

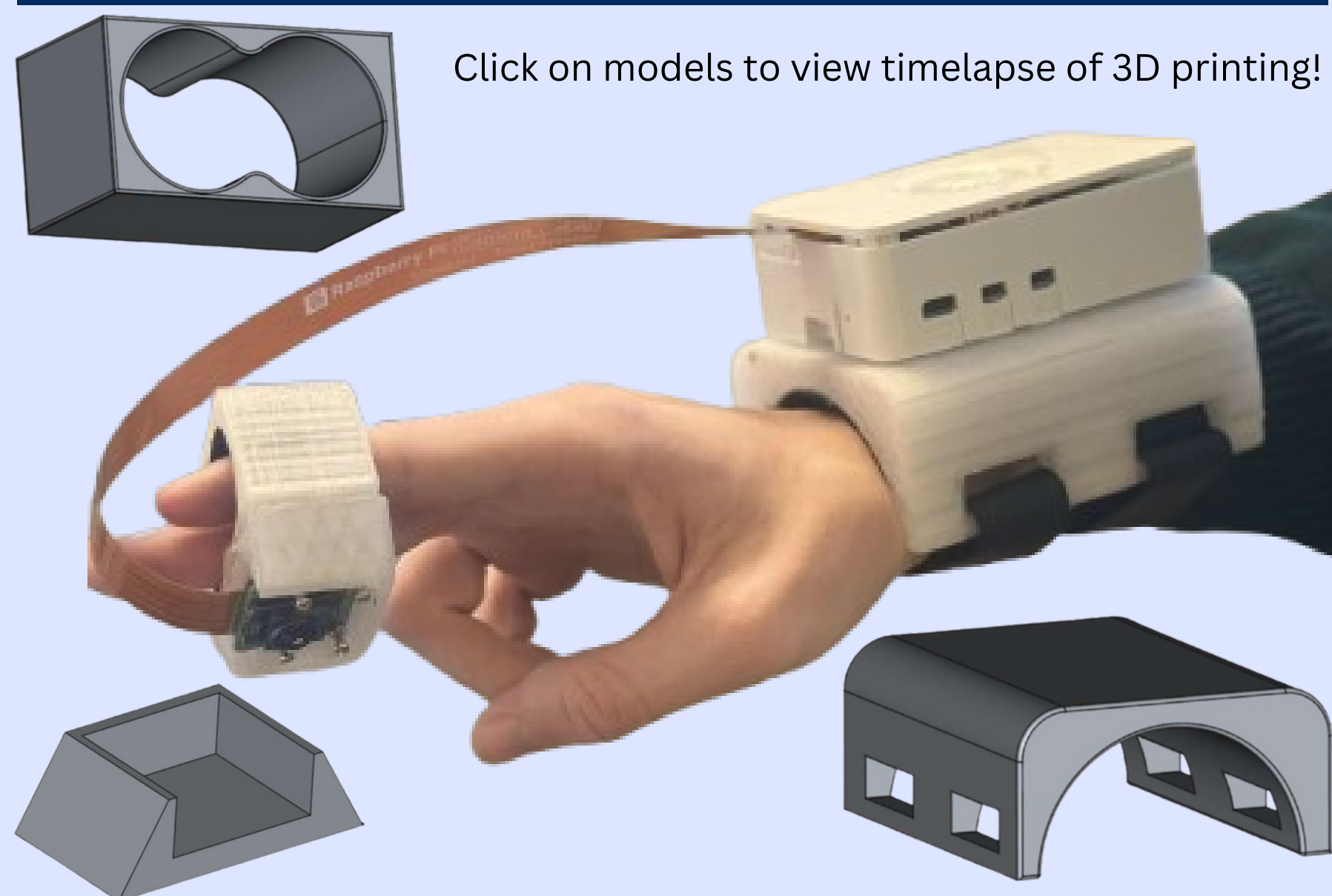
Engineering Design Process

- **Ask:**
 - State: ~ 32,800 legally blind people
 - Nationally: ~1.1 million legally blind people
 - Everyday, these people struggle to complete basic tasks that require reading, such as learning from textbooks or reading menus.
 - How could we develop a solution for those who are visually impaired that limits costs?
- **Research:**
 - Audio books can be expensive and impractical.
 - Braille books cost on average \$1/printed page.
 - Cellular apps can be hard to navigate, especially for the visually impaired, oftentimes requiring users to scan an entire page at a time.
 - MIT students previously created a prototype for a finger-mounted reader in 2016. No model has even been marketed.
- **Imagine/Plan:**
 - We want to help the visually-impaired community as well as the general public.
 - How do we incorporate AI and machine learning to the reader?
 - Unlike MIT, our camera replicates Braille reading, increases comfort, and removes strain.
 - Planned on using the Raspberry AI camera after initial model with reused Chromebook parts proved ineffective.
- **Create:**
 - Utilizing FreeCAD, we designed a mount.
 - Next we coded the Pi AI Camera to recognize text using Google Tesseract. It was programmed to break down sentences into words, and broadcast the text as audio.
- **Test:**
 - Initially, the finger mount was too small.
 - Troubleshooted code numerous times.
 - Pi AI camera had a minimum focus of 20 cm, which was too far to discriminate small text.
- **Improve:**
 - Increased size of finger mount and increased comfort with added felt.
 - Decided to test the Pi Camera 3 Wide, which has a minimum focus of 5 cm.
 - To continue using the Pi 5, we will also implement the AI HAT or Compute Module 5 to utilize AI.

RAVI

What is RAVI?

“RAVI”, which stands for “Reading Assistant to the Visually Impaired”, is a finger-mounted device that uses artificial intelligence to assist blind people in reading. By attaching the camera to two fingers, RAVI interprets print, converts the print to audio, and transmits this audio to an earpiece, allowing the user to hear what is written.



Click on models to view timelapse of 3D printing!

How It Was Made

- Finger Mount/Camera Indent/Board Mount
 - FreeCAD and 3D Printing
- Text Interpretation
 - Pi AI Camera, Pi 3 Wide Camera, Raspberry Pi 5, Compute Module 5/AI HAT, Google Tesseract
- Audio Transmission
 - Connect with personal earbuds
 - Potentially Bluetooth in the future

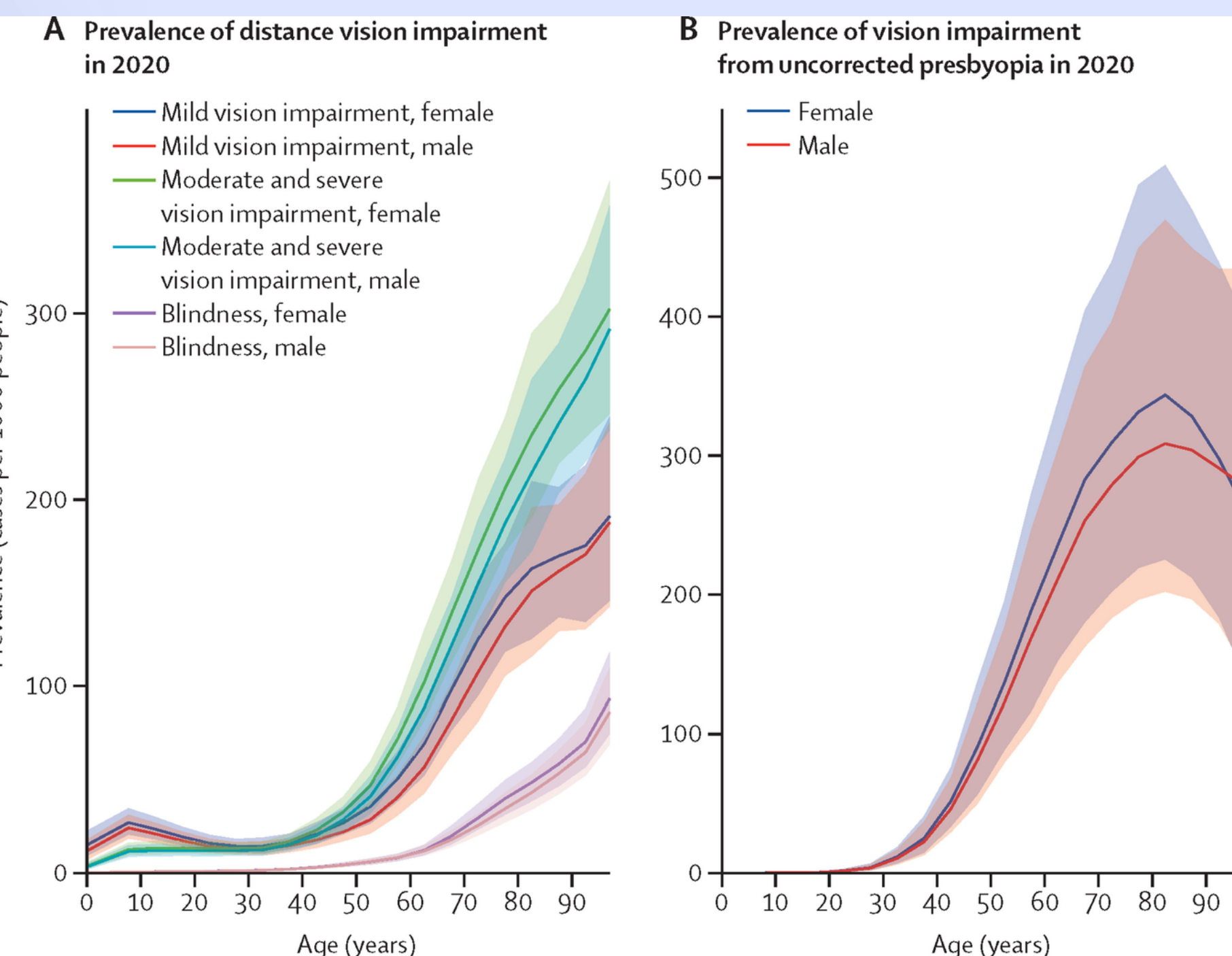
Meet the Team

References

Acknowledgements

Budget

Professional Guidance



Universal Design Principles

- **Equitable Use:** RAVI can potentially assist people who are practicing a new language, traveling abroad, or learning to read for the first time.
- **Flexibility in Use:** RAVI can be worn in multiple ways depending on what needs to be read. For instance, when reading flat texts, such as a menu, the camera can go on the bottom of your fingertips. For vertical texts, such as signs on a wall or a computer screen, the camera can be worn on your knuckles with the same result.
- **Size and Space for Approach and Use:** The addition of felt inside the mount not only adds comfort for the wearer but also allows for broader use. By lining the inside of a larger mount with felt, various hand sizes can fit inside the device. Furthermore, the 3D print can be easily re-scaled, permitting us to create a model that fits any range of hand sizes.
- **Simple and Intuitive Use:** RAVI's reading method follows natural or experienced Braille reading patterns, something the visually impaired are very familiar with.
- **Perceptible Information/Tolerance for Error:** RAVI reads text in real-time. In cases when no text can be identified, the user can always swipe back over the text. If someone is below or above the line of text, RAVI sounds a low or high pitched tune, respectively.
- **Low Physical Effort:** Reading with RAVI requires a simple swipe of your fingers! For future developments, we could use less dense filament for the 3D prints and lightweight lithium batteries.