworks. *The American Mathematical Monthly* 88, 1 (1981), 6–21. DOI:http://dx.doi.org/10.2307/2320705

26. Sutphen S., Sharlin E., Watson B., and Frazer J. Tangible Computer Interfaces and the Revival of the Segal Model. In *In Western Computer Graphics Symposium (WCGS '00)*.

27. 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 (TEI '11)*. ACM, New York, NY, USA, 73–80. DOI:http://dx.doi.org/10.1145/1935701.1935717

28. Valkyrie Savage, Sean Follmer, Jingyi Li, and Björn Hartmann. Makers' Marks: Physical Markup for Designing and Fabricating Functional Objects. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15)*. ACM, New York, NY, USA, 103–108. DOI: http://dx.doi.org/10.1145/2807442.2807508

29. Ryan Schmidt and Karan Singh. Meshmixer: An Interface for Rapid Mesh Composition. In *ACM SIGGRAPH 2010 Talks (SIGGRAPH '10)*. ACM, New York, NY, USA, Article 6, 1 pages. DOI: http://dx.doi.org/10.1145/1837026.1837034

30. Hyunyoung Song, François Guimbretière, Chang Hu, and Hod Lipson. ModelCraft: Capturing Freehand Annotations and Edits on Physical 3D Models. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology (UIST '06)*. ACM, New York, NY, USA, 13–22. DOI: http://dx.doi.org/10.1145/1166253.1166258

31. Christian Weichel, Jason Alexander, Abhijit Karnik, and Hans Gellersen. SPATA: Spatio-Tangible Tools for Fabrication-Aware Design. In *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction (TEI '15)*. ACM, New York, NY, USA, 189–196. DOI: http://dx.doi.org/10.1145/2677199.2680576

32. Christian Weichel, John Hardy, Jason Alexander, and Hans Gellersen. ReForm: Integrating Physical and Digital Design Through Bidirectional Fabrication. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15)*. ACM, New York, NY, USA, 93–102. DOI: http://dx.doi.org/10.1145/2807442.2807451

33. Christian Weichel, Manfred Lau, David Kim, Nicolas Villar, and Hans W. Gellersen. MixFab: A Mixed-reality Environment for Personal Fabrication. In *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 3855–3864. DOI: http://dx.doi.org/10.1145/2556288.2557090

34. Michael Philetus Weller, Ellen Yi-Luen Do, and Mark D Gross. Posey: Instrumenting a Poseable Hub and Strut Construction Toy. In *Proceedings of the 2Nd International Conference on Tangible and Embedded Interaction (TEI '08)*. ACM, New York, NY, USA, 39–46. DOI:http://dx.doi.org/10.1145/1347390.1347402

35. Robert C. Zeleznik, Kenneth P. Herndon, and John F. Hughes. 1996. SKETCH: An Interface for Sketching 3D Scenes. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '96)*. ACM, New York, NY, USA, 163–170. DOI:http://dx.doi.org/10.1145/237170.237238