Enabling the Fabrication of Smart Devices

Carlos E. Tejada
Rochester Institute of Technology
Rochester, NY, USA
cet1318@rit.edu

Daniel Ashbrook
Rochester Institute of Technology
Rochester, NY, USA
daniel.ashbrook@rit.edu

Osamu Fujimoto
Rochester Institute of Technology
Rochester, NY, USA
oaf7862@rit.edu

Zhiyuan Li
Rochester Institute of Technology
Rochester, NY, USA
zl7904@rit.edu
author4@hchi.anotherco.com

Abstract

In recent years, digital fabrication equipment has experienced a significant drop in price. Devices that once were only available to scientists are now being marketed to enthusiasts as well. Sensors and complex computational abilities are also now available in mobile and wearable devices. Taken together, these trends suggest the possibility for end users to design and fabricate their own customized smart objects. In this paper, we describe our initial efforts towards easy fabrication of 3D-printable smart objects that use natural properties such as acoustic resonance for interaction.

Author Keywords

Digital Fabrication; 3D-printing; Acoustic Interaction

Introduction

While research on digital fabrication technology is perceived as a trend, this technology has been around for decades. It is thanks to the expiration of commercial patents in the 1980s and 1990s that this technology is now reaching the point where enthusiasts can acquire digital fabrication equipment at a fraction of the original cost.

Example of amateur work using these newly available technologies are wide-ranging. Shewbridge et al. discovered in their work [19] that, for many users, one of the most important tasks undertaken when presented with the opportunity

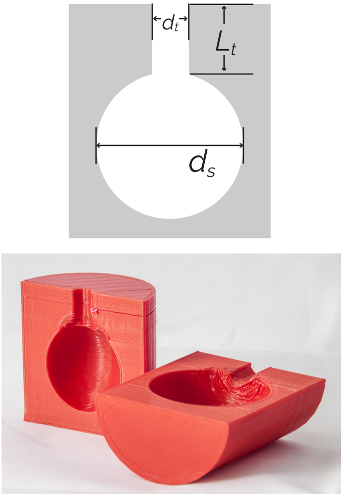

Figure 1: Top: an ideal spherical Helmholtz resonator, with tube diameter d_t, tube length L_t, and sphere diameter d_s. Bottom: cross-sections of two Blowhole test objects, showing the resonator structures.

to use a 3D printer is one of *augmented fabrication*: designing and fabricating new objects that work with already existing ones [2]. Additionally, this has been explored in the literature [1, 4, 5, 7–9], ranging from generating adaptations to already existing objects to increase performance in a specific task [6] to creating a custom housings for electronics [2].

Additionally, other fabrication-related research has explored adding interactivity to 3D-printed objects. By enabling interactions with 3D-printed objects, researchers can enable an abundance of educational and accessibility applications. Previous work has achieved this in two fundamental ways: by adding sensing capabilities to the 3D-printed object itself [3, 10, 15, 16, 18, 21–23], or to the environment where the interactions would take place [11, 12, 14, 17, 20, 24]; however, these two techniques have significant drawbacks. Adding sensing capabilities to the 3D-printed objects adds more complexity to an already complicated process. End-users would have to possess significant electronic and fabrication knowledge to be able to enable these interactions this way. In contrast, instrumenting the environment would constrain the users in a physical place where these interactions would be possible, limiting in turn the interaction possibilities.

Inspired by these trends and limitations, we propose a way for end-users to fabricate custom smart objects. In order to fabricate these objects, the users should not be required to possess any previous knowledge of electronics, advanced fabrication techniques, or programming—pressing print on a computer should be enough. We propose that these “smart objects” be powered by the increasing computational power and sensing capabilities of our everyday devices, like smartphones and smartwatches, which we can only asume would only get smaller and more reliable in the future.

Blowhole

An embodiment of this idea is Blowhole: a system to add interactivity to 3D printed objects by embedding blowable, resonant cavities in them. Blowhole is based on the property of acoustic resonance; a familiar example is the sound created when blowing across the mouth of a bottle. Blowhole embeds cavities into 3D models, with tubular openings to the surface. Varying the volumes of the cavities and lengths of the tubes produces varying frequencies in response to gentle blowing into the holes, with the object held 5–10 cm away from the mouth. Our system recognizes the characteristic sound of each hole, linking the blow sound to an action associated with the hole’s location on the model. Our design tool allows a user to select the placement of holes on arbitrary 3D models and associate actions with each hole; the software then optimizes blowhole size and placement, providing a printer-ready file.

The cavities used in Blowhole must satisfy several criteria: they must support sufficient variation in parameters to produce a range of frequencies when blown into; they must be sufficiently small to embed into models small enough to hold and manipulate; they should present a consistent hole appearance to the user; and they should be printable at any orientation and without support material on a consumer-grade printer.

Blowhole operates on the principle of acoustic resonance, where particular frequencies are amplified or attenuated due to the physical properties of a cavity. Blowhole uses spherical cavities inside a 3D-printed model with straight pipes opening onto the surface; the resonant frequency of a cavity depends on the area and length of the opening and the volume of the cavity, and is classically modeled using the Helmholtz resonance equation [13]:

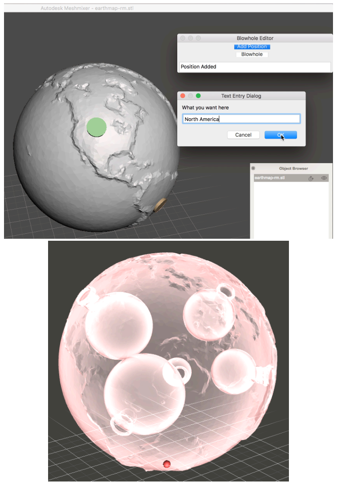

Figure 2: Blowhole design environment. Top: Detail view of our Blowhole design software, based on Meshmixer, showing the model after inserting blowholes. Bottom: Transparent visualization of the Blowhole cavities inside an object.

$$f = \frac{cd_t}{\pi}\sqrt{\frac{3}{8(L_t + .75d_t)d_s^3)}} \qquad (1)$$

with c the speed of sound, d_s the diameter of the spherical cavity and d_t and L_t the diameter and length, respectively, of the tube connecting the cavity to the surface of the object. Figure 1 illustrates these parameters of Blowhole cavities.

Blowhole Characterization

In order for Blowhole to be of the most practical use, we want to understand how many different cavities we can fit inside a given object. As can be seen from Equation (1), we can vary three parameters—d_t, L_t or d_s—to change the resonant frequency of a Blowhole cavity. As the tube is the only user-facing element of Blowhole, its appearance should be consistent, with the size of the opening large enough to easily blow into, but not so large as to interfere with the features of the printed model. After some initial experimentation, we set d_t to 5 mm, leaving L_t and d_s as the available parameters to manipulate. Multiple combinations of these can produce the same predicted frequency; for example, $L_t = 2.5$ mm and $d_s = 35.3$ mm produce a prediction of 1000 Hz, as do $L_t = 5$ mm and $d_s = 28$ mm.

To understand the practical limits on the frequencies we could detect and differentiate between, we produced a large number of test objects using consumer FDM printers (Qidi Technology X-One, LulzBot Taz 4, and LulzBot Taz Mini). Wanting to understand the practical limits on the frequencies we could detect and differentiate between, we produced a 48 objects with cavities and tubes of different sizes. Holding r_t at 2.5 mm, we manipulated d_s from 8–40 mm in steps of 4 mm and tested L_t at 2.5, 3.5, 5, 7.5, 8.5 and 10 mm.

We asked ten people to blow into each cylinder between one and four times, recording the data via a laptop computer's built-in microphone at a 44,100 Hz sampling rate. We extracted the fundamental frequencies of each blow using Welch's method [25]. To validate the printability of our cavities, we tested multiple cavity sizes, from 5 mm to 60 mm in diameter. While all objects printed correctly, the smallest cylinders resulted in frequencies highly variable over the course of a single blow, and the 60 mm cavity failed to produce any strong harmonic at all.

We also tested the consistency of sound at different angular positions of the tube opening, from 0° (straight down) to 180° (straight up) in 22.5° increments with d_s as 16 mm and L_t as 5 mm. Although different orientations revealed different (minor) printing artifacts such as slight stringing and tube opening shape inconsistency, the results were consistent, with a mean deviation of under 240 Hz from the Helmholtz-predicted value.

We tested Blowhole with multiple printers: a LulzBot Taz 4, a LulzBot Taz Mini, two Qidi X-One v2 printers, and a Form Labs Form 2 resin-based SLA printer. All printed successfully; inspecting the spectrograms, we found little variance amongst the FDM prints, and that the SLA prints produce dominant frequencies on average 100 Hz closer to the Helmholtz-predicted frequency than the FDM prints and with less variation over the signal.

System Implementation

As a system, Blowhole consists of three parts: the design software to modify existing 3D models to add blowholes; the physical printed-out models with resonant cavities and holes embedded; and the software that recognizes the

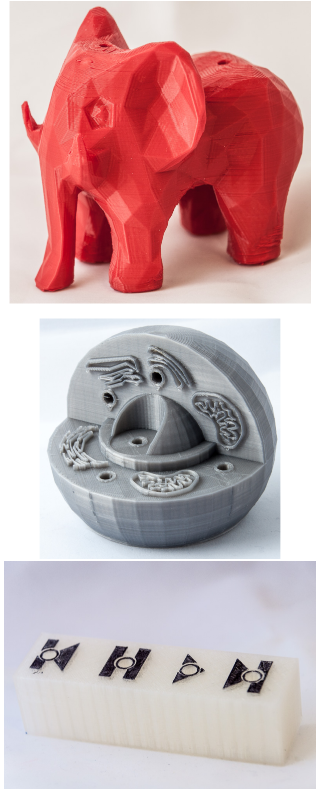

Figure 3: Examples of Blowhole-enabled objects
Top: An augmented elephant.
Center: An augmented cell model.
Bottom: A Blowhole-powered music remote control.

sound of the user blowing into a cavity and performs an action.

Design Software: Our design software is built on top of Autodesk Meshmixer (Figure 2, left) using its Python API. To add Blowhole tags to a model, a user simply imports an existing model and then clicks on the model to specify tag locations and desired actions. Currently supported actions include opening URLs, launching files such as images and movies, and reading text via text-to-speech. After the user indicates all of their desired blowhole positions, the software determines the best set of cavity sizes to embed in the model. We use a backtracking search algorithm that attempts to find an optimal set of L_t and d_s that will fit all requested blowholes without cavities colliding; we can optionally fix L_t to a single value. We represent the solution space as a tree, with each node mapping a set of available cavities to requested tag positions. At each step, we test the unused cavities in each candidate location until no collisions occur, pruning the tree whenever a solution cannot be found. The final solution is a set of location/cavity pairs, which we then use to construct the model for printing (Figure 2, right). Once the cavities are placed, the software writes out a configuration file linking the cavity parameters L_t and d_s to the specified action. The final model may be exported to a STL file for 3D printing on a commodity printer.

Our software places blowholes into existing models, therefore models that are 3D-printable will remain so with the addition of cavities and openings. Because the cavities are spherical, and most hobbyist-level 3D printers can print up to 45° of overhang, the models can be produced on most printers with no modification; importantly, no support material is necessary inside the cavities or tubes.

Blow Sound Recognition: The last component of our system recognizes the sounds produced by the user blowing into the blowholes, producing the resonant frequency characteristic to cavity/tube combinations (Equation (1)), allowing us to link the sound to the particular location the user is interacting with. Our software is implemented in Python running on a laptop, but is simple enough to run on phones and smartwatches as well.

To identify the resonant frequency, we window the 44,100 Hz incoming audio signal in 0.1s non-overlapping segments. We compute the RMS value of each and look for .5s worth of contiguous windows that exceed an empirically determined threshold. We apply Welch's method to extract the power spectrum of the signal [25], and use the strongest frequency as the resonance. We then take the set of cavity/tube (d_s/L_T) combinations available and match the resonant frequency to the Helmholtz-predicted frequencies to determine which hole the user is interacting with. Once a blow is classified, the system executes the action referenced in the configuration file produced by the design software.

To prevent false positives due to background noise such as speech or music, we extract the highest-amplitude frequency for the entire 1.2s segment and compare it with the median frequency of the individual windows. If these values are within 100Hz of each other, there is enough internal consistency in the sound to signal a blow, making Blowhole relatively robust against ambient noise.

Our main implementation is on a laptop computer, using its built-in microphone. We also tested with a LG-R Android smartwatch which transmitted audio data to the same recognition pipeline. Our software runs in Python and uses the scikit-learn library for recognition.

Blowhole Objects: Our software places blowholes into existing models, therefore models that are 3D-printable will remain so with the addition of cavities and openings. Because the cavities are spherical, and most hobbyist-level 3D printers can print up to 45ř of overhang, the models can be produced on most printers with no modification; importantly, no support material is necessary inside the cavities or tubes.

Once a Blowhole-enabled object has been printed, some minor cleanup may be required: with larger spherical cavities, the top of the sphere becomes nearly horizontal, and the printer may produce some “3D printer spaghetti” (a small amount is visible in Figure 1) that can slightly muffle the sound. A simple solution is to simply insert a drill bit of the appropriate size and twist it by hand to quickly remove the strands.

Our software shines in its simplicity and playfulness. We envision Blowhole to be used by low vision users, who have limited interactions with technology. Additionally, Blowhole-enabled objects can be used in classroom settings, specifically for youngsters. Below, we present multiple examples of Blowhole-enabled objects and their applications.

Cell Model. We adapted an existing model of an animal cell[1] to add Blowhole tags to the different parts of the cell (Figure 3). When the tags are activated, the listening computer application launches the Wikipedia page for the associated cell component.

Interactive Animals. We printed three different cetaceans: a dolphin[2], a whale[3], and an orca[4], and adapted the position of the cavity to the location of the animal’s blowhole (Figure 3). When the user blows, the application plays a video about that animal.

Music Controller. A “music box” with raised controls (Figure 3) allows a user to control the flow of music by blowing. Each “button” has a different blowhole underneath it. Our segmentation algorithm described earlier is robust to background sound and in initial testing, its performance was not affected by the sound of the music playing.

Current Work

Our next project is closely related to our work in Blowhole, as we are still interested in the sounds generated by air currents. The difference is, this time, twofold: the air currents used won’t be generated from the user, but from a compressor; and we are interested in exploring touch-based interactions, rather than blow-based.

This next project is based on fundamental fluid dynamics. We designed a mechanism where airflow can easily flow to one side, but when encountered with an obstruction, like a finger, would back down and take another path to the surface, where, using a whistle-like structure, the air would make an identifiable sound.

The objects enhanced with this mechanism would inherit the most fundamental characteristics from Blowhole: there is no post-print assembly required–only provide a steady airflow. This can guarantee that users with any background can enhance their designed objects to interact with them.

Conclusion

In this paper we presented our concept of users fabricating themselves their own smart objects leveraging the sensors present in their mobile devices and smartwatches. Additionally, we presented a representation of this idea in Blowhole,

[1] http://www.thingiverse.com/thing:689381
[2] http://www.thingiverse.com/thing:1121803
[3] http://www.thingiverse.com/thing:232247
[4] http://www.thingiverse.com/thing:665571

a system that allows users to add interactivity to their 3D-printed devices by embedding blowing-activated tags inside them.

REFERENCES

1. Yoh Akiyama and Homei Miyashita. 2016. Fitter: A System for Easily Printing Objects that Fit Real Objects. In *UIST '16: Adjunct Proceedings of the 29th annual ACM symposium on User interface software and technology*. ACM, New York, New York, USA, 129–131.
2. Daniel Ashbrook, Shitao Stan Guo, and Alan Lambie. 2016. Towards Augmented Fabrication: Combining Fabricated and Existing Objects. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. ACM, New York, New York, USA, 1510–1518.
3. Moritz Bächer, Benjamin Hepp, Fabrizio Pece, Paul G Kry, Bernd Bickel, Bernhard Thomaszewski, and Otmar Hilliges. 2016. DefSense: Computational Design of Customized Deformable Input Devices. In *CHI '16: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM Press, New York, New York, USA, 3806–3816.
4. Patrick Carrington, Shannon Hosmer, Tom Yeh, Amy Hurst, and Shaun K. Kane. 2015. "Like This, But Better": Supporting Novices' Design and Fabrication of 3D Models Using Existing Objects. In *iConference*.
5. Xiang 'Anthony' Chen, Stelian Coros, Jennifer Mankoff, and Scott E Hudson. 2015. Encore: 3D Printed Augmentation of Everyday Objects with Printed-Over, Affixed and Interlocked Attachments. In *UIST '15: Proceedings of the 28th annual ACM symposium on User interface software and technology*. ACM Press, New York, New York, USA, 73–82.
6. Xiang 'Anthony' Chen, Jeeeun Kim, Jennifer Mankoff, Tovi Grossman, Stelian Coros, and Scott E Hudson. 2016. Reprise: A Design Tool for Specifying, Generating, and Customizing 3D Printable Adaptations on Everyday Objects. In *UIST '16: Proceedings of the 29th annual ACM symposium on User interface software and technology*. ACM Press, New York, New York, USA, 29–39.
7. Sean Follmer, David Carr, Emily Lovell, and Hiroshi Ishii. 2010. CopyCAD: remixing physical objects with copy and paste from the real world. In *UIST '10: Proceedings of the 23nd annual ACM symposium on User interface software and technology*. ACM, New York, New York, USA, 381–382.
8. Sean Follmer and Hiroshi Ishii. 2012. KidCAD: digitally remixing toys through tangible tools. In *CHI '12: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*.
9. Madeline Gannon, Tovi Grossman, and George Fitzmaurice. 2016. ExoSkin: On-Body Fabrication. In *CHI '16: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. To appear.
10. Timo Götzelmann. 2016. *LucentMaps: 3D Printed Audiovisual Tactile Maps for Blind and Visually Impaired People*. ACM, New York, New York, USA.
11. Chris Harrison, Robert Xiao, and Scott E Hudson. 2012. Acoustic Barcodes: Passive, Durable and Inexpensive Notched Identification Tags. In *UIST '12: Proceedings of the 25th annual ACM symposium on User interface software and technology*. ACM Press, New York, New York, USA, 563–567.

12. Liang He, Gierad Laput, Eric Brockmeyer, and Jon E Froehlich. 2017. SqueezaPulse: Adding Interactive Input to Fabricated Objects Using Corrugated Tubes and Air Pulses. In *International Conference on Tangible, Embedded, and Embodied Interaction*. ACM, New York, New York, USA, 341–350.

13. Herman L F Helmholtz. 1885. *On the Sensations of Tone as a Physiological Basis for the Theory of Music* (2 ed.). Longmans, Green, and Co., London, United Kingdom.

14. Dingzeyu Li, David I W Levin, Wojciech Matusik, and Changxi Zheng. 2016. Acoustic voxels: computational optimization of modular acoustic filters. *ACM Transactions on Graphics (TOG)* 35, 4 (July 2016), 88–12.

15. Huaishu Peng, Jennifer Mankoff, Scott E Hudson, and James McCann. 2015. A Layered Fabric 3D Printer for Soft Interactive Objects. In *CHI '15: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM Press, New York, New York, USA, 1789–1798.

16. Valkyrie Savage, Colin Chang, and Björn Hartmann. 2013. Sauron: embedded single-camera sensing of printed physical user interfaces. In *UIST '13 Proceedings of the 26th annual ACM symposium on User interface software and technology*. New York, New York, USA, 447–456.

17. Valkyrie Savage, Andrew Head, Björn Hartmann, Dan B Goldman, Gautham Mysore, and Wilmot Li. 2015. Lamello: Passive Acoustic Sensing for Tangible Input Components. In *CHI '15: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM Press, New York, New York, USA, 1277–1280.

18. Martin Schmitz, Mohammadreza Khalilbeigi, Matthias Balwierz, Roman Lissermann, Max Mühlhäuser, and Jürgen Steimle. 2015. Capricate: A Fabrication Pipeline to Design and 3D Print Capacitive Touch Sensors for Interactive Objects. In *UIST '15: Proceedings of the 28th annual ACM symposium on User interface software and technology*. ACM Press, New York, New York, USA, 253–258.

19. R Shewbridge, A Hurst, and S K Kane. 2014. Everyday making: identifying future uses for 3D printing in the home. *ACM Conference on Designing Interactive Technology* (2014), 815–824.

20. Lei Shi, Idan Zelzer, Catherine Feng, and Shiri Azenkot. 2016. Tickers and Talker: An Accessible Labeling Toolkit for 3D Printed Models. In *CHI '16: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, New York, New York, USA, 4896–4907.

21. Ronit Slyper and Jessica Hodgins. 2012. Prototyping robot appearance, movement, and interactions using flexible 3D printing and air pressure sensors. In *2012 RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*. IEEE, 6–11.

22. Brandon Taylor, Anind K Dey, Dan Siewiorek, and Asim Smailagic. 2016. Customizable 3D Printed Tactile Maps as Interactive Overlays. In *Proceedings of the 18th International ACM SIGACCESS Conference on Computers and Accessibility*. ACM, New York, New York, USA, 71–79.

23. Stuart Taylor, Alex Butler, Nicolas Villar, and Shahram Izadi. 2009. A reconfigurable ferromagnetic input device. *UIST '09: Proceedings of the 22nd annual ACM symposium on User interface software and technology* (Oct. 2009).

24. Nobuyuki Umetani, Athina Panotopoulou, Ryan Schmidt, and Emily Whiting. 2016. Printone: interactive resonance simulation for free-form print-wind instrument design. *ACM Transactions on Graphics (TOG)* 35, 6 (Nov. 2016), 184–14.

25. P Welch. 1967. The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. *IEEE Transactions on Audio and Electroacoustics* 15, 2 (June 1967), 70–73.