Enabling non-experts to author tangible interactions

Carlos Tejada
University of Copenhagen
www.carlosetejada.com
@ctejada10

About Me

About Me

- BS Systems Engineering • 2012
  Pontificia Universidad Católica Madre y Maestra
  Santiago, Dominican Republic
- MS Information Science • 2016
  Rochester Institute of Technology
  Rochester, New York, USA
- PhD Computer Science • 2020
  University of Copenhagen
  Copenhagen, Denmark

How to create interactive devices?

- The rise of smartphones and touchscreen applications has drawn focus from the physical world and its dedicated physical user interfaces (controllers, musical instruments, medical devices)

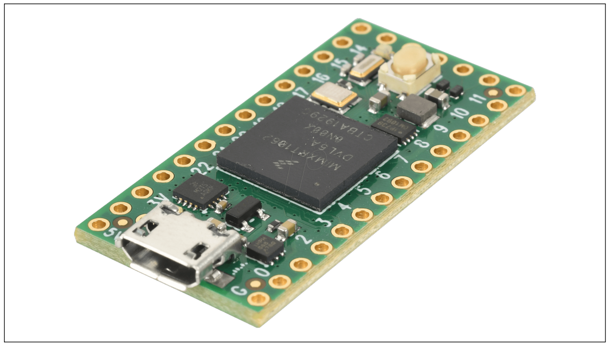
Why is physicality important in interactions?

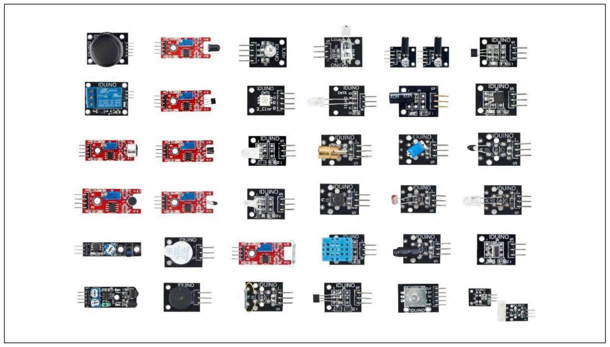
- Physical interfaces have some intrinsic benefits over 'virtual' ones, like tactile feedback, and high performance manipulation

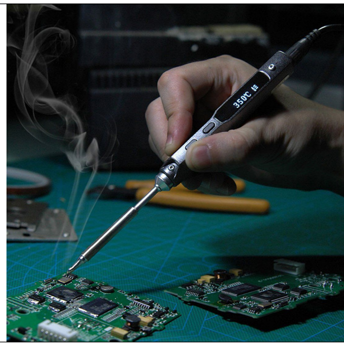
- Physical interfaces have some intrinsic benefits over 'virtual' ones, like tactile feedback, and high performance manipulation
- Gamers prefer physical input for speed and performance, musicians for virtuosity and control

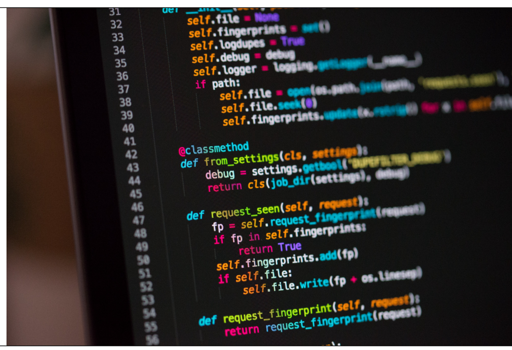
- Screen are very variable and customizable but they lack these features
- Maybe skip

How to create interactive devices?

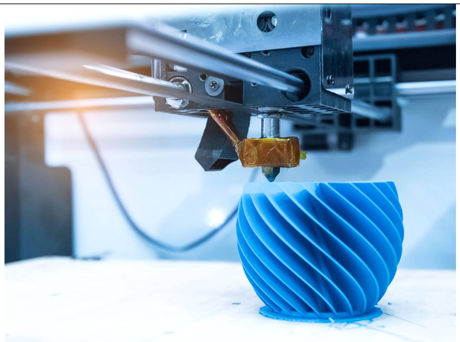


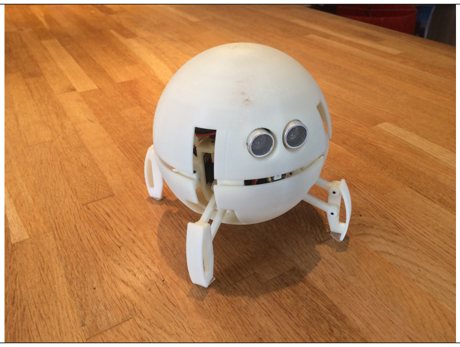






self.file = None
self.fingerprints = set()
self.logdupes = True
self.debug = debug
if path:
self.file = open(os.path.join(path,
self.file.seek(0)
@classmethod
def from_settings(cls, settings):
debug = settings.getbool(
return cls(job_dir(settings), debug)
def request_seen(self, request):
fp = self.request_fingerprint(request)
if fp in self.fingerprints:
return True
self.fingerprints.add(fp)
if self.file:
self.file.write(fp + os.linesep)
def request_fingerprint(self, request):
return request_fingerprint(request)





Not trivial

- Might seem easy for us
  - Electrical engineering/Computing background

Not trivial

- Users need to have:
  - Electrical engineering expertise.
  - Programming knowledge.
  - Time to test and debug.

- EE expertise to choose, and assemble components
- Programming background to tell the MCU what to do
- Lots of time to test
- Might be trivial to us

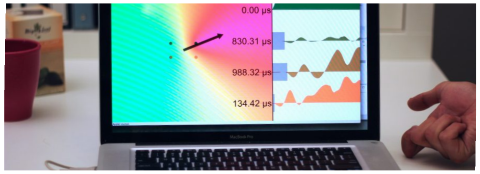
- A lot of people are experts in their own field.
- We can't ask them to become experts in EE, or computing to do this stuff
- How can we use our expertise to help them?

Research

Answer must be research

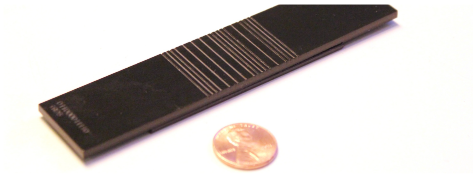
Scratch Input: Harrison et al.
UIST '08

Interact with tabletops by scratching



Toffee: Xiao et al.
MobileHCI '14

Or by knocking

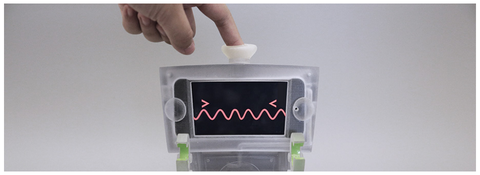


Acoustic Barcodes: Xiao et al.
UIST '12

Or creating grooves to create a makeshift barcode

Touch & Activate: Ono et al.
UIST '13

Other efforts use active acoustic techniques to enable touch, and force touch sensing



Acoustruments: Laput et al.
CHI '15

Or even more complex applications

Limitations

- Super interesting and exciting techniques
- Calibration of machine learning models:
    - Get clean data
    - Label it
    - Train ML learning models
    - Test

Limitations

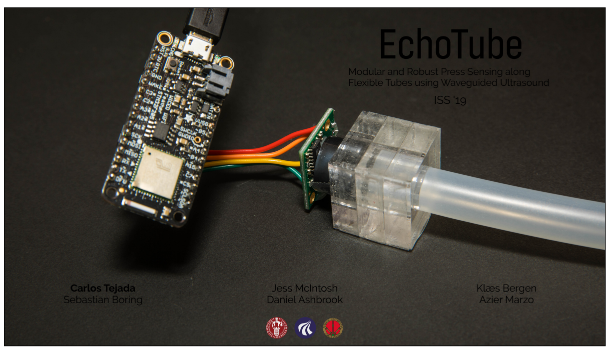
- Become data scientists:
  - Get clean data.
  - Label it.
  - Train a machine learning model.
  - Evaluate its performance

Engineering expertise is required

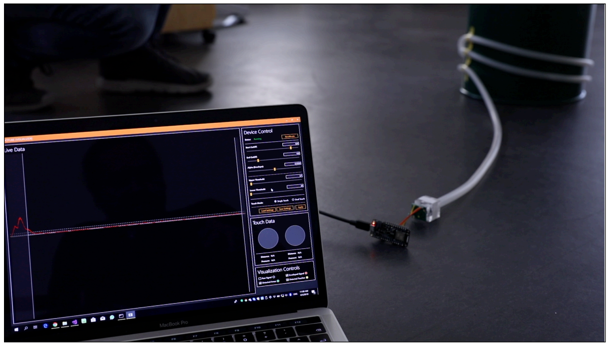
- Need to know what you're doing

Enable domain-experts to create interactive devices

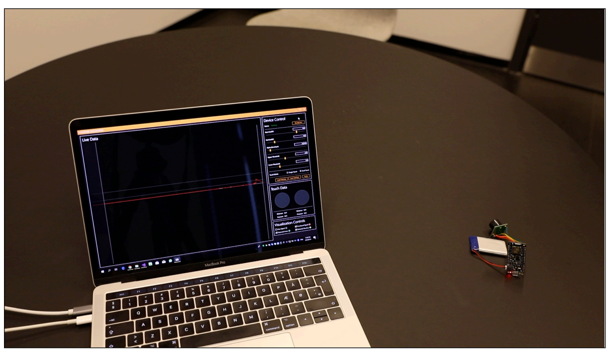
- How to help people lack these domain knowledge to create interactive devices?
- Add interactivity to already existing objects, which is what i try to do with the first project i want to show you guys



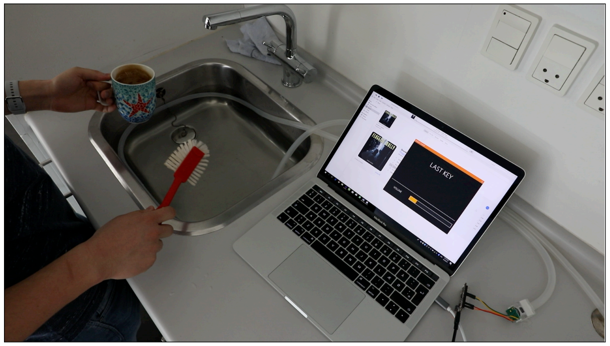
ET allows to quickly and robustly deploy press sensing on surfaces
In the video you'll see next, you'll see a rubber tube being pressed, and a computer identifying the interactions



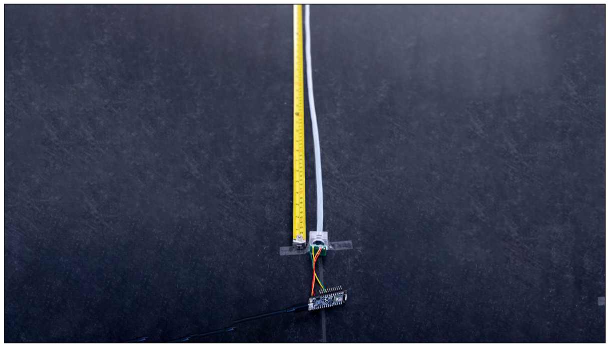
- We sense press interactions in rubber tubes using waveguided ultrasonic waves.
- EchoTube has some cool properties:



- Echotube is easy to deploy
- Made up of inexpensive components

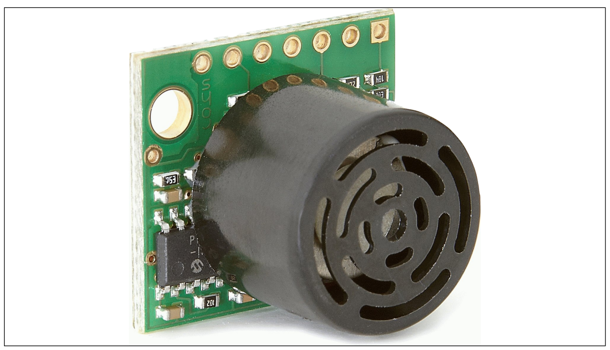


- EchoTube is robust: electronics are isolated where the interaction is taking place

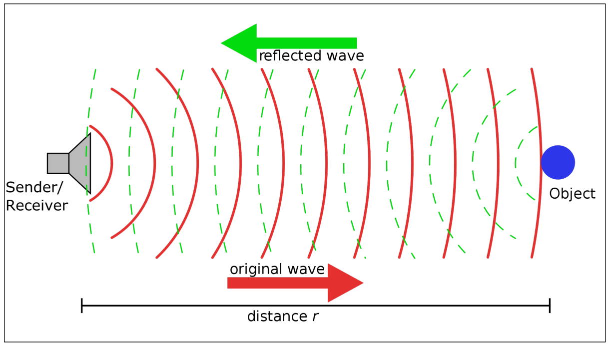


- EchoTube works at long distances
- Up to 4m

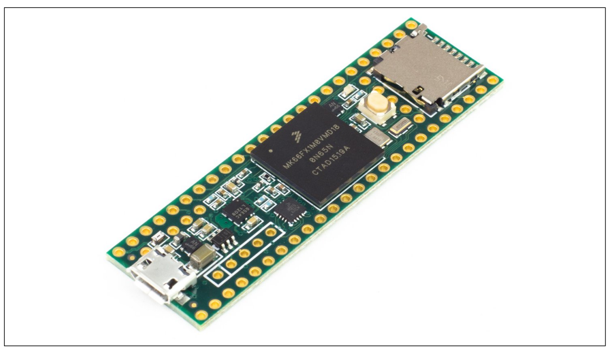
- EchoTube starts with a tube.
- Off the shelf, inexpensive rubber tubes.



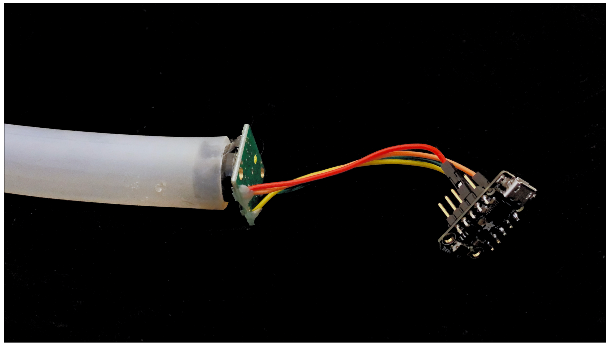
- To do sensing on these tubes, we attach an USRF.
- Can be bought off the shelf.
- Are used to get the distance to objects and obstacles.



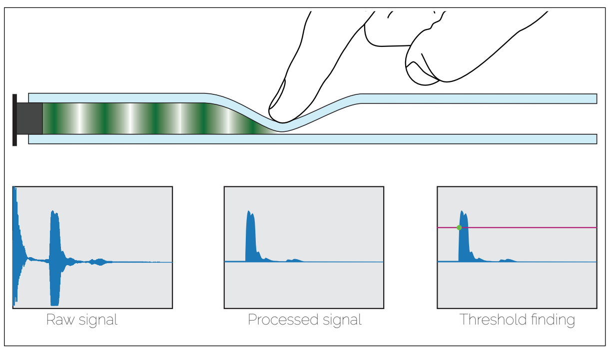
- Pulse gets sent out from the transducer
- Bounces off objects and gets reflected back
- By the delay from the time it was emitted to the time it returns, it can calculate the distance to the object.
- Enclose them in the tube gets us:
    - Constraint the sensing to the tube
    - Enable routing through corners and objects surfaces.



- Last piece is a micro controller
    - Drives and samples the rangefinder.
    - Sends the data to the computer for processing.
    - Even though the RF gives a digital output of the distance to the first obstacle it encounters, we sample the raw signal instead to get more precise info on what is happening, and enable our limited dual-touch capabilities.



- First, we connect all these things together, something like this.
- Deploy the tube to where we'd like to add sensing



- When somebody or something touches the tube
    - Measurable change in the signal we get back from the transducer
    - We do some simple filtering on the signal
    - Identify peaks that surpass a threshold
    - We calculate the delay it took for the sound to be reflected that.
    - Use that delay to calculate the distance.
    - With the locations of the touch, we can then do some actions like the ones showed on the videos.

Performance Testing

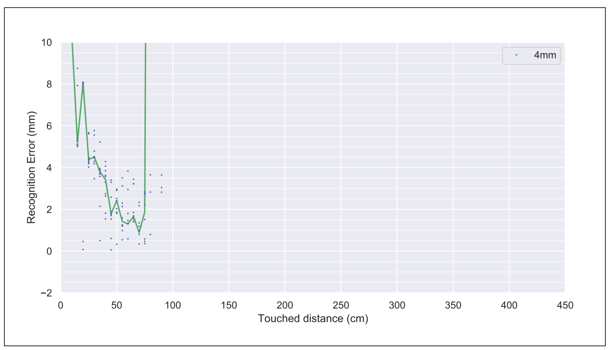
- Questions:
  - How accurately we could detect single touches on a tube?
  - How did this technique perform in different tubes?
  - How many touches we could detect?

How we tried to find out?

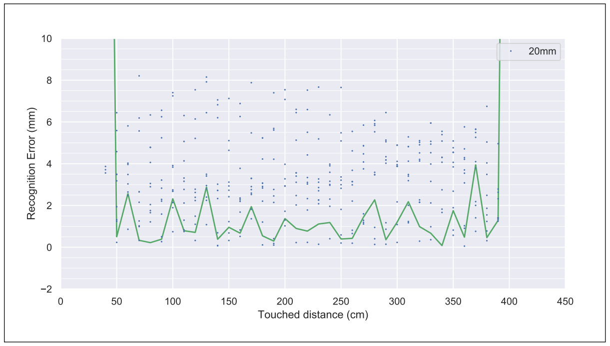
Introduce coupling

- Laid the tube straight on a surface and the other end open.
- Touched at 5 cm intervals.
- Fully compressed the tube.
- Repeated 10 times per location, with all 5 tubes

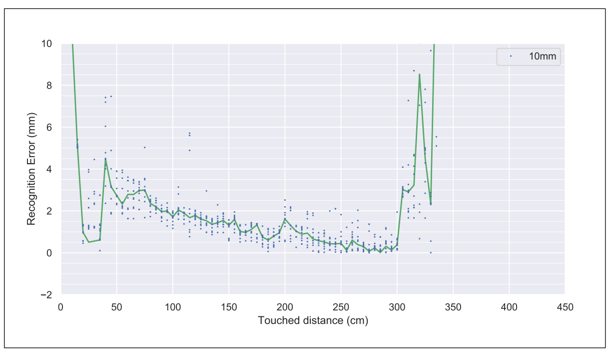
- Error in mm, versus distance in cm
- Blue dots represent each individual touch, green lines represent the mean for the location.
- Some stuff to highlight:
- Tubes at scale



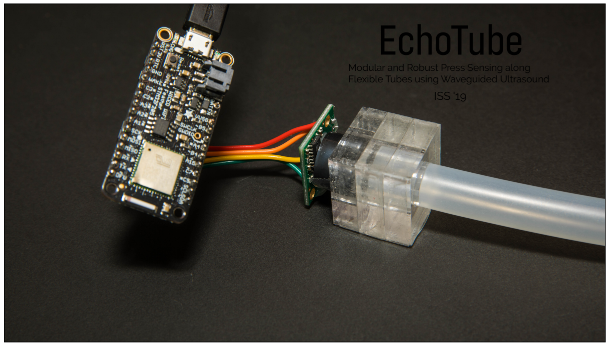
- Smaller tubes have shorter ranges.
- Suitable for smaller scale applications



- Larger tubes have the longest ranges, but are less precise.
- Suitable for large scale applications where the exact position of the interaction is not of interest.



- Best case scenario is obtained with 10mm tubes
- Up to 3 meters of accurate sensing



- ET allows to deploy press sensing easily

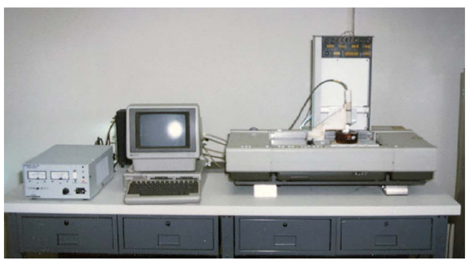
Limitations

- Smaller tubes don't perform so well
- Even though 4mm might seem small, it's pretty big
- Lost of at least the first 20 cm due to the USRF ringing

Scale is a problem

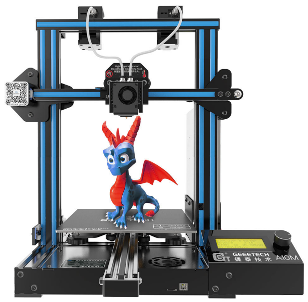
- Good for larger scale deployments, bad for smaller, hand-held stuff

Fabricate interactive objects



The first 3D printer: The SLA-1 from 3D Systems

- 3D-printers have come a long way since the day of the SLA-1
- While before
- Reserved for research and industry workers



- Accessible and reliable enough for hobbyists and enthusiasts to use

From passive to interactive

- Researchers have proposed a lot of techniques

Assembly

Calibration

Use limitations

- Need domain knowledge to implement these techniques
- Comes in the form of assembly, calibration or just general limitations on how the object can be used
- Putting these amazing techniques out of the reach for ordinary users

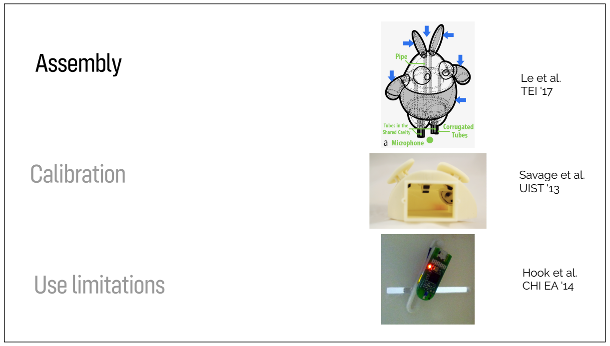
---

Assembly

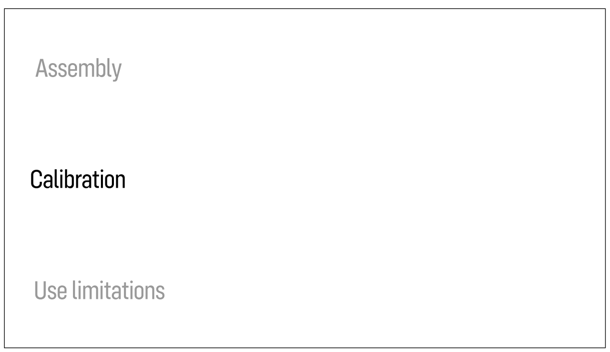
Calibration

Use limitations

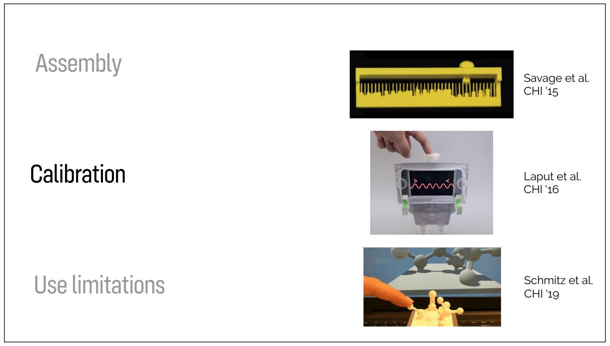
- Some efforts require the designer to assemble either physical parts, or electronic circuits to enable interactions



- Varying from assembling circuits inside a printed shell
- To complex fabrication techniques like silicon molding



- Other techniques ask for the designer something arguably harder: collect clean data, label it and train machine learning models



- Require calibration of machine learning models, either per object or per user

- Others generally limit how they can be used

Assembly

Hudin et al.
CHI EA '15

Calibration

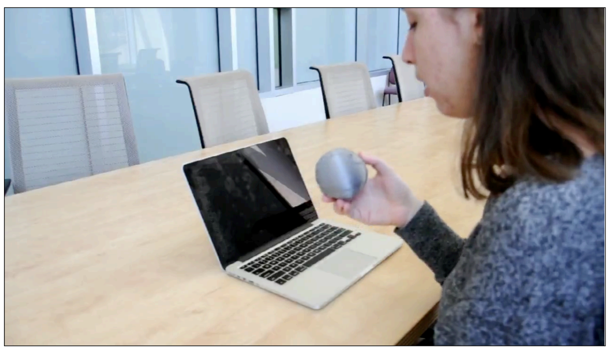
Use limitations

Shi et al.
CHI '16

- Either limiting where the interaction must take place or by adding disruptive features on the objects geometry

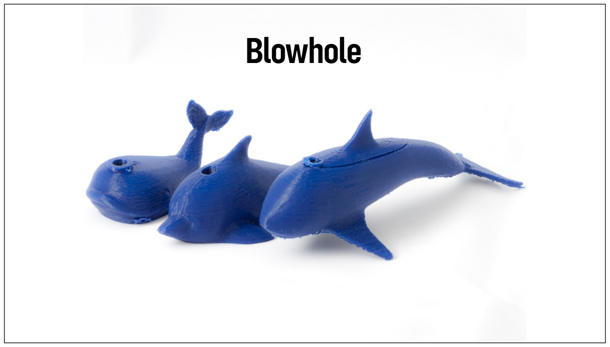
How to fabricate interactive objects in a designer-friendly way?

Blowhole, a system to add interactivity to 3D printed objects by embedding resonant cavities into them with openings to the surface
When we blow on these cavities, we generate a unique sound based on the cavity configuration
You'll now see a video of it at work

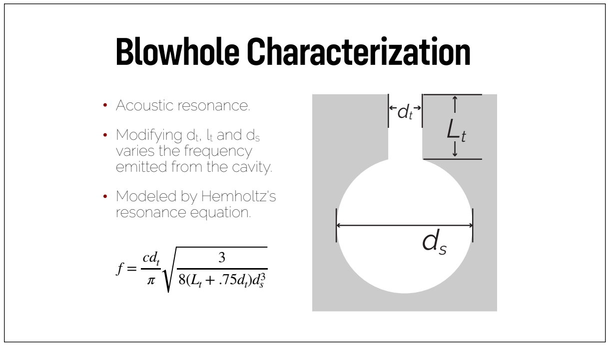


short demo video

- BH is a system to acoustically tag 3D-objects
- Embed resonant cavities inside 3D-printed objects
- By varying the size of the cavity, we vary the pitch of the sound that these emit when gently blown



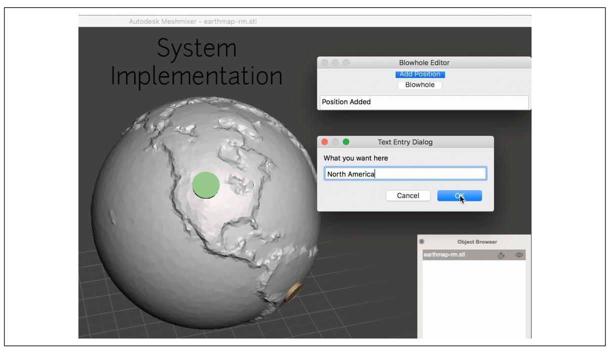
- Blowhole-enabled objects do not require advanced printing techniques, are designed to be generated using an off-the-shelf printer, do not require assembly, little size constraints.
- The interactivity is based on the principle acoustic resonance: the same property that makes a bottle whistle when you blow on its mouth.



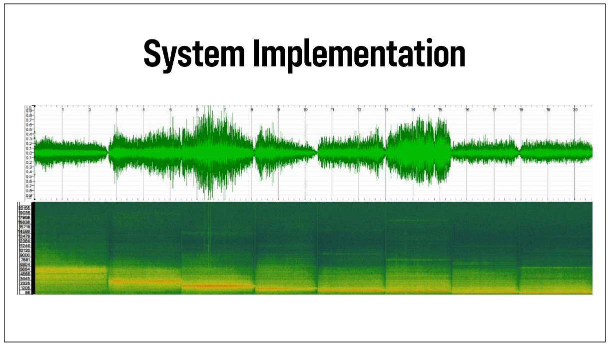
Modeled by the Helmholtz resonance equation
The Helmholtz equation tells us that the resulting frequency from a resonating cavity depends on the diameter of said cavity, the length of the tube that connects it to the surface and the diameter of the opening to the surface.

- We wanted BH to be of the most practical use
- Printed a lot of cylinders containing variations of tube length, sphere diameter

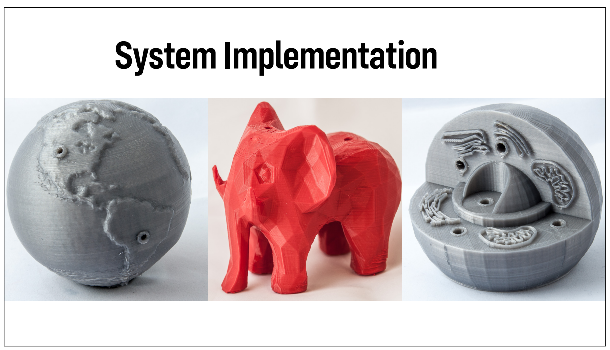
- Recorded 12 participants and obtained the fundamental frequency of their blowing sounds.
- Extracted the fundamental frequencies of each sound using Welch's method
- Print and recognize reliably frequencies ranging from 500 Hz to 5900 Hz



- Blowhole is comprised of three fundamental parts: a design environment and a sound recognition environment.
- Using MeshMixer’s Python API
- The user imports a model and selects the place it wants to tag
    - Adding a corresponding action to this place
- The system optimizes the configuration of the cavities using a backtracking algorithm to avoid collisions.



- Blow sound recognition
  - Captures blowing sounds and gets the fundamental frequency using Welch's method
  - Use the set of cavity sizes and tube lengths to find the combinations that would give us this frequency



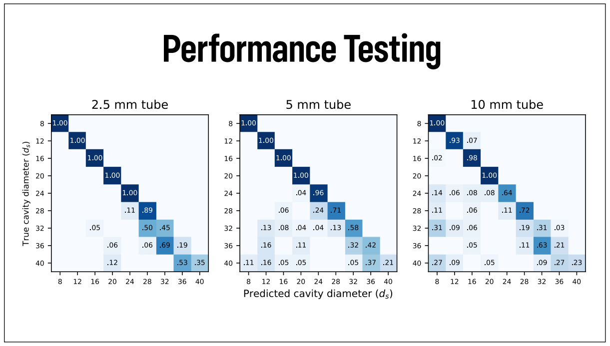
- Printed as a single structure on commodity 3D-printers
- Used 5mm as an opening size for uniformity
- Added a ring feature to distinguish the openings from the objects geometry

Performance Testing

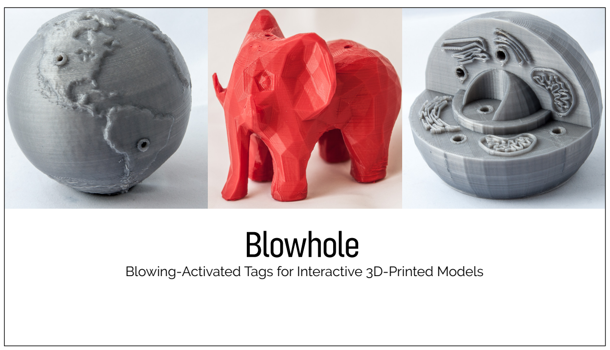
Performance Testing

- Collected 830 blow segments.
- 10 participants.
- Tested both overall and per-user.

- User-dependent
  - Trained with one blow from the user, tested with its remaining blows.
  - 96% accuracy.
- User-independent
  - Held out a user's data for testing, and trained with the other users.
  - 84.8% accuracy.



- Results per tube length
- Larger spheres and larger tubes, accuracy decreases.
- Best performance/versatility is with 2.5mm tubes and 6 spheres from 8-28 -> 98% accuracy.

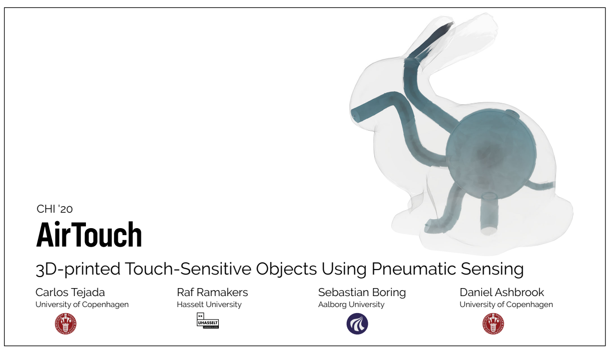


BH enables novices to design and fabricate interactive 3D objects without any assembly of circuits or parts, nor calibrating machine learning models.

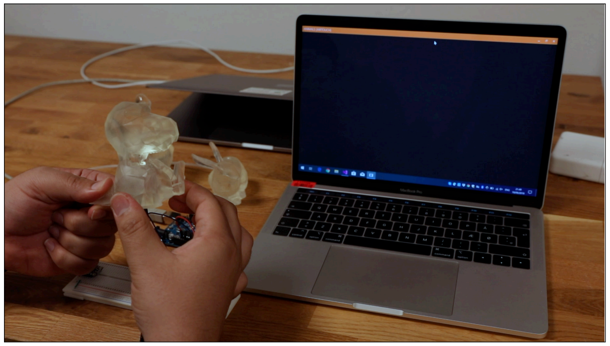
Limitations

Sensitive to environmental noise

Only 6 locations



AT is a technique to fabricate touch sensitive objects without the need of assembling part, circuits or train machine learning models
In the video you'll se now, you'll see how AT is used to identify interactions of objects of varying geometries

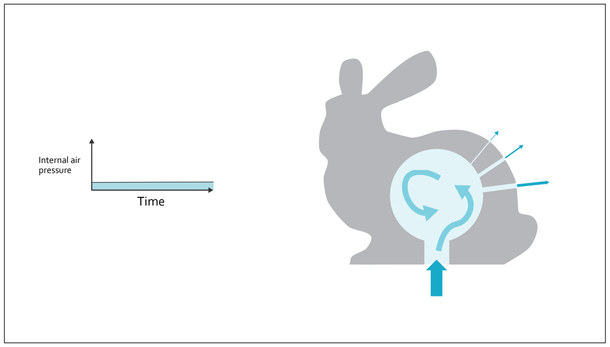


- Because all animals share the same outlet configuration
- Interactions can be identified with the same ML model

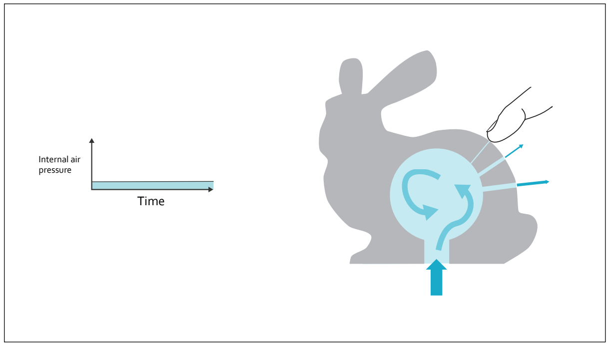
How does it work?

- Makes use of basic principles of fluid dynamics

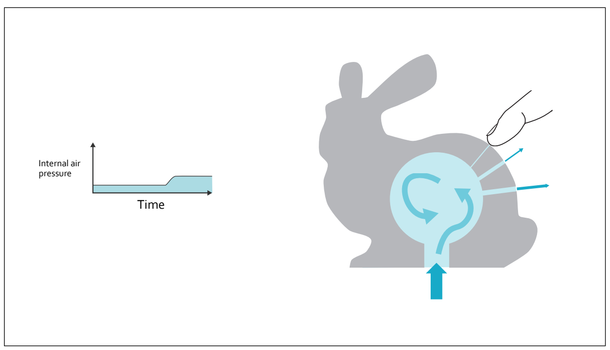
- Familiar example is partially covering the end of a garden hose.
- The more you cover the end of the hose, the faster the water comes out.
- Discharge area is reduced -> internal pressure increases -> water comes out faster



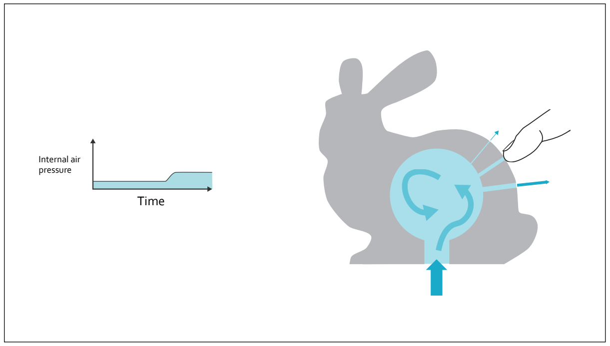
- Adapt that concept to fabricated objects
- Instead of a single, big opening
  - Multiple, small openings



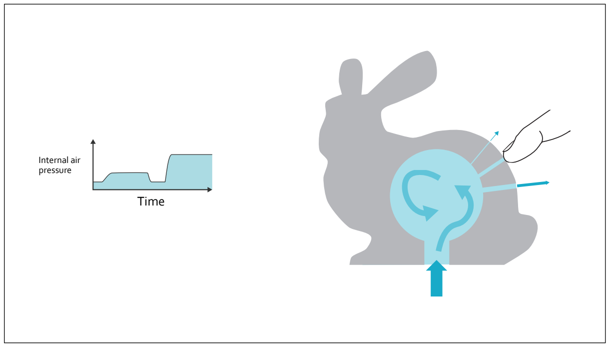
- When we cover an opening, we reduce the discharge area



- Pressure increases, proportionally to the area of the opening we covered.



- Because every opening is unique in size



- The pressure increases are also unique per opening

How does it work?

- Principle of fluid continuity.
- Bernoulli's principle.

$$\Delta P_x = \frac{(\sum A_i)^2 \Delta P}{(\sum A_i - A_x)^2}$$

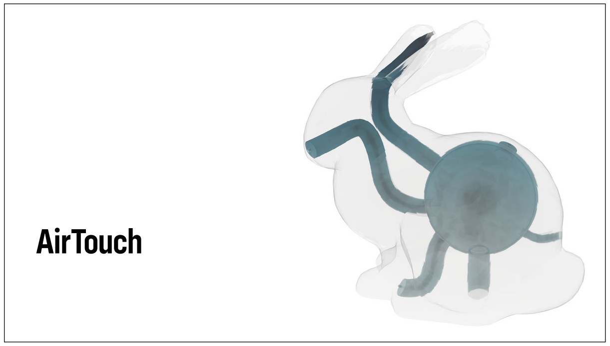
- Change in size of openings → Change in pressure.

- Math that describes this phenomenon
- Combining the principle of fluid continuity tells us that the flow that goes into a system must come out
- and Bernoulli's principle which relates pressure and flow.
- Equation that describes change in pressure given changes in area
- More detail in the paper
- Gist is: change the size of an opening that fluid is passing though, the pressure varies in response.
    - Like we saw with the hose example

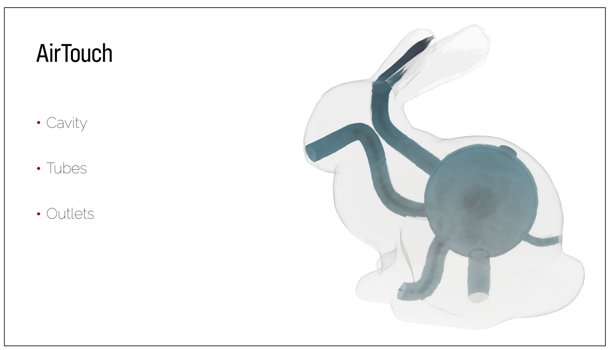
How does it work?

- Assumptions:
    - Incompressible fluids.
    - Perfectly shaped outlets and structures.
- Reality:
    - Air is compressible.
    - 3D-printed objects are not perfect.
    - Complex internal geometries.

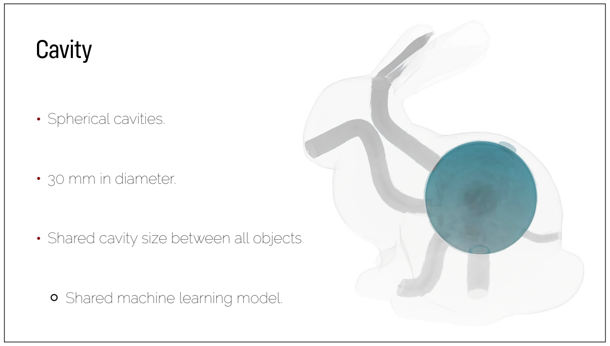
- Problem with this model
- Makes some assumptions that are not present in our scenario
    - Dealing with incompressible fluids, like liquids
    - Geometry is perfect and smooth
- Reality
    - Use air, which is compressible
    - Fabricated objects are not perfect
    - Make use of complex internal geometries, can cause turbulence
- Use our knowledge of this phenomena to inform the creation of pre-trained machine learning models.



- Modify the interior of the object
- Structure that disperses the flow to the surface

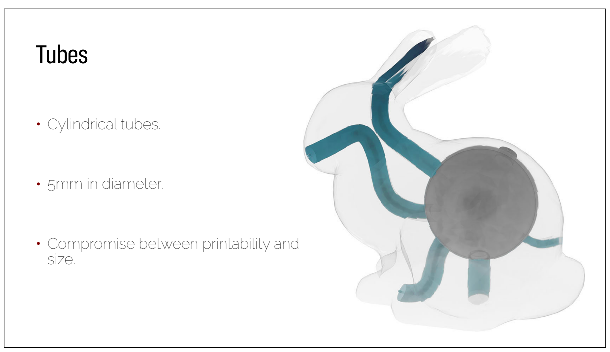


- Parts of the structure

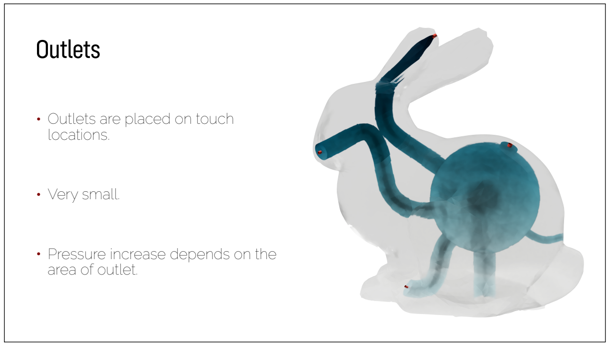


- First thing we do is embed a cavity
- Ø30mm in diameter
    - Small objects
- Standard cavity -> Same pre-trained ML model with objects of different geometries

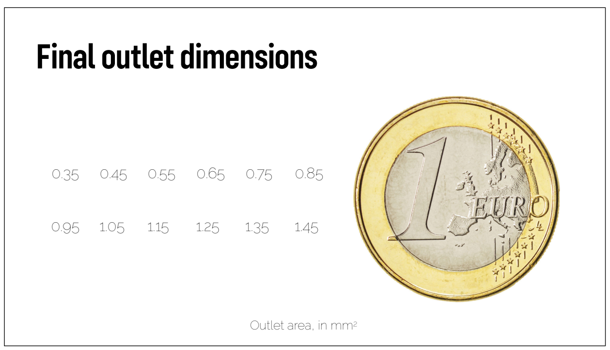
---



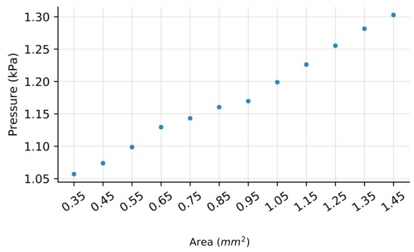
- Cylindrical tubes
    - Connect the cavity to the outlets
- Ø5mm in diameter
    - Smallest size we could print reliably



- Outlets, highlighted in red
- Go wherever you want to add touch sensitivity
- Tiny, don't disturb the object's original geometry
- Pressure increase when covered will depend on the area of the outlet
    - Similarly to the hose example from before



- Experimented with different steps
- Got the best performance with a step of 0.1mm2 in area
- Gives us outlets diameters from 0.6 to 1.5mm
- See outlets with an euro coin for stacle



- See the pressure increases when covering each
- Pressure (kPa)
- Area of outlet
- Notice how linear it is
- Also the separation between the outlets
- These two things allow us to get our high accuracies

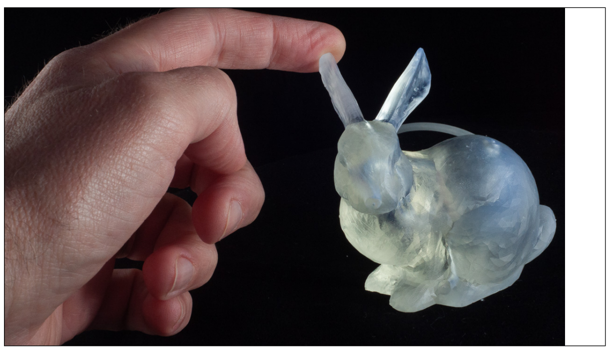
Performance Testing

- Printed 4 objects of different geometries and outlet configurations
- Pre-trained ML models using 1 instance per touch location
- Cycled through all locations 4 times and recorded the predictions

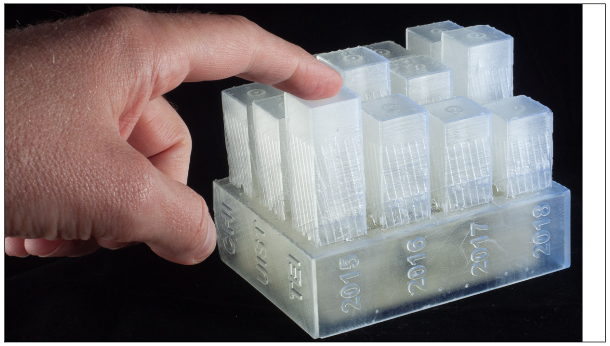
Performance Testing

- Printed four objects.
- Pre-trained a machine learning model.
    - One instance per touch.
- Cycled through all touch locations.
- Repeated four times per object.

- Printed 4 objects of different geometries and outlet configurations
- Pre-trained ML models using 1 instance per touch location
- Cycled through all locations 4 times and recorded the predictions

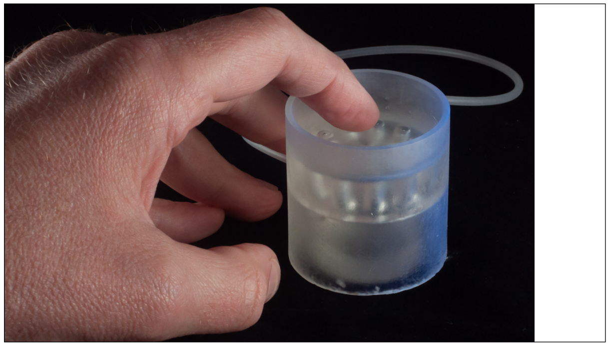


- Stanford bunny (4 outlets)

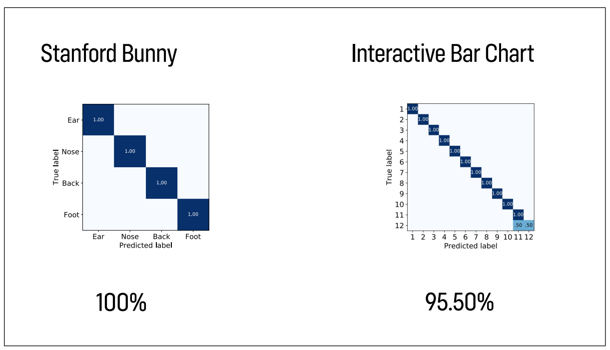


- Interactive bar chart (12 outlets)
-

- Grasp sensing sphere (6 outlets)

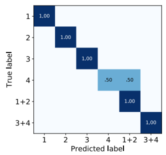


- Color hue selector (12 outlets)
-



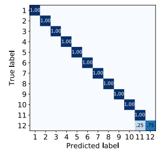
- Individual confusion matrices:
    - Grasp sensing cube 91.6%
    - Color hue selector 97.75%

Grasping Sphere



91.60%

Color Hue Selector

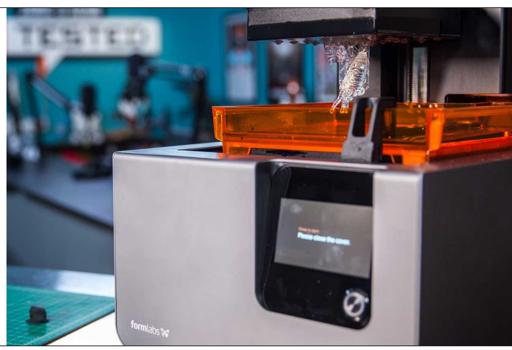


97.75%

- Individual confusion matrices:
  - Grasp sensing cube 91.6%
  - Color hue selector 97.75%

Limitations

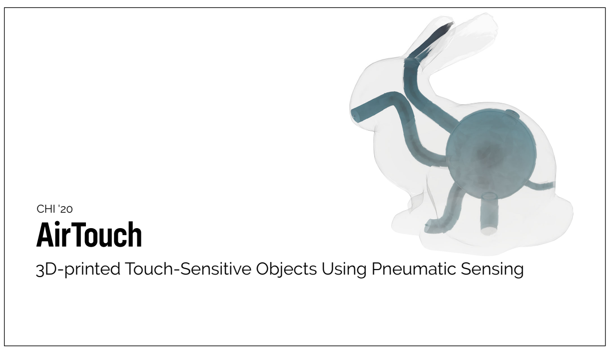
- Used an air compressor for our tests
    - We wanted high control on the pressure and flow for our experiments
- Other air sources might be used
- Tested mini pumps
    - Some degree of success
    - More experimentation is needed



- Our technique requires high-res 3D-printing
    - Like STL
- These technologies are becoming more accessible.

Atmospheric Pressure Changes

- Noticed day to day changes in atmospheric pressure in our experiments
- Performed some tests, and it seems it doesn't affect our performance
- Need more data



AT helps designers to fabricate touch sensitive objects without needing to assemble any circuits or calibrate machine learning models

Sensors, not artifacts

All these projects behave closer to sensors than actual interactive devices
Are tethered to a computation device
Hard to deploy in the wild

How to fabricate stand-alone interactive devices?

Microcontrollers + sensors?

MCU and sensors are not the answer
Require engineering expertise

Fabricate object with computation embedded into it

Interesting alternative is to fabricate objects with computation already embedded into them

3D-printed Logic

Ongoing project explores

Logic gates

Investigating how to embed logical operations into fabricated objects as a first step towards 3D-printed computing

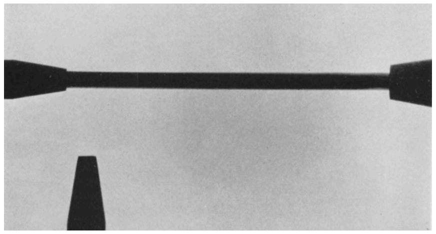
Fluidics

The way I'm doing it is making use of fluidics

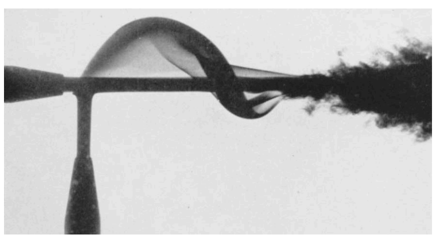
Fluidics

- Technique of using small interacting flows and fluid jets for functions usually performed by electronic devices.
- Require no moving parts or electricity.

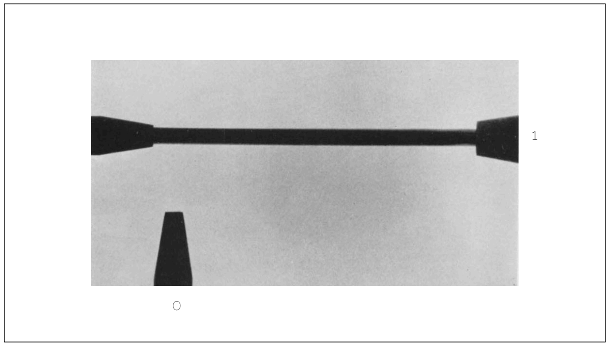
Technique pioneered in the 60s that makes use of fluid jets to represent functions usually performed by electronic devices



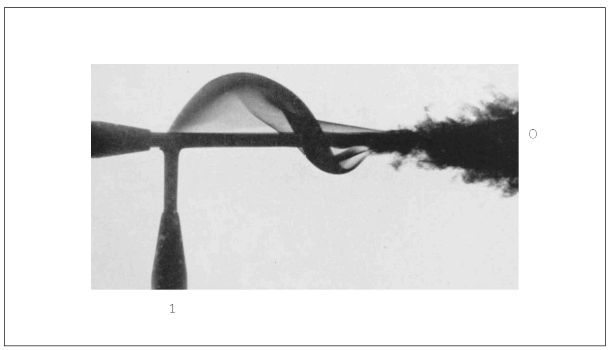
Black line across the picture is a jet of air



When we shoot another jet of air towards it, it gets dispersed



So what we have hear is a NOT gate



So what we have hear is a NOT gate

Previous work is unaccessible

Most of this work was carried out in the 60s and there's very little guidance on how to fabricate these structures

---

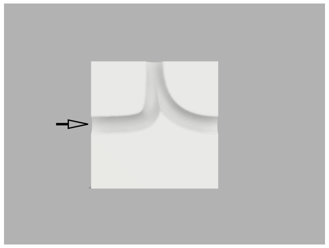
3D-printed Logic Toolkit

Proposing a 3D-printed Logic toolkit to allow novices to experiment with this technique
Once the design is complete and tested, it can be transferred to the interior of a 3D-model

Components

- Toolkit is made of a set of components that the designer can play with
- About an inch wide, half an inch tall

Input

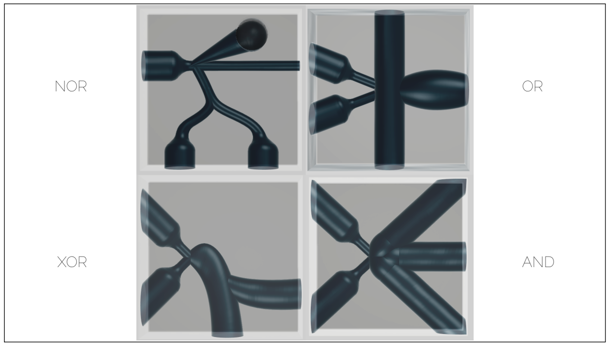


Touch

- How it works
- Also exploring blowing, and squeezing as input

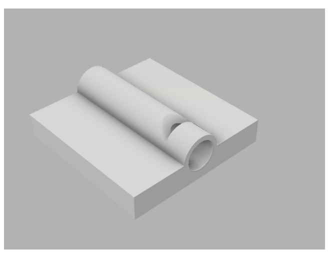
Logic

Then we have our logic components
Here's a video to show how they work


NOR
OR
XOR
AND

Output

- Output components to help physicalize the computation



Whistle

- Whistle
- Balloons
- Sensors

Translate components to 3D-model using design tool

Working on a design tool to allow

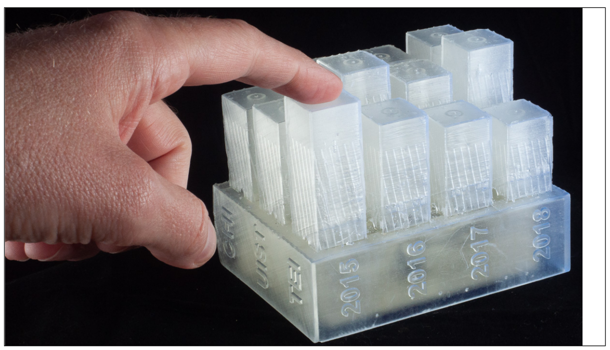
Research Agenda

Non-experts can create interactive artifacts in the physical world

Sensing techniques

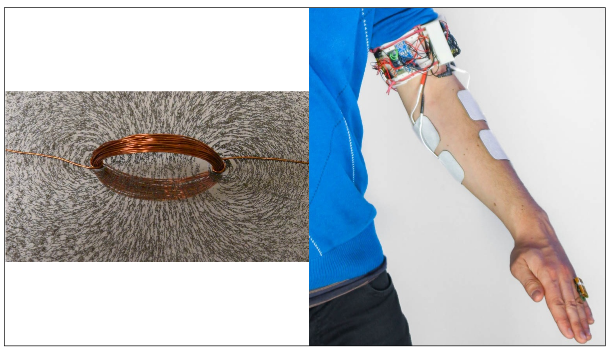
- Interactive artifacts not only sense, but process and provide feedback

Haptic Feedback

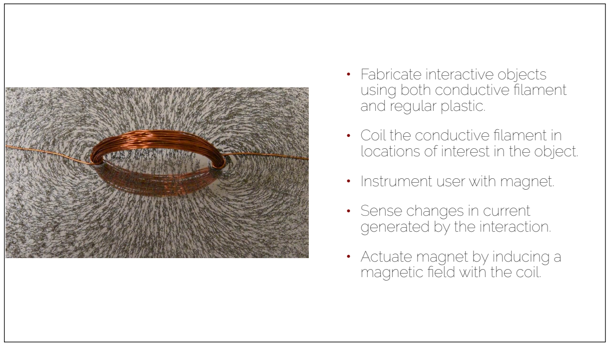


- When you interact with a fabricated object, there's no feedback that the interaction has taken place

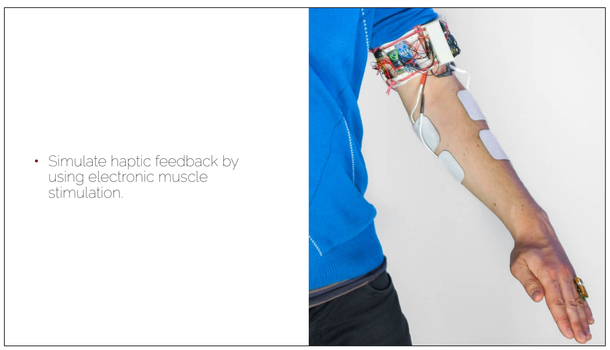
- Feedback is not like interacting with an actual device



Interested in exploring two main ideas



First one electromagnetism to enable touch interactivity on fabricated objects, and also provide haptic feedback



Enabling non-experts to author tangible interactions

Carlos Tejada
University of Copenhagen
www.carlosetejada.com
@ctejada10