

## MIT Open Access Articles

*Demonstrating InfraredTags: Decoding Invisible 3D Printed Tags with Convolutional Neural Networks*

The MIT Faculty has made this article openly available. **Please share** how this access benefits you. Your story matters.

**Citation:** Dogan, Mustafa Doga, Yotamornsunthorn, Veerapatr, Taka, Ahmad, Zhu, Yunyi, Gupta, Aakar et al. 2022. "Demonstrating InfraredTags: Decoding Invisible 3D Printed Tags with Convolutional Neural Networks."

**As Published:** <https://doi.org/10.1145/3491101.3519905>

**Publisher:** ACM|CHI Conference on Human Factors in Computing Systems Extended Abstracts

**Persistent URL:** <https://hdl.handle.net/1721.1/146136>

**Version:** Final published version: final published article, as it appeared in a journal, conference proceedings, or other formally published context

**Terms of Use:** Article is made available in accordance with the publisher's policy and may be subject to US copyright law. Please refer to the publisher's site for terms of use.



# Demonstrating InfraredTags: Decoding Invisible 3D Printed Tags with Convolutional Neural Networks

Mustafa Doga Dogan  
MIT CSAIL  
Cambridge, MA, USA  
doga@mit.edu

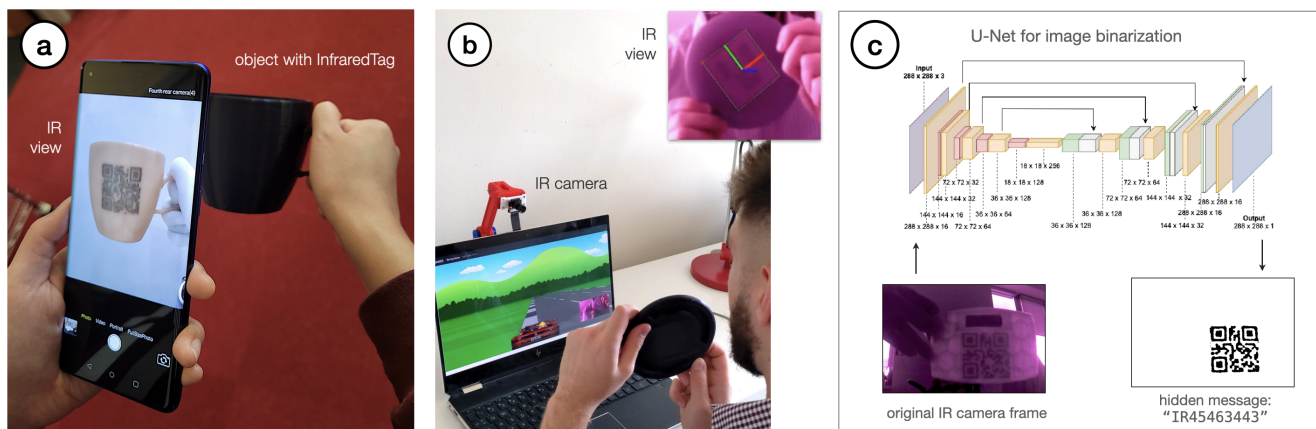
Yunyi Zhu  
MIT CSAIL  
Cambridge, MA, USA  
yunyizhu@mit.edu

Veerapatr Yotamornsunthorn  
MIT CSAIL  
Cambridge, MA, USA  
veerapat@mit.edu

Aakar Gupta  
Facebook Reality Labs  
Redmond, WA, USA  
aakarg@fb.com

Ahmad Taka  
MIT CSAIL  
Cambridge, MA, USA  
ahmadtak@mit.edu

Stefanie Mueller  
MIT CSAIL  
Cambridge, MA, USA  
stefanie.mueller@mit.edu



**Figure 1: InfraredTags are 2D markers embedded unobtrusively into objects using infrared-based 3D printing and imaging tools. Our method allows users to embed (a) QR codes and (b) ArUco markers to store information in objects or to track them. The hidden markers are decoded from infrared camera images (c) using a convolutional neural network based on *U-Net*.**

## ABSTRACT

We demonstrate InfraredTags, which are invisible 2D markers and barcodes that are 3D printed as part of objects. We show how InfraredTags can be decoded using a convolutional neural network (CNN) after being captured by low-cost near-infrared cameras. InfraredTags are formed by printing objects from an infrared-transmitting filament, which infrared cameras can see through, and by having air gaps inside for the tag's bits, which appear at a different intensity in the infrared image. We built a user interface that facilitates the integration of common tags (QR codes, ArUco markers) with the object geometry to make them 3D printable as InfraredTags. Once printed, the tags can be robustly decoded with a U-Net model that we trained using a custom dataset for optimal

binarization and detection. We show how our method enables different applications, such as object tracking and embedding metadata for tangible interactions and augmented reality.

## CCS CONCEPTS

• **Human-centered computing** → *Human computer interaction (HCI)*.

## KEYWORDS

unobtrusive tags; barcodes; 3D printing; infrared imaging; computer vision; augmented reality; tracking

## ACM Reference Format:

Mustafa Doga Dogan, Veerapatr Yotamornsunthorn, Ahmad Taka, Yunyi Zhu, Aakar Gupta, and Stefanie Mueller. 2022. Demonstrating InfraredTags: Decoding Invisible 3D Printed Tags with Convolutional Neural Networks. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts (CHI '22 Extended Abstracts)*, April 29-May 5, 2022, New Orleans, LA, USA. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/3491101.3519905>

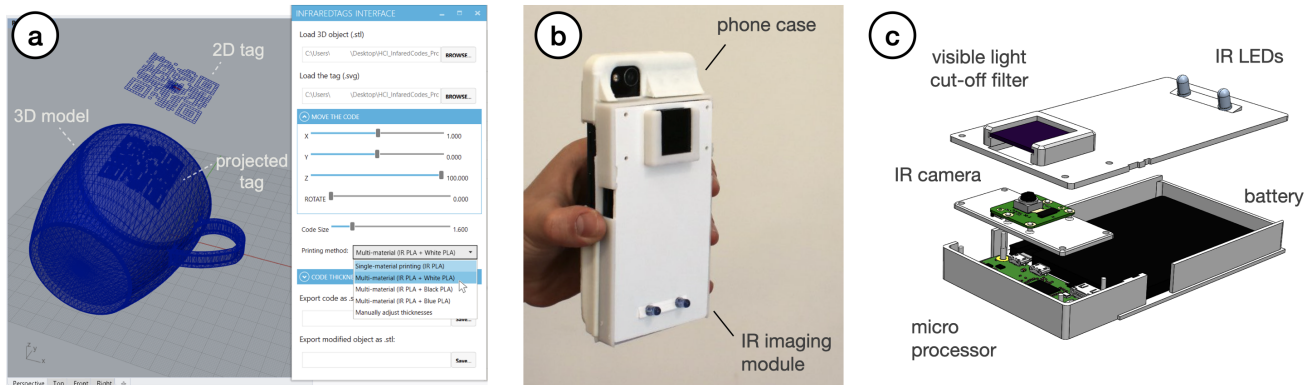
Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

CHI '22 Extended Abstracts, April 29-May 5, 2022, New Orleans, LA, USA

© 2022 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-9156-6/22/04.

<https://doi.org/10.1145/3491101.3519905>



**Figure 2: InfraredTags workflow. (a) A 2D tag is projected and embedded into the desired object using our UI. (b) The optional IR imaging module can be attached to existing mobile devices to read the tags. (c) The module's hardware components.**

## 1 INTRODUCTION

The ability to embed unobtrusive tags in 3D objects while they are being fabricated is of increasing relevance due to its many applications in augmented and virtual reality (AR/VR), packaging, tracking logistics, and robotics [2].

In the last decade, researchers have investigated several ways to identify objects using features imperceptible to the naked eye [3]. One method to accomplish this is to leave air gaps inside the object that represent the bits of a tag. For instance, *AirCode* [8] embeds air gaps underneath the surface of 3D printed objects and uses scattering of projected structured light through the material to detect where the air gaps are located. *InfraStructs* [9] also embeds air gaps into the object but scans it in 3D using terahertz imaging, which can penetrate better through material than visible light.

While both of these methods can embed tags inside 3D objects, they require complex hardware setups (e.g., a projector-camera setup as in *AirCode*), expensive equipment (e.g., a terahertz scanner as in *InfraStructs*), and long imaging time on the order of minutes. To address these issues, we propose a new method that combines air gaps inside the 3D printed structure with infrared transmitting filament. This makes the object semitransparent, and the air gaps are detectable when viewed with an infrared camera. Thus, our method only requires a low-cost infrared imaging module, and because the tag is detected from a single frame, scanning can be achieved much faster.

In this paper, we present InfraredTags, a method to embed markers and barcodes in the geometry of the object that does not require complex fabrication or high-cost imaging equipment [5]. We accomplish this by using off-the-shelf fused deposition modeling (FDM) 3D printers and a commercially available infrared (IR) transmitting filament [1] for fabrication, and an off-the-shelf near-infrared camera for detection. The main geometry of the object is 3D printed using the IR filament, while the tag itself is created by leaving air gaps for the bits. Because the main geometry is semitransparent in the IR region, the near-infrared camera can see through it and capture the air gaps, i.e., the marker, which shows up at a different intensity in the image. The contrast in the image can be further improved by dual-material 3D printing the bits from an infrared-opaque filament instead of leaving them as air gaps. Our method

can embed 2D tags, such as QR codes and ArUco markers, and can embed multiple tags within the object, which allows for scanning from multiple angles while tolerating partial occlusion. To detect InfraredTags with conventional smartphones, we added near-infrared imaging functionality by building a compact module that can be attached to existing mobile devices.

To enable users to embed the tags into 3D objects, we created a user interface that allows users to load tags into the editor and position them at the desired location. The editor then projects the tags into the 3D geometry to embed them with the object geometry. After fabrication, when the user is taking a photo with our imaging module, our custom image processing pipeline detects the tag by increasing the contrast to binarize it accurately. This enables new applications for interacting with 3D objects, such as remotely controlling appliances and devices in an augmented reality (AR) environment, as well as using existing passive objects as tangible game controllers.

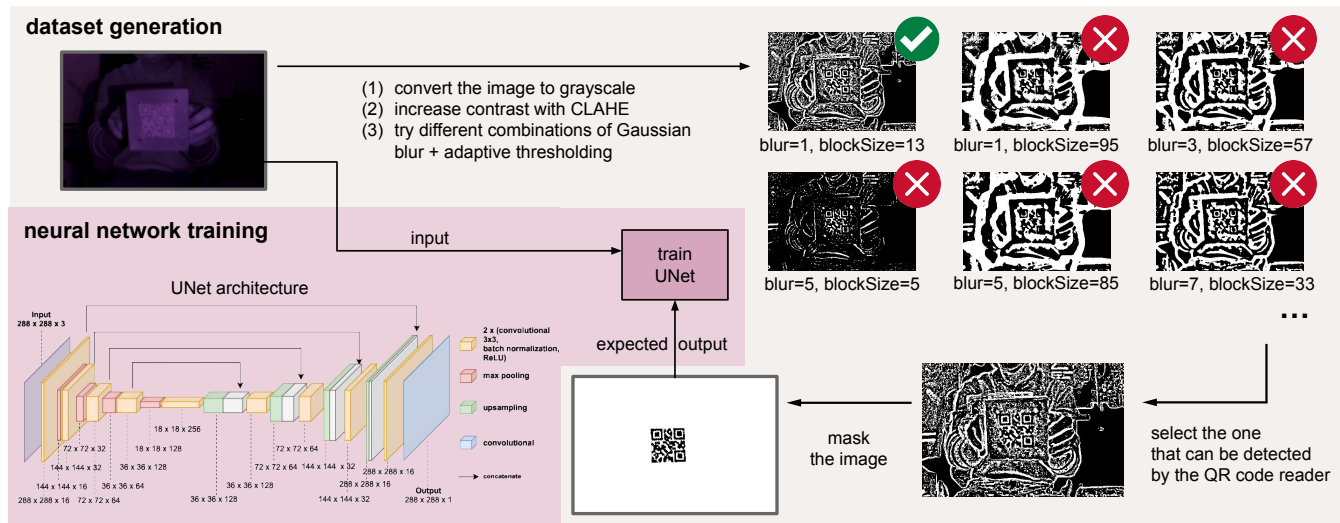
## 2 INFRARED TAGS WORKFLOW

InfraredTags are embedded such that the objects appear opaque and unmodified under visible light but reveal the tag under near-infrared light. We accomplish this by 3D printing the main geometry of the object using an infrared-transmitting filament, while the tag itself is created by leaving air gaps for the bits. Because the main geometry is semitransparent in the infrared region, the near-infrared camera can see through it and capture the air gaps, i.e., tag, which shows up at a different intensity in the image. We refer to the infrared-transmitting filament as *infrared filament* or *IR filament* in the remainder of the paper.

We next explain how users can embed codes into 3D objects using our user interface and then discuss our custom add-on for mobile devices and the corresponding image processing pipeline for tag detection.

### 2.1 User Interface for Encoding InfraredTags

**Import and Position Tags:** The user starts by loading the 3D model (.stl file) into our user interface, which is a custom add-on to an existing 3D editor (*Rhinoceros 3D*). Next, users import the



**Figure 3: The workflow of dataset generation and training. The input is a frame extracted from a video recording. Then, image processing is performed on the input image with multiple sets of parameters. The image in which the QR code can be detected by the QR code reader selected, and masked such that the image has only the QR code on a white background. The resulting image is the expected output. Finally, the input and expected output are used to train the *U-Net* model.**

tag as a 2D drawing (.svg) into the editor, which loads the marker into the 3D viewport. The marker is then automatically projected onto the surface of the 3D geometry (Figure 2). Users can move the code around in the viewport and scale it to place it in the desired location on the object.

**Select Printing Method:** In the user interface, users can then select the printing method, i.e., if they want to fabricate the object with single material (IR-PLA only) or dual-material printing (IR-PLA + regular PLA). The interface then modifies the geometry to accommodate the selected printing method. The user can then slice both files with the 3D printer’s slicing software and print the object.

## 2.2 IR Imaging Module for Reading the Tags

InfraredTags can be read with digital devices that have an infrared camera attached to them. Even conventional USB webcams for personal computers can be used for this purpose by manually removing their infrared cut-off filter<sup>1</sup>.

Today, several recent smartphones already come with an IR camera, e.g., *OnePlus 8 Pro* (Figure 1a). However, most of these phones do not give developers IR camera access for non-native applications. To make our method compatible independent of the platform, we built an additional imaging add-on that can easily be attached to any existing mobile phone.

**Attaching the IR camera module:** As shown in Figure 2b and c, our add-on contains an infrared camera (model: *Raspberry Pi NoIR*). This camera can see infrared light since it has the IR cut-off filter removed that normally blocks IR light in regular cameras. Additionally, to remove the noise from visible light and improve

detection, we added a visible light cut-off filter<sup>2</sup>, as well as 2 IR LEDs (940nm) which illuminate the object when it is dark. This add-on has two 3D printed parts: a smartphone case from flexible TPU filament that can be reprinted based on the user’s phone model, and the imaging module from rigid PLA filament that can be slid into this case.

**Detecting the Tag:** To detect the tag, users open the InfraredTags detection application on their mobile phones and point the camera to their object. The application shows the phone camera’s view, which is what the user sees with their eyes instead of the IR view (Figure 1a). This way, more information can be overlaid on top of the regular view for AR applications. Under the hood, the imaging module continuously streams the images to our image processing server. If the server detects any tags, it sends the location and the encoded message to the smartphone app to show to the user.

## 3 DETECTING INFRARED TAGS VIA CONVOLUTIONAL NEURAL NETWORK

A problem with IR images is insufficient resolution. Hence, image processing is required to increase contrast of the image. The state-of-the-art IR image processing implementation involves the following three steps. First, the IR image is converted to grayscale, and a contrast limited adaptive histogram equalization (CLAHE) filter is applied to improve local contrast. Second, a Gaussian blur is applied to reduce the noise caused by CLAHE. Third, the image is binarized by Gaussian adaptive thresholding and then used as input to the QR code reader (*Dynamsoft*<sup>3</sup>). However, each frame, depending on the location (distance) of the marker, requires a different set of parameters for these filters. This implementation tries

<sup>1</sup><https://publiclab.org/wiki/webcam-filter-removal>

<sup>2</sup><https://www.edmundoptics.com/p/1quot-x-1quot-optical-cast-plastic-ir-longpass-filter/5421/>

<sup>3</sup><https://www.dynamsoft.com/barcode-reader/overview/>



**Figure 4: Controlling appliances using mobile AR. The user points at (a) the speakers to adjust the volume, and the (b) thermostat to adjust the temperature. The IR camera on the phone identifies the appliances via the embedded QR codes. (c) Pairing a phone with a WiFi router: the SSID is visible from all sides but the password is visible only from its bottom.**

all possible sets of parameters until the ones that make the QR code readable are found, and therefore it takes a long time to finish. The long processing times make it impractical for real-time applications, which presents a major limitation of this technology. To reduce processing time, we circumvent parameter tuning for each filter, and use a convolutional neural network (CNN) to convert IR images, which originally have RGB channels, into binary images (Figure 3). This reduces the detection time from several seconds to approximately 0.35 seconds per frame.

### 3.1 Architecture

We use the *U-Net* architecture for this task. The architecture of *U-Net* is comprised of 2 parts: an encoder and a decoder. In our task, an encoder takes an IR image as the input, captures information from the image, and returns a feature map. Then a decoder receives the feature map from the encoder and returns a binary image as the output.

### 3.2 Dataset Generation

Datasets used come from IR video recordings of an object with an InfraredTag QR code. We also use a visible light cut-off filter during the video recording. To ensure generalizability of the algorithms, the following factors are varied across video recordings. The QR code is moved around to vary the QR code position in each frame. The recording location (background) is changed in each recording to vary background noise. The light source is placed behind the camera in some recordings or in front of the camera in others.

We extract input frames from the video recordings and perform image processing on each one to get the final binary image. We discard the frames in which a QR code is not present. The QR code reader returns not just the data stored in the QR code but also the coordinates of the four corners of the marker. The four coordinates are used to create a polygon mask. We use the mask on the generated binary images such that the images have only QR codes on a white background. The masked images are the expected output of the binarization CNN. As usual, the dataset is split into training, validation, and test sets. We also use data augmentation

techniques, such as translation, rotation, and color jitter, to increase the variance and diversity in the training data.

## 4 APPLICATIONS

We demonstrate how InfraredTags enable different use cases for interactions with objects and devices, storing data in them, and tracking them for sensing user input.

### 4.1 Distant Augmented Reality (AR) Interactions with Physical Devices

InfraredTags can be embedded into physical devices and appliances to identify them individually through the embedded unique IDs and show the corresponding device controls that can be directly manipulated by the user.

In the application shown in Figure 4a and b, a user points their smartphone camera at the room and smart home appliances are identified through their InfraredTags, which are imperceptible to the human eye. A control menu is shown in the AR view, where the user can adjust the volume of the speaker or set a temperature for the thermostat. InfraredTags could also allow multiple appliances of the same model (e.g., multiple smart speakers or lamps) in the room to be identified individually, which is not possible with standard vision-based object classification approaches.

*Multiple tags on a single object for spatial awareness:* Furthermore, InfraredTags enable *multiple tags* to be embedded in the same object. For example, Figure 4c illustrates how the front, side, and top faces of a WiFi router only have its network name (SSID) information, whereas its bottom also shows the password information, which can automatically pair the router to the phone. This enables quick pairing and authentication with devices without users having to type out complex character strings, while maintaining the physical use metaphors, such as the paper slip containing the password typically attached to the base of the router. While we demonstrate this application for mobile AR, InfraredTags could also enable lower friction, distant interactions with physical devices for head-mounted AR glasses.

## 4.2 Embedding Metadata about Objects

Spatially embedding metadata or documentation information within the object itself can provide richer contextualization and allow information sharing [4, 6]. For example, we can embed the object's fabrication/origin link (e.g., a shortened *Thingiverse* URL) as an InfraredTag for users to look up in case they would like to get more information from its creator or 3D print it themselves as shown in Figure 1. Other types of metadata that could be embedded include user manuals, expiry dates, date of fabrication, materials used to fabricate the object, weight, or size information.

## 4.3 Tangible Interactions: Use Anything as a Game Controller

Because fiducial markers can be embedded as InfraredTags, they can be used to track the object's movement. Thus, any passive object can be used as a controller that can be held by users when playing video games.

Figure 1b shows a 3D printed wheel with no electronics, being used as a game controller. The wheel contains an ArUco marker InfraredTag which is used to track the wheel's location and orientation. Even though the wheel is rotationally symmetric, the infrared camera can see the square marker inside and infer the wheel's position and orientation. Our method does not require any electronics as opposed to other approaches [7, 10].

## REFERENCES

- [1] 3dk.berlin. 2021. PLA Filament IR-Black. <https://3dk.berlin/en/special/115-pla-filament-ir-black.html>
- [2] Mustafa Doga Dogan, Patrick Baudisch, Hrvoje Benko, Michael Nebeling, Huaishu Peng, Valkyrie Savage, and Stefanie Mueller. 2022. Fabricate It or Render It? Digital Fabrication vs. Virtual Reality for Creating Objects Instantly. In *Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, 5. <https://doi.org/10.1145/3491101.3516510>
- [3] Mustafa Doga Dogan, Steven Vidal Acevedo Colon, Varnika Sinha, Kaan Aksit, and Stefanie Mueller. 2021. SensiCut: Material-Aware Laser Cutting Using Speckle Sensing and Deep Learning. In *Proceedings of the 34th Annual ACM Symposium on User Interface Software and Technology*. ACM, Virtual Event USA, 15. <https://doi.org/10.1145/3472749.3474733>
- [4] Mustafa Doga Dogan, Faraz Faruqi, Andrew Day Churchill, Kenneth Friedman, Leon Cheng, Sriram Subramanian, and Stefanie Mueller. 2020. G-ID: Identifying 3D Prints Using Slicing Parameters. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–13. <https://doi.org/10.1145/3313831.3376202>
- [5] Mustafa Doga Dogan, Ahmad Taka, Michael Lu, Yunyi Zhu, Akshat Kumar, Aakar Gupta, and Stefanie Mueller. 2022. InfraredTags: Embedding Invisible AR Markers and Barcodes Using Low-Cost, Infrared-Based 3D Printing and Imaging Tools. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New Orleans LA USA, 9. <https://doi.org/10.1145/3491102.3501951>
- [6] Omid Etehad, Fraser Anderson, Adam Tindale, and Sowmya Somanath. 2021. Documented: Embedding Information onto and Retrieving Information from 3D Printed Objects. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. ACM, Yokohama Japan, 1–11. <https://doi.org/10.1145/3411764.3445551>
- [7] Jack Forman, Mustafa Doga Dogan, Hamilton Forsythe, and Hiroshi Ishii. 2020. DefeXtiles: 3D Printing Quasi-Woven Fabric via Under-Extrusion. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. ACM, Virtual Event USA, 1222–1233. <https://doi.org/10.1145/3379337.3415876>
- [8] Dingzeyu Li, Avinash S Nair, Shree K Nayar, and Changxi Zheng. 2017. AirCode. *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology - UIST '17 (2017)*. <https://doi.org/10.1145/3126594.3126635>
- [9] Karl D. D. Willis and Andrew D. Wilson. 2013. InfraStructs: fabricating information inside physical objects for imaging in the terahertz region. *ACM Transactions on Graphics* 32, 4 (July 2013), 1–10. <https://doi.org/10.1145/2461912.2461936>
- [10] Junichi Yamaoka, Mustafa Doga Dogan, Katarina Bulovic, Kazuya Saito, Yoshihiro Kawahara, Yasuaki Kakehi, and Stefanie Mueller. 2019. FoldTronics: Creating 3D

Objects with Integrated Electronics Using Foldable Honeycomb Structures. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 1–14. <https://doi.org/10.1145/3290605.3300858>