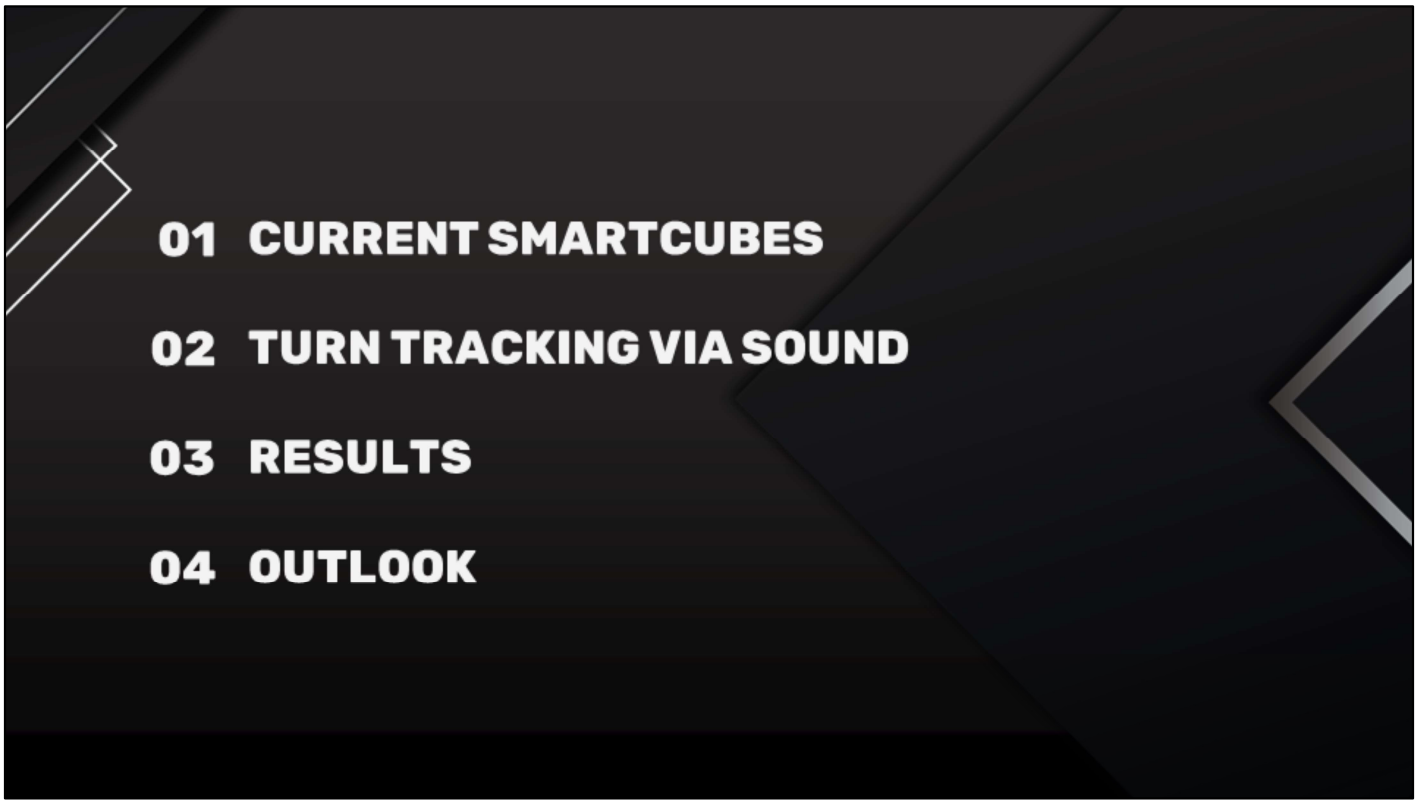
Thank you for that introduction! And thanks to all of you for attending!

We have a lot of people here in-person, and many more on Zoom including one of the software developers for the World Cube Association which governs Rubik's Cube competitions.

As you can see, the title of my presentation is D.I.Y. Smartcube, and you may be wondering what is a smartcube?

In short, a smartcube is a special version of a Rubik's Cube that can record the turns you make on the cube and transmit them wirelessly to another electronic device.

They have been hugely impactful in the Rubik's Cube speedsolving community because they enable advanced statistics that help speedcubers more quickly identify areas for improvement.

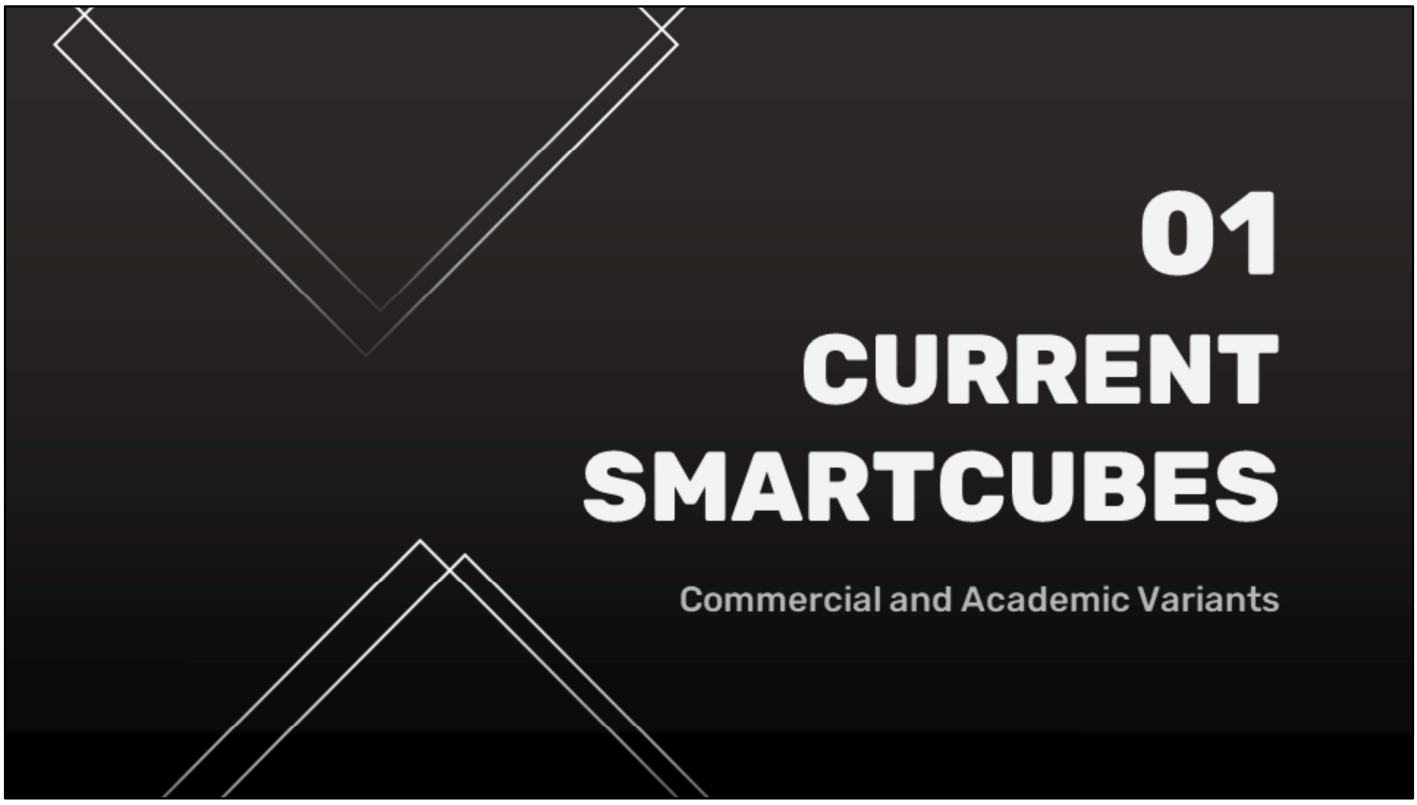


For this presentation, we will start by looking at how other people have built smartcubes.

From there, I'll describe the alternative, sound-based smartcube design I created in this project.

After that, we'll look at some test results that evaluated the design's strengths and pitfalls.

And finally, we'll end with a discussion of the next steps that someone can follow to build off of my work.



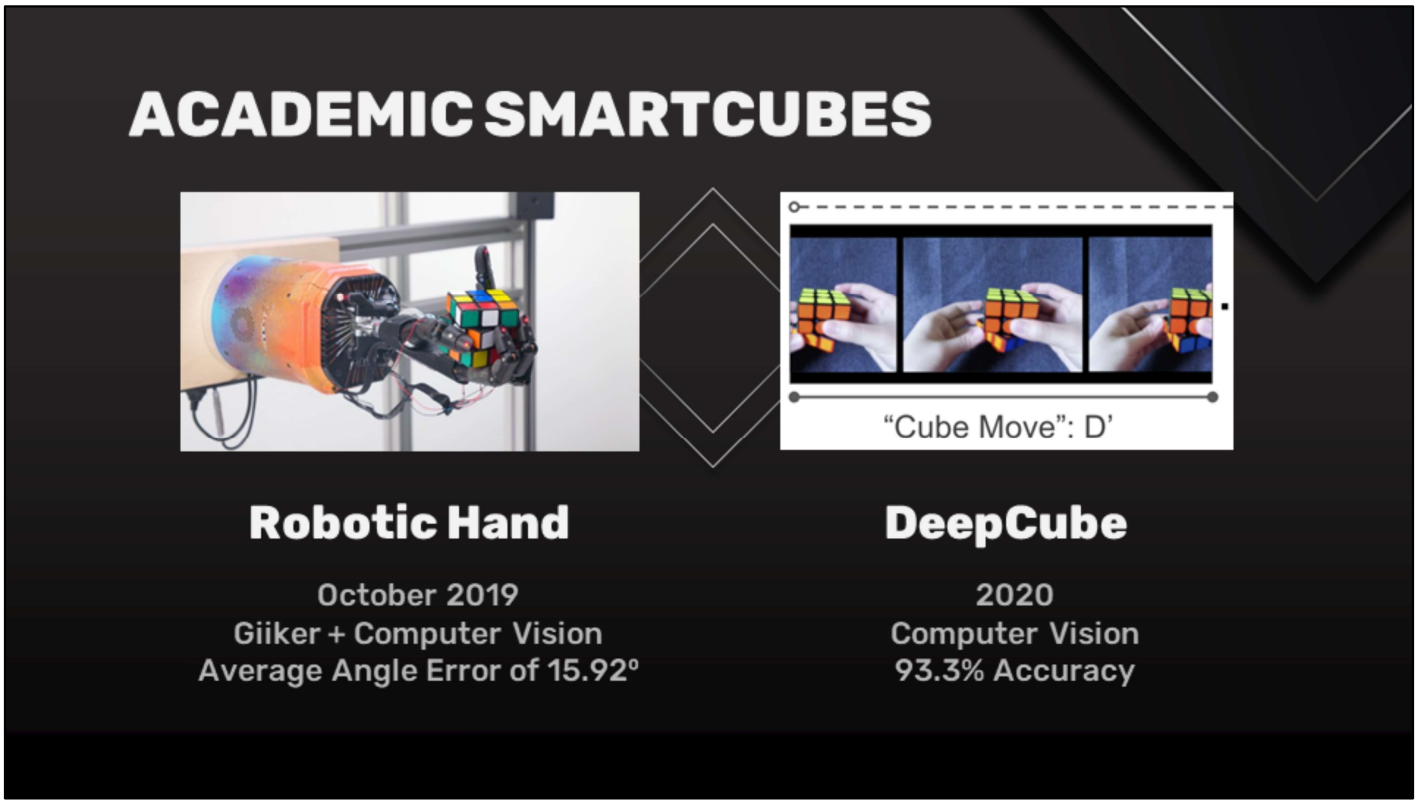
Lots of people and companies have researched various ways to track the face turns of a Rubik’s Cube. In my thesis document, I describe all of these in great detail, but for this presentation I’ll focus on the most successful commercial and academic designs.

We'll start off on the commercial side.

What you see here are the internal circuit boards that power the first few smartcubes to enter the market.

Each of these boards is slightly larger than a quarter and located in the core of the cube, with all the other pieces custom built around it.

While there is some variation from cube to cube, they all work along the same lines: they use some sort of electronic sensor to detect the face turns and leverage a Bluetooth transmitter to communicate those turns with a nearby phone or laptop.



In the academic realm, there's been a lot of focus on leveraging Computer Vision to detect face turns simply by "watching" them happen.

This first image shows OpenAI's the robot hand that taught itself to solve the cube. To do that, they used a combination of Computer Vision and an enhanced, Giiker smartcube to measure the angle of rotation within 16 degrees of accuracy.

On the right is a sample of the DeepCube dataset created by Stanford graduate students. They recorded themselves solving the cube, labelled each face turn in the video, and trained a their own Computer Vision algorithm to predict those labels with over 90% accuracy.

As I reviewed all these smartcube designs, I also researched other ways to detect and communicate rotation information.

One of these methods was sound, and it stood out to me because the same smartphones and laptops that connect to smartcubes over Bluetooth also come equipped with microphones.

Furthermore, I couldn't find anyone else who had tried to build a smartcube this way, so I figured it would be a fun challenge.

In fact, as I started building it, I found **five** key challenges that needed to be overcome in order to produce a functional design.

The first was **feasibility**, which requires knowing which audible tones can be detected by the microphones built into laptops and smartphones.

Second, was understanding how far apart two tones need to be so that the microphone could **accurately** distinguish them.

Third, I wanted the design to be **compatible** with existing speedcubes, that is to enable converting a normal speedcube into a smartcube without requiring any permanent modifications.

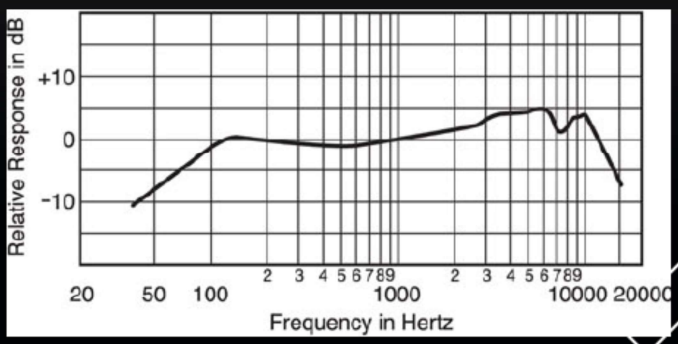
Fourth, I wanted my design to offer the same level of **granular** statistics offered by existing commercial smartcubes.

And Fifth, I wanted my design to **comply with competition regulations** so that speedcubers could use the same cube in their personal practice and in competition.

So let’s take these one by one.

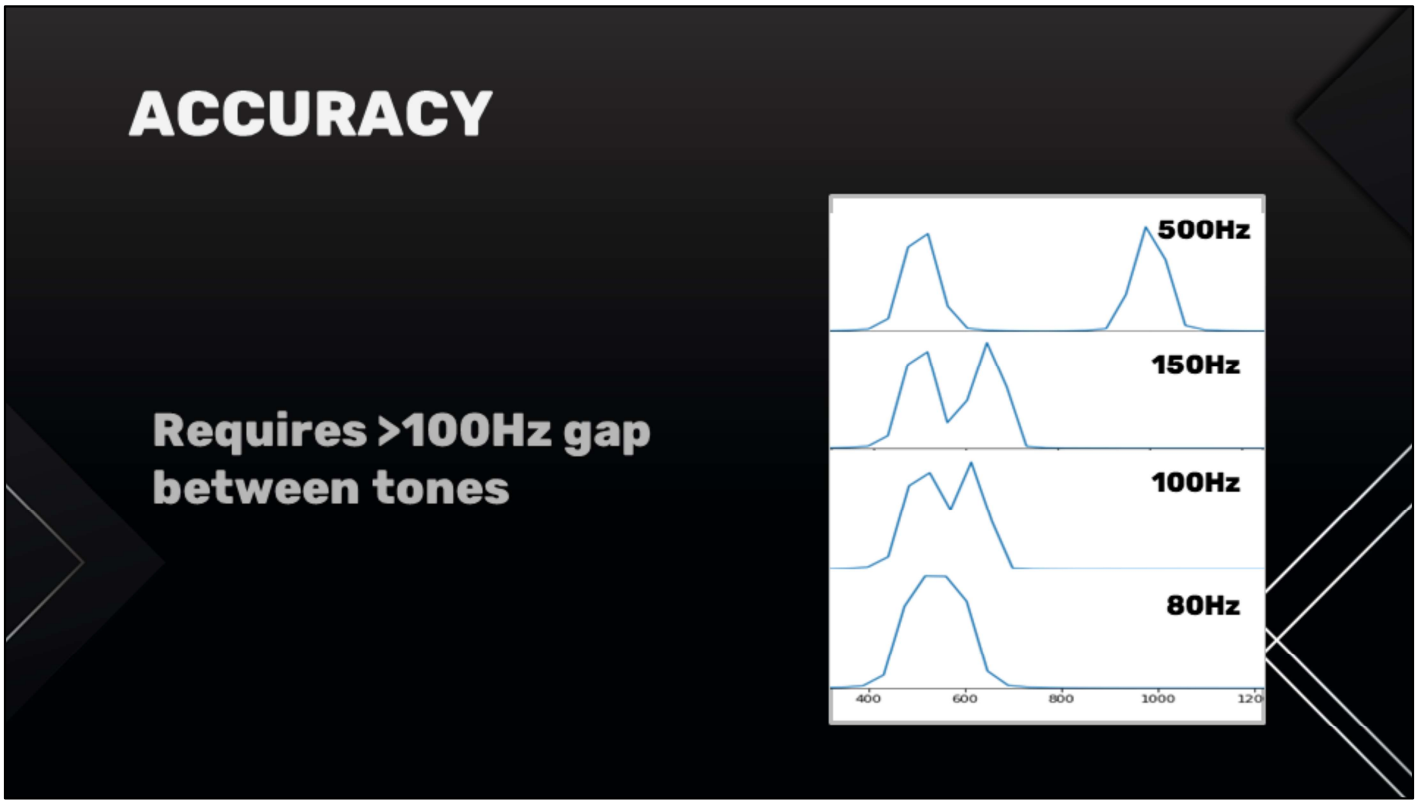
FEASIBILITY

Requires tones between 20Hz and 20,000Hz



Typical frequency response range of a consumer smartphone (Image credit: Hristo Bojinov et al.)

For **feasibility**, it turns out that the microphones in your laptop and smartphone are effective at detecting tones as low as 20 Hz and as high as 20kHz. Interestingly, that is the same range of tones that the human ear can detect.



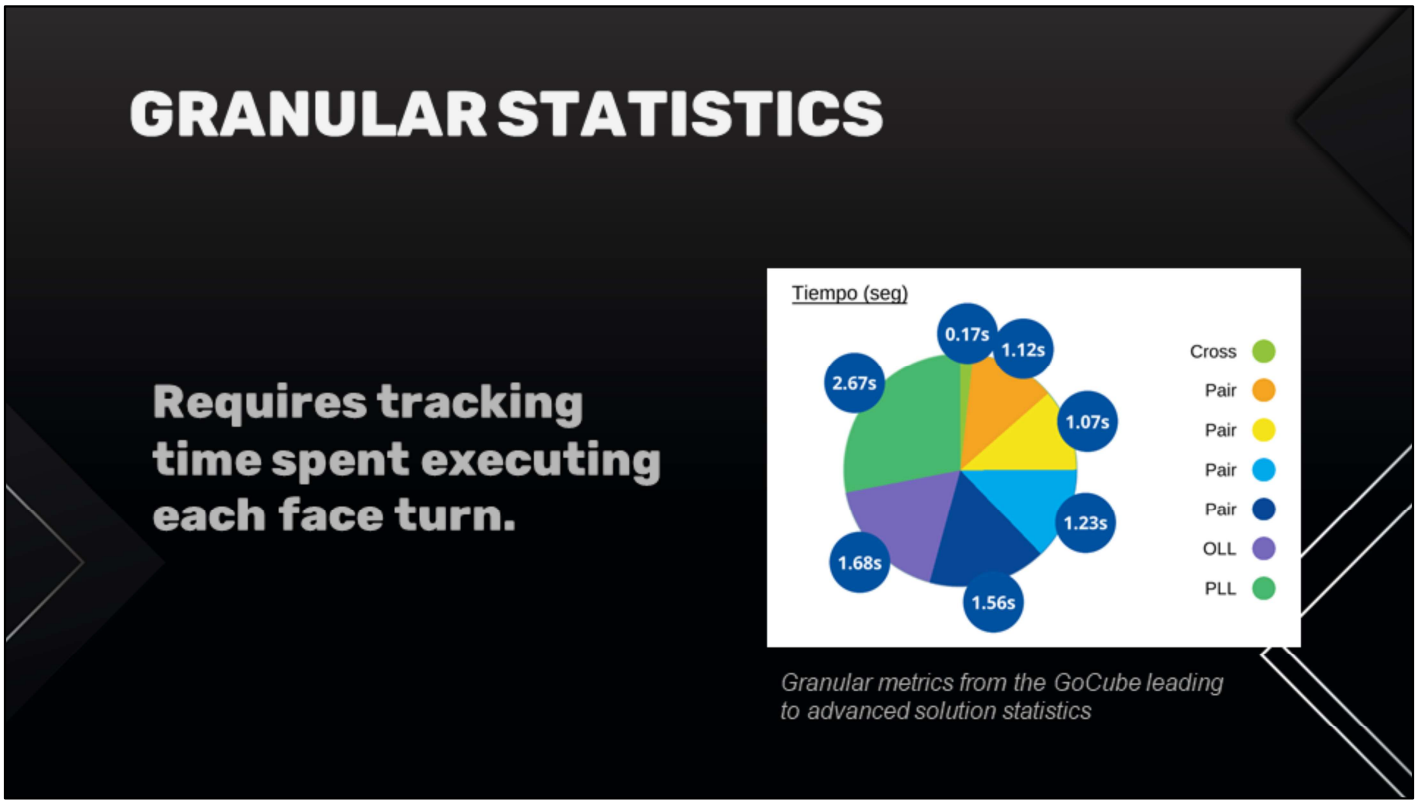
With regards to **accuracy**, I ran some tests using the microphone of my Google Pixel phone and found that it could accurately distinguish tones that were at least 100 hertz apart.

To make a design that is **compatible** with an existing speedcube, we have to know how much space is available for embedding a turn tracking system.

As an example, I took apart the Gans 356 speedcube I used in my first competition which you see in the picture on the right.

The biggest open space in the cube is located inside of the centercap where you see the screw and spring sticking up. This space also works well for a turn tracking system because the black plastic of the center cap rotates around the fixed screw, giving a point of reference to measure a rotation.

However, that space is still really small. In total, there is less than 16 square millimeters of open space, which is comparable to the surface area of a dime.



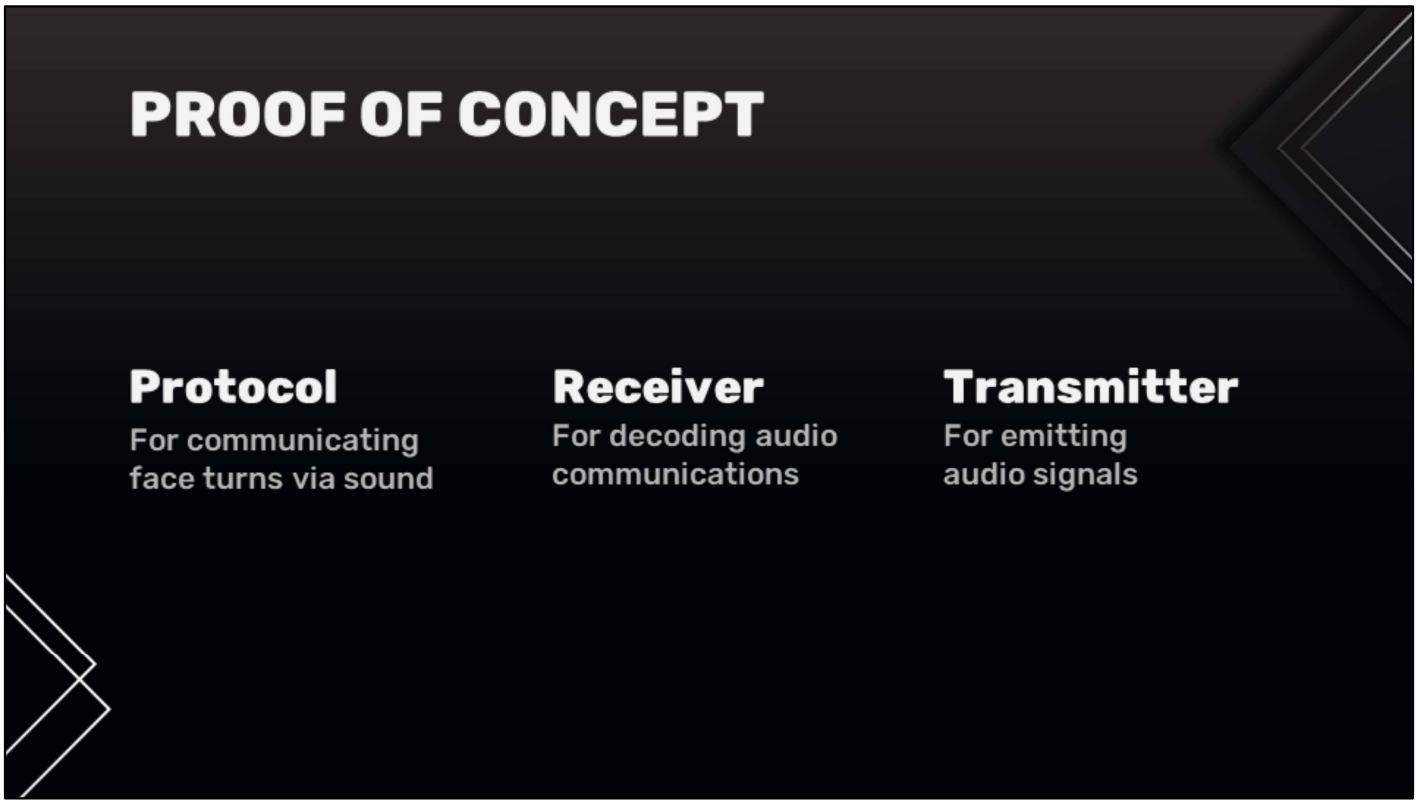
Granular metrics from the GoCube leading to advanced solution statistics

For **granular** statistics, I pulled up the applications provided by various smartcube manufacturers to figure out what statistics they offered. After reviewing them, I determined that they could all be calculated from just two pieces of information: which face turn happened and when?

And finally, I read through all the **competition** regulations to find the ones relevant to smartcubes.

The key regulation is cited here which specifies that cubes used in competition cannot include embedded electronics (which is why commercial smartcubes are not competition legal).

As a result, any electronics I added to the cube to track its turns would have to be easily removable.



So, with those challenges outlined and constraints defined, it’s time to explore the prototype I designed to overcome them.

We’ll start by creating a protocol for the data to send from the smartcube to the microphone to communicate face turns.

With the protocol in place, we’ll discuss the creation of a prototype receiver to decode the audio, and the transmitter that will create it.

PROTOCOL

Assign a unique tone to each face position

TABLE 4.1: Sample frequencies for encoding a Rubik's Cube's state

Alignment	White	Yellow	Red	Orange	Green	Blue
Full	800 Hz	1300 Hz	1800 Hz	2300 Hz	2800 Hz	3300 Hz
90°	900 Hz	1400 Hz	1900 Hz	2400 Hz	2900 Hz	3400 Hz
180°	1000 Hz	1500 Hz	2000 Hz	2500 Hz	3000 Hz	3500 Hz
270°	1100 Hz	1600 Hz	2100 Hz	2600 Hz	3100 Hz	3600 Hz

I explored several options for a protocol design in my thesis document, but the one I settled on is fairly straightforward.

Notice that the cube has six faces, *gesture on a cube 1,2,3,4,5,6*

PROTOCOL

Assign a unique tone to each face position

TABLE 4.1: Sample frequencies for encoding a Rubik's Cube's state

Alignment	White	Yellow	Red	Orange	Green	Blue
Full	800 Hz	1300 Hz	1800 Hz	2300 Hz	2800 Hz	3300 Hz
90°	900 Hz	1400 Hz	1900 Hz	2400 Hz	2900 Hz	3400 Hz
180°	1000 Hz	1500 Hz	2000 Hz	2500 Hz	3000 Hz	3500 Hz
270°	1100 Hz	1600 Hz	2100 Hz	2600 Hz	3100 Hz	3600 Hz

Each of these faces can be in one of four possible alignments. *gesture with cube*

PROTOCOL

Assign a unique tone to each face position

TABLE 4.1: Sample frequencies for encoding a Rubik's Cube's state

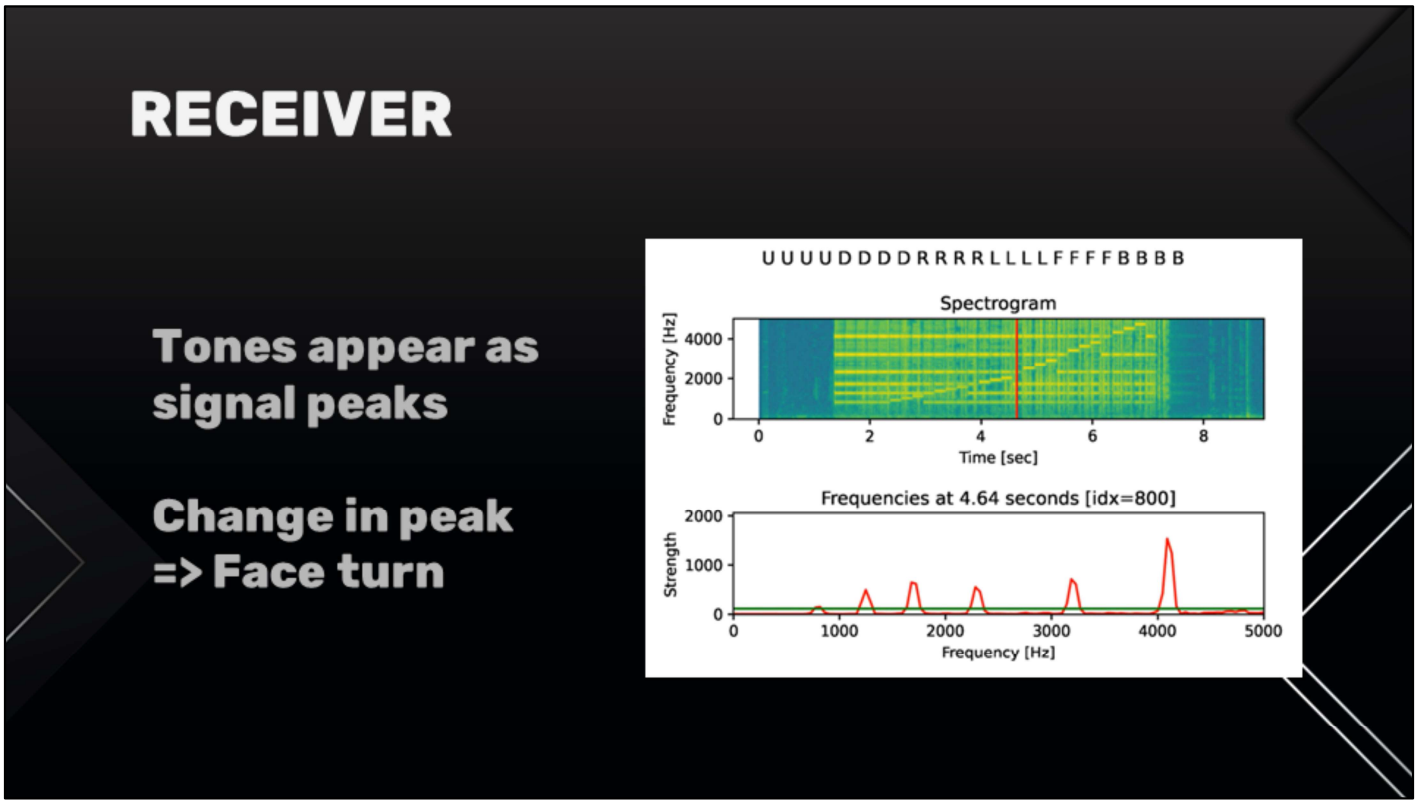
Alignment	White	Yellow	Red	Orange	Green	Blue
Full	800 Hz	1300 Hz	1800 Hz	2300 Hz	2800 Hz	3300 Hz
90°	900 Hz	1400 Hz	1900 Hz	2400 Hz	2900 Hz	3400 Hz
180°	1000 Hz	1500 Hz	2000 Hz	2500 Hz	3000 Hz	3500 Hz
270°	1100 Hz	1600 Hz	2100 Hz	2600 Hz	3100 Hz	3600 Hz

So, these six faces with four alignments each gives us total of 24 unique alignments, each of which can be identified with a unique tone.

Each centerpiece will be responsible for generating the single tone corresponding to its current alignment, so the cube will be broadcasting six tones at all times.

Whenever a face is rotated, the transmitter in its centercap will broadcast the new tone corresponding to the new alignment.

By recording these tone changes over time, we can build a software algorithm that will convert the sequence of tone changes back into the original moves that were applied to the cube.



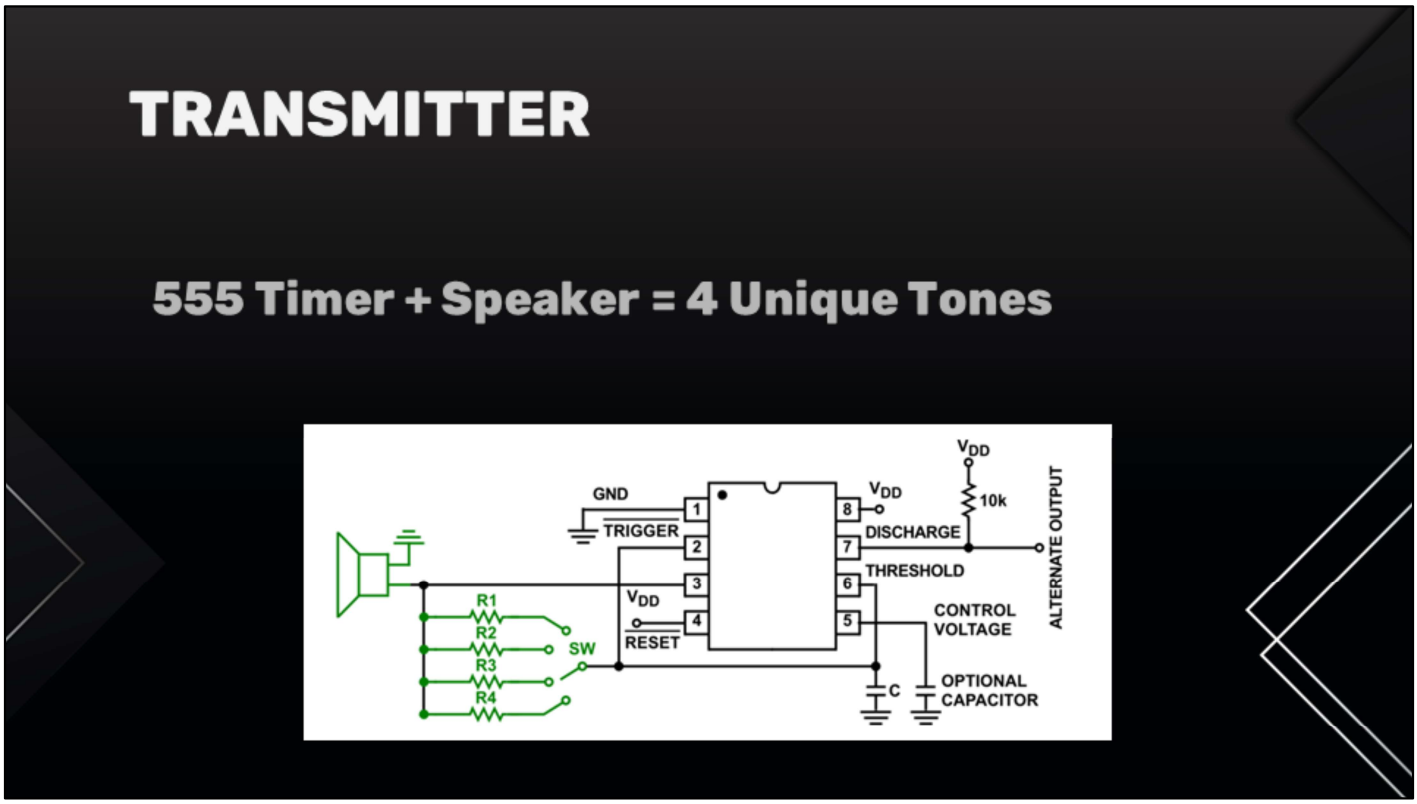
To understand how that algorithm works, we need to understand what those tone changes look like to the computer.

The blue and yellow image you see on the top right, is called a spectrogram. It helps us to visualize an audio recording by showing loud tones over time in bright yellow and quite tones in dark blue.

The lower graph represents a slice of the spectrogram and the six peaks you see in red represent the six tones broadcasting the alignment of each of the cube's faces at that moment in time.

The fuzziness in the spectrogram represents the noise created by the cube itself as it turns.

The software algorithm simply scans over that spectrogram and records the moments when those peak frequencies move and converts those changes into the sequence of face turns executed on the cube.



Since the transmitter is only responsible for switching between four tones, it can also use a relatively simple design.

The circuit design you see below has three main sections.

First, the large black component is a 555 timer, which produces a consistent rhythm of high and low voltages.

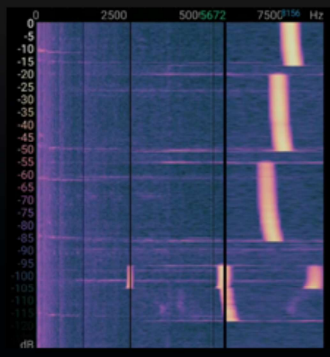
Second, the small group of green lines at the bottom left is a set of resistors which control the frequency of the voltage swings output by the 555 timer. Each of the four resistors creates a different frequency, and the switch ensures that only one is active at a time.

Finally, the green speaker on the far left converts the 555 timer's voltage swings into audible sound.

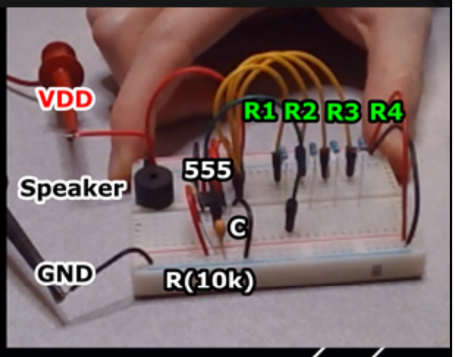
So that’s the design! Now let’s explore how well it works at resolving the five challenges shared earlier.

FEASIBILITY

Successful audio detection by a consumer phone



Audio recording of the sound generated by the prototype transmitter

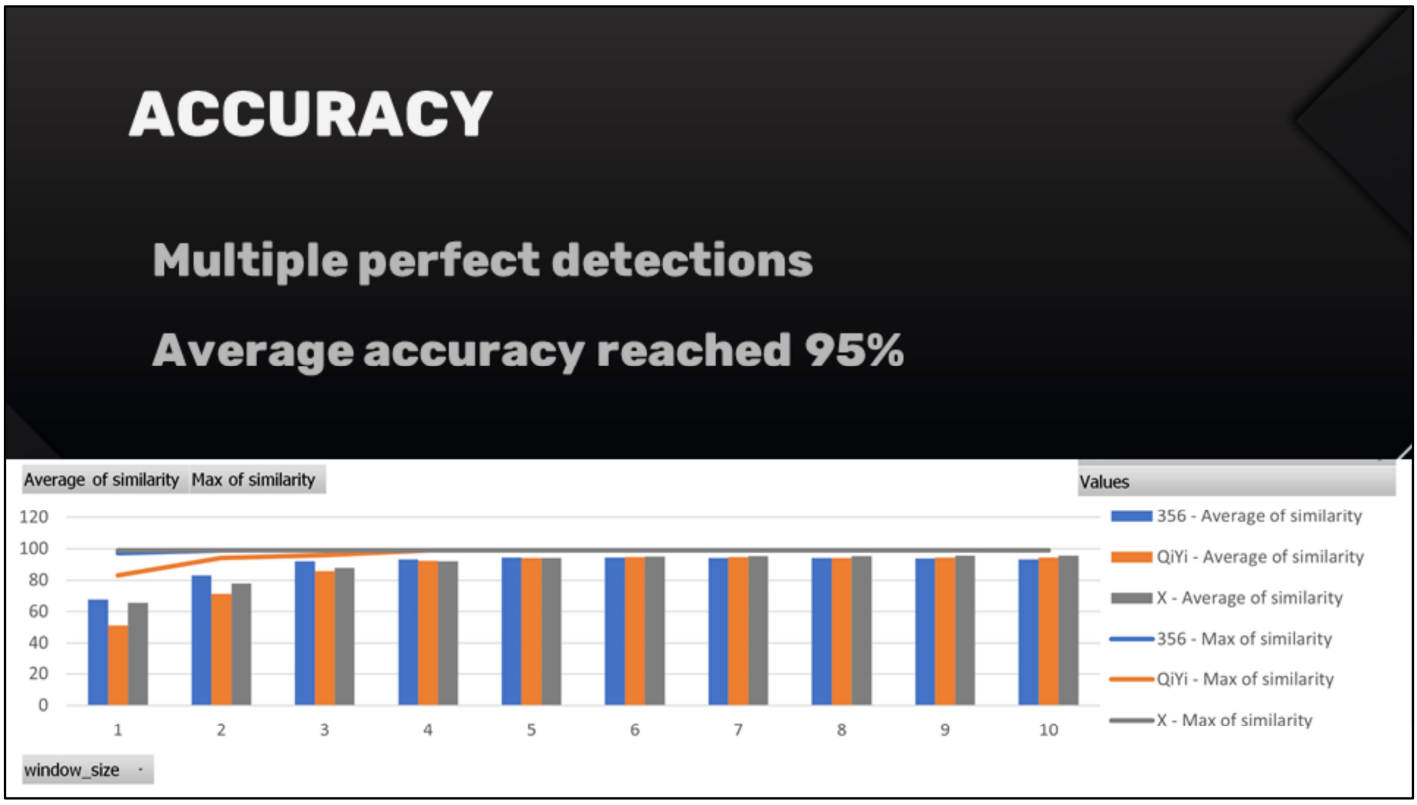


Prototype board for the transmitter

To test the **feasibility**, of the design I built a larger-sized version of the circuit to make sure that it played four distinct tones within the range detectable by a consumer smartphone, specifically my Google Pixel.

As you can see in the spectrogram on the left, as we go up over time, the bright pink line jumps through four unique frequencies.

Those jumps occurred at the exact moments I moved a wire on the breadboard to a different resistor to simulate a manual switch inside of the cube as it turns.

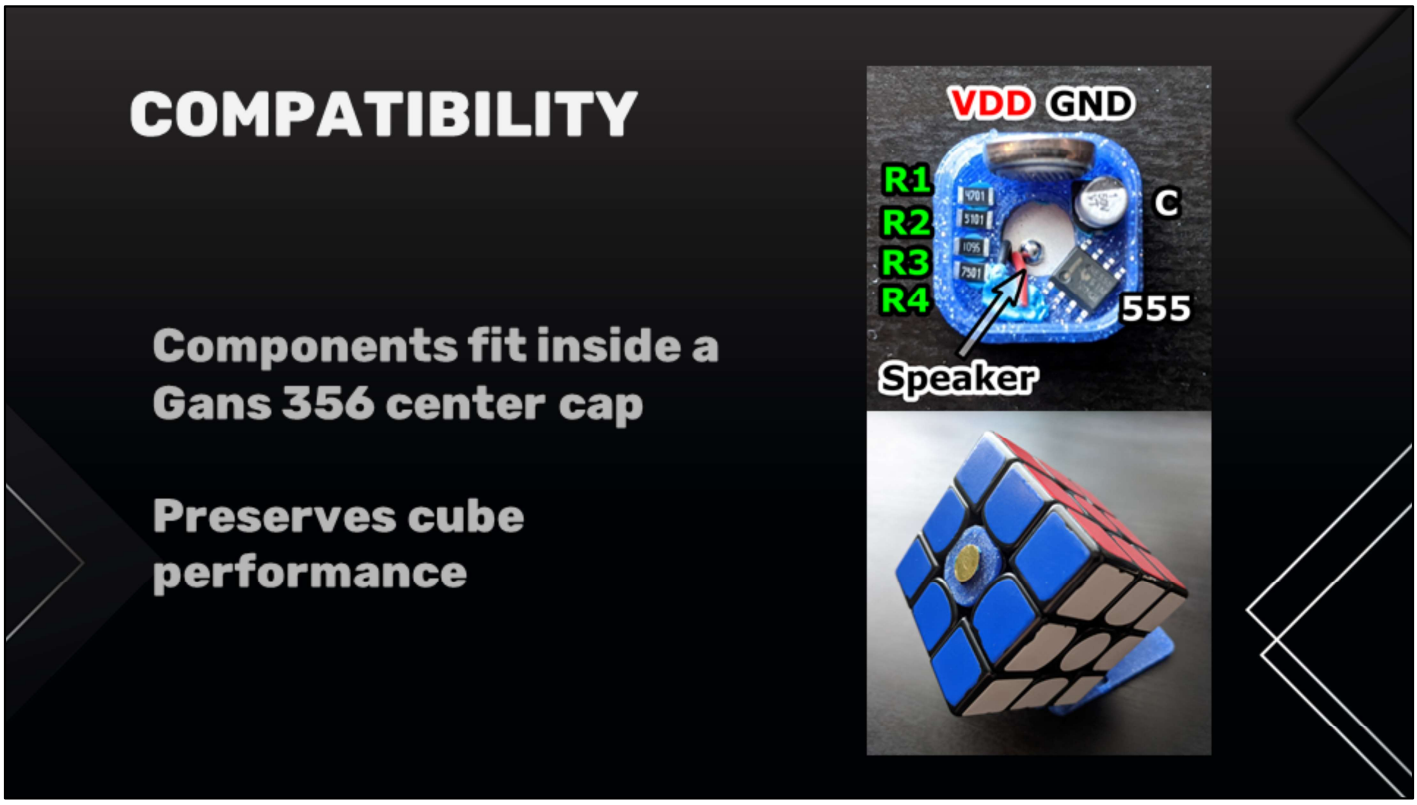


To measure the **accuracy** of the receiver, I generated six audio samples, two for each of three different types of cubes and ran them through the receiver 720 times using a variety of receiver configurations.

I then compared each move sequence reported by the receiver with the original one I had synthetically transmitted to measure the receiver's accuracy.

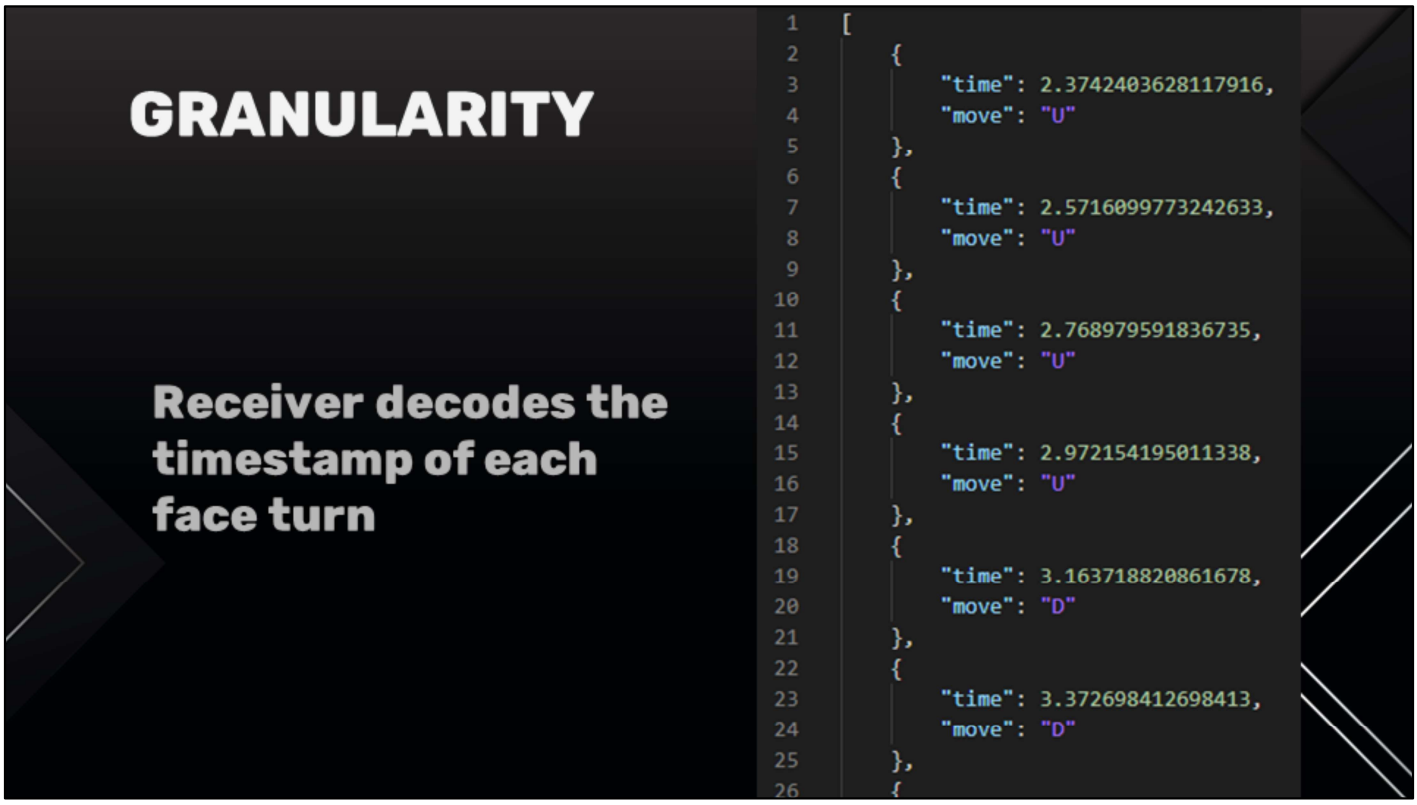
The line across the top shows the best accuracy achieved as the receiver required additional verification steps before reporting a face turn.
-> As you can see, once the receiver required successful validation of a face turn over four consecutive time steps, the receiver achieved a perfect accuracy for all cubes at least once.

The bars across the bottom show the average decoding accuracy across all configuration settings.
-> Again, after four time steps of validation, the average accuracy reached 95%.



To test the design's **compatibility** with existing speedcubes, I 3D printed a custom centercap for my Gans 356 and made sure it could hold all the components for the transmitter.

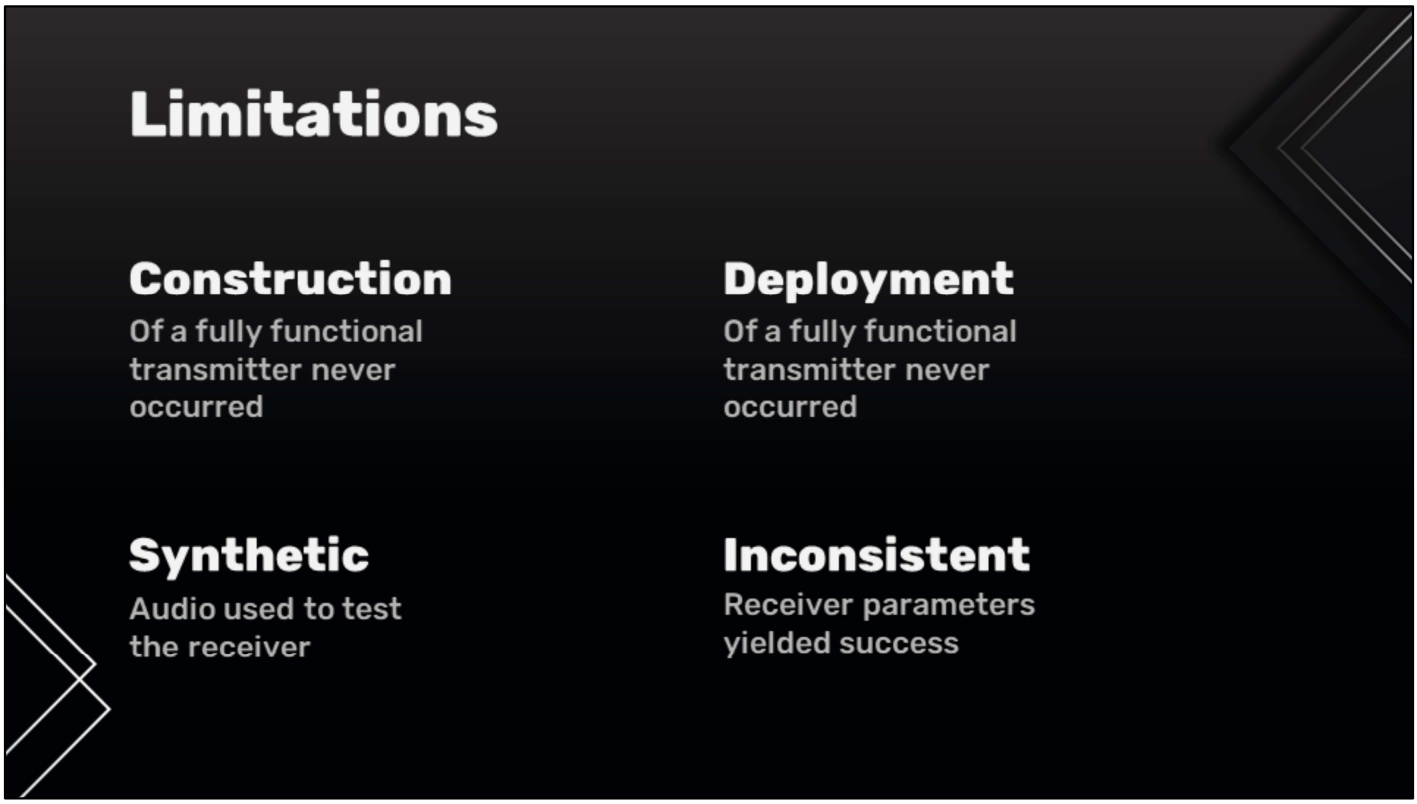
Then I replaced original centercap with my custom one to make sure it fit snugly and retained the original performance of the cube.



Moving on to **granularity**, whenever the software algorithm in the receiver detects a face turn, it also records the timestamp at which the turn happened.

As discussed earlier, those two pieces of information enable the calculation of useful metrics like turn speed, and the time spent in different phases of the solution.

Finally, any conflicts with **competition** regulations are resolved by simply swapping out the center cap containing all the transmitter electronics with the original one that came with the cube.



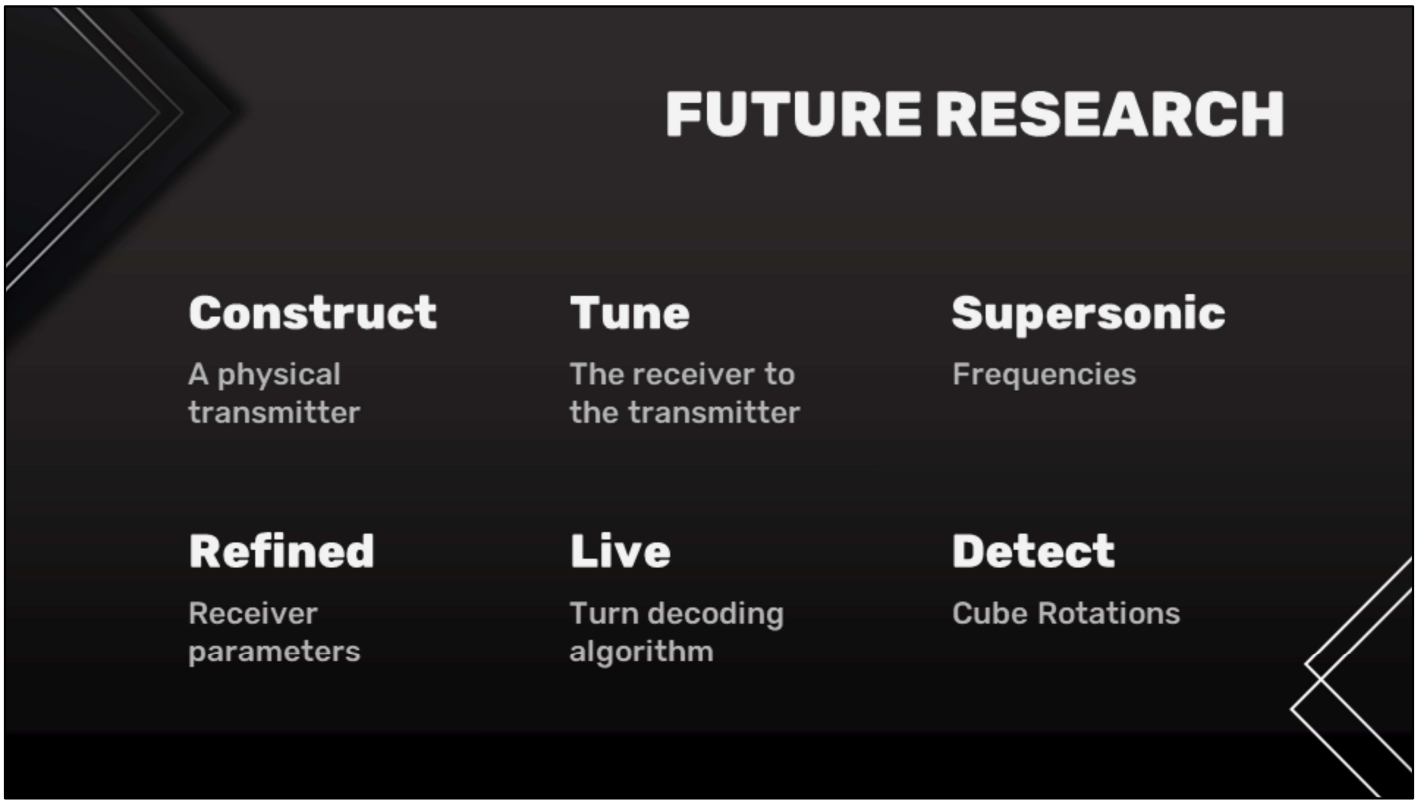
I also want to take some time to be honest about the potential pitfalls of this design.

First, while I did build a functional transmitter on a breadboard and demonstrated that the transmitter's components could fit into a Gans 356 center cap, I didn't get far enough to professionally design and **construct** that tiny circuit and **deploy** it within my Gans 356 speedcube.

As a result, though I went to great lengths to make sure to test the receiver with as realistic of synthetic audio as possible, it was still synthetic and might have fundamental differences from the audio that would be generated by an actual transmitter in a cube.

Lastly, even though the receiver did achieve perfect turn detection accuracy for each test case, the specific configuration settings that yielded that perfect result differed from test case to test case. As a result, even though some parameters, namely the window size, strongly correlated with higher accuracy, a perfect detection accuracy cannot be guaranteed on new test cases without additional configuration.

But let’s not let those shortcomings block continued research. For those intrigued by this proof-of-concept, let me outline six directions in which improvements can be made by future researchers.



First and foremost, the transmitter needs to be constructed and deployed within a standard smartcube.

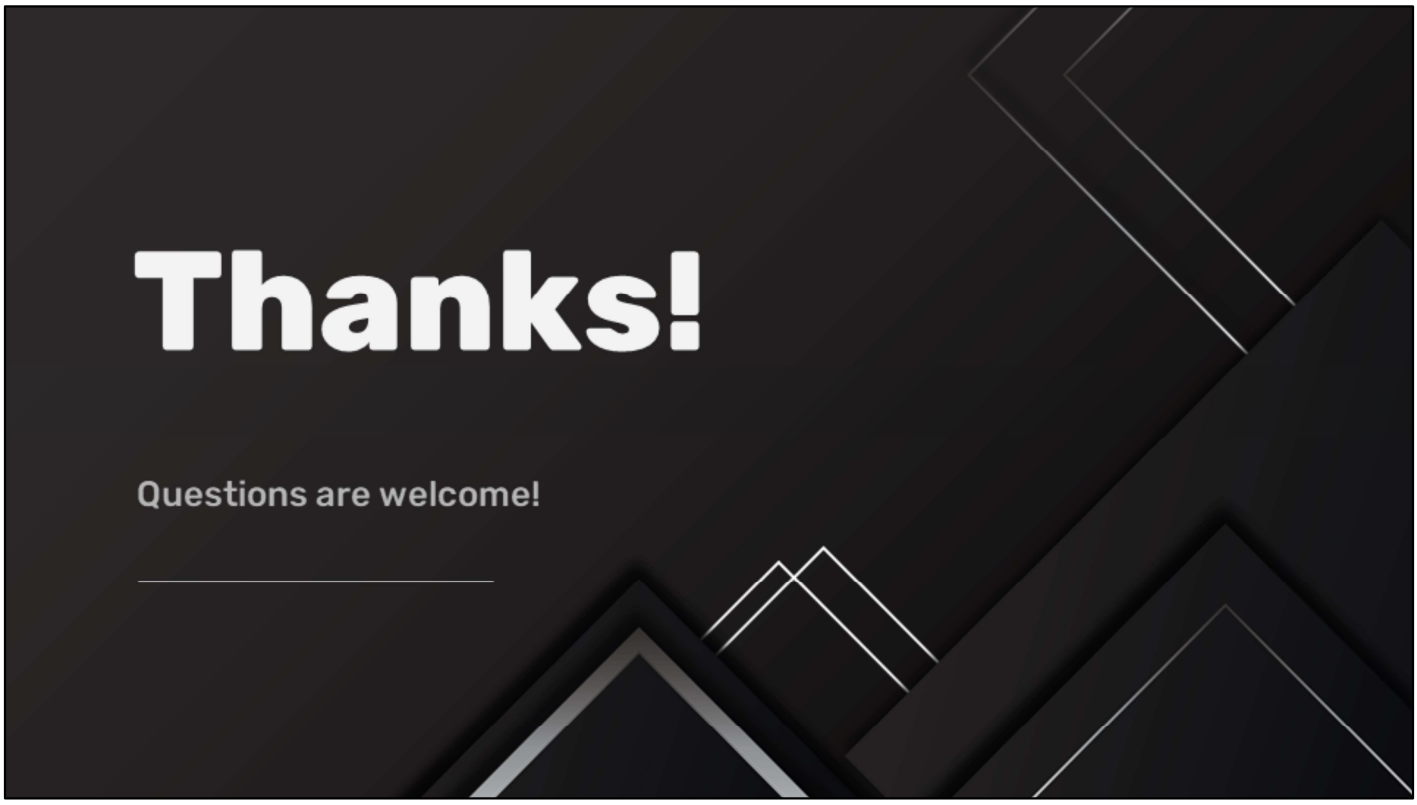
Second, from there the receiver can be re-tested to either validate that it still works or determine unforeseen shortcomings that need to be fixed.

Third, as proposed, this proof-of-concept requires tones that humans can hear simply because that's the same range of tones that smartphones and laptops can consistently detect. However, one could investigate the viability of connecting a more sensitive microphone to a phone or laptop and configuring the transmitter to broadcast supersonic frequencies so that the protocol is completely silent to human ears.

Fourth, the configuration parameters for the receiver could be further refined so that a consistent set of parameters could be found that work well for nearly all test cases.

Fifth, the current receiver requires the entire audio recording of the solve at the start of its analysis. This could be enhanced to support detecting the face turns of the cube in real time.

Finally, one could research strategies for transmitting/detecting cube rotations which are more challenging since no face alignments are altered.



And with that, I’d like to thank you for your attendance and your attention. I’ll turn the time back over to the thesis committee for any questions they have before we open the floor for questions from anyone in the audience.

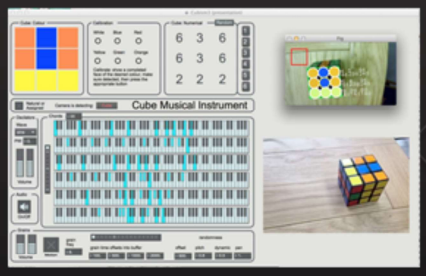
Extra Slides

Valuable slides that didn't get included in the final presentation.

If you are curious, you'll find the puzzles around you.
If you are determined, you will solve them.

Erno Rubik

ACADEMIC SMARTCUBES #1/2



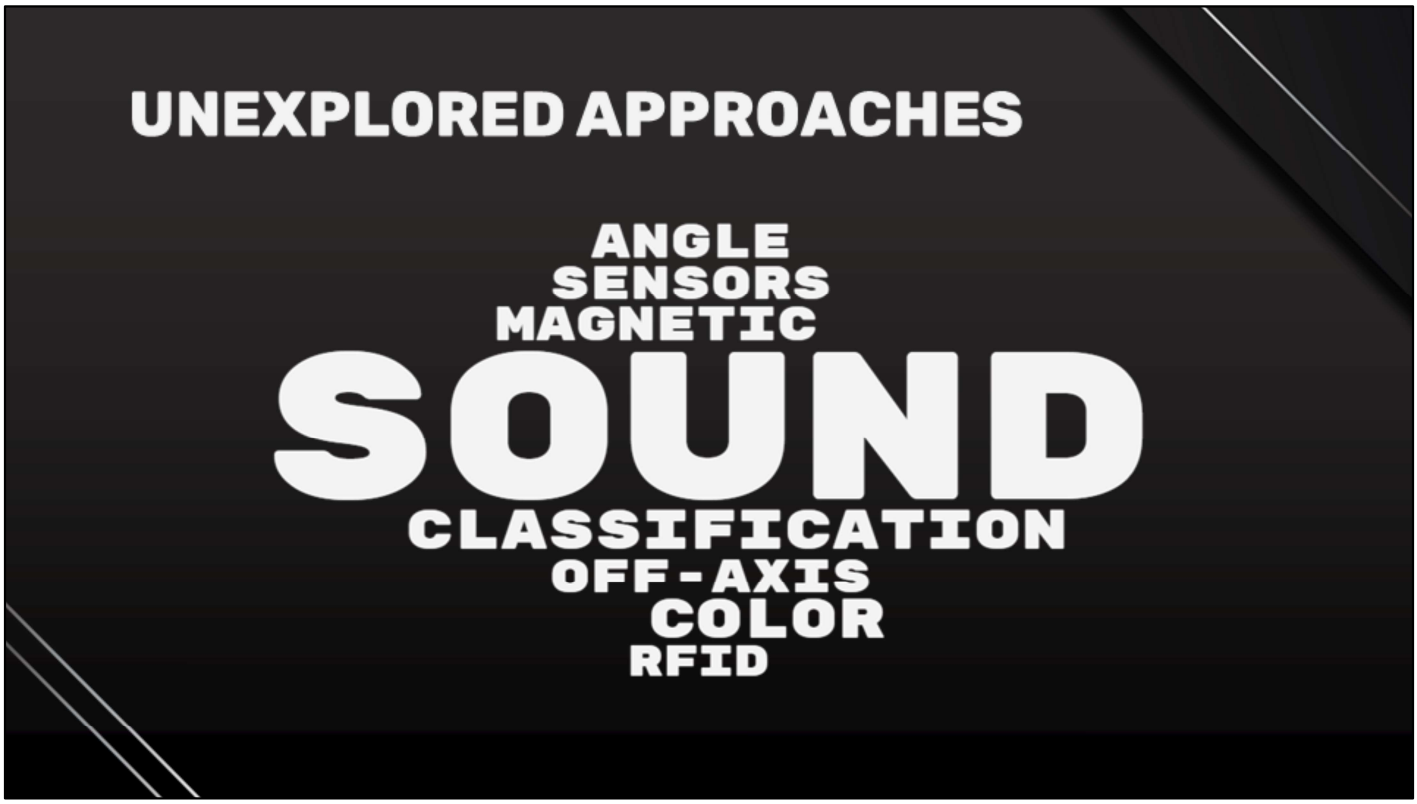
Music's Cube

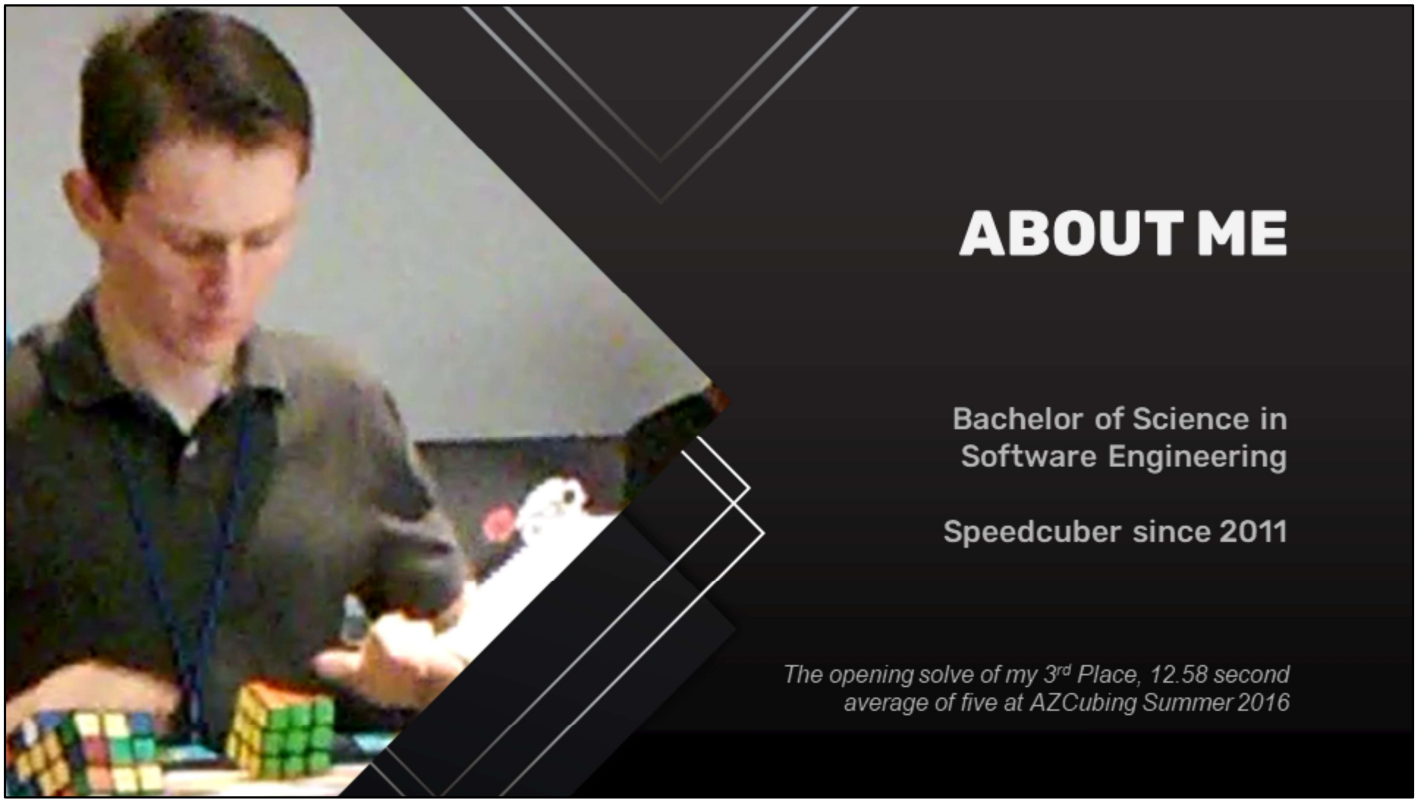
2017
Muscle-Tracking Armband
Low Accuracy

CubeHarmonic

2018
IM3D Trackers
No Errors Reported

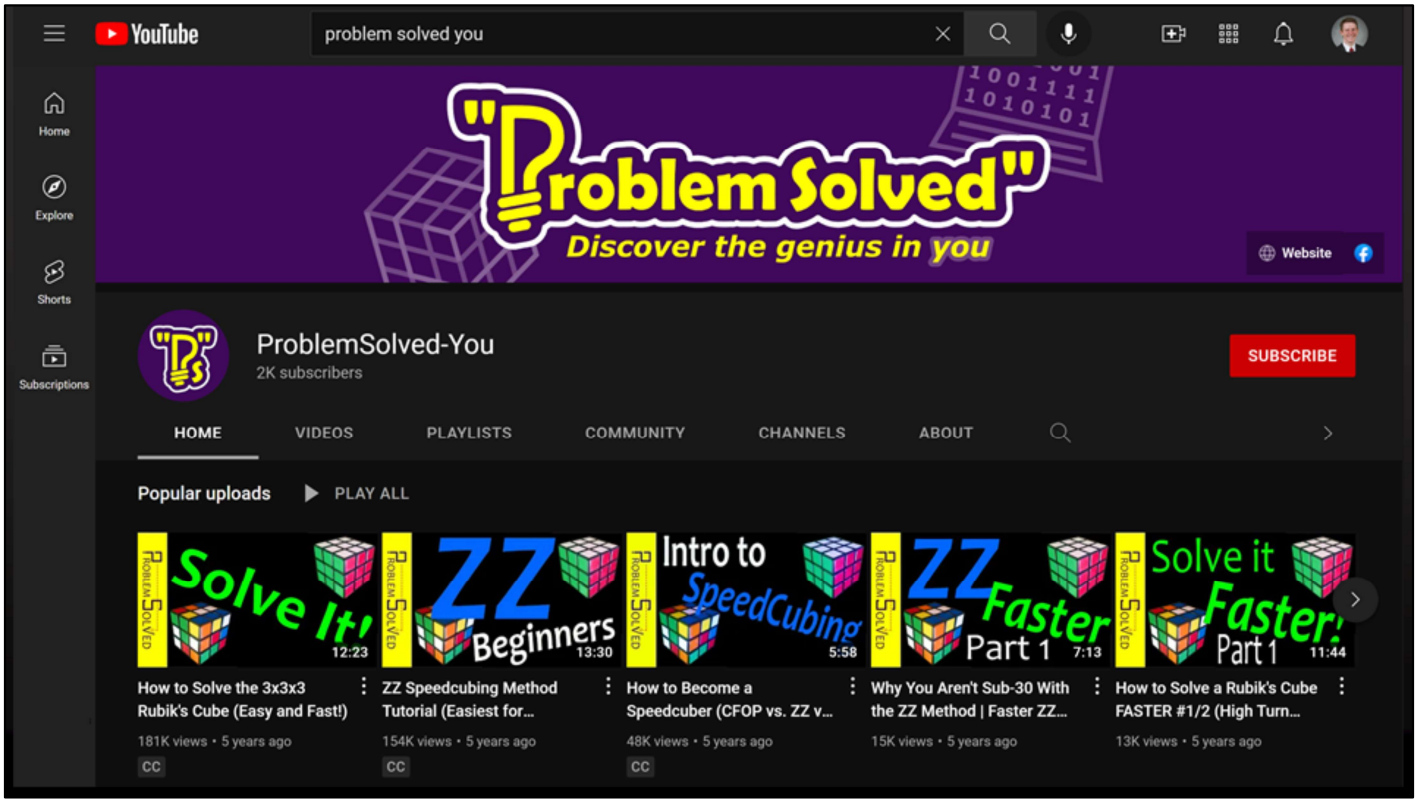
- IM3D: Irradiation of Materials in 3D
  - Magnetic Resonance


UNEXPLORED APPROACHES
ANGLE
SENSORS
MAGNETIC
SOUND
CLASSIFICATION
OFF-AXIS
COLOR
RFID



I first learned to solve a Rubik's Cube in middle school. Before school ended, my science teacher challenged me to a race after the summer break. Determined to win, I practiced all summer long getting my times down to just over a minute. By the end of the break, I was hooked on speedcubing and easily beat my teacher.

After graduating High School, I attended my first competition, where I took third in the state of Arizona with an average solution time of 12.5 seconds.



Around that same time, I started a YouTube Channel called “Problem Solved” where I documented many of the things I had learned about the cube. My videos have done very well, reaching over 400,000 viewers and 2,000 subscribers.

But since starting my collegiate studies as a Software Engineer, I’ve been really interested in the computational side of the Rubik’s Cube. At the same time, smart Rubik’s cubes that could automatically track each face turn and provide detailed solution analytics started to hit the market.

When it came time for me to choose an honors thesis topic, I decided I wanted to build my own smartcube!

THANKS!

Do you have any questions?

youremail@freepik.com
+91 620 421 838
yourcompany.com