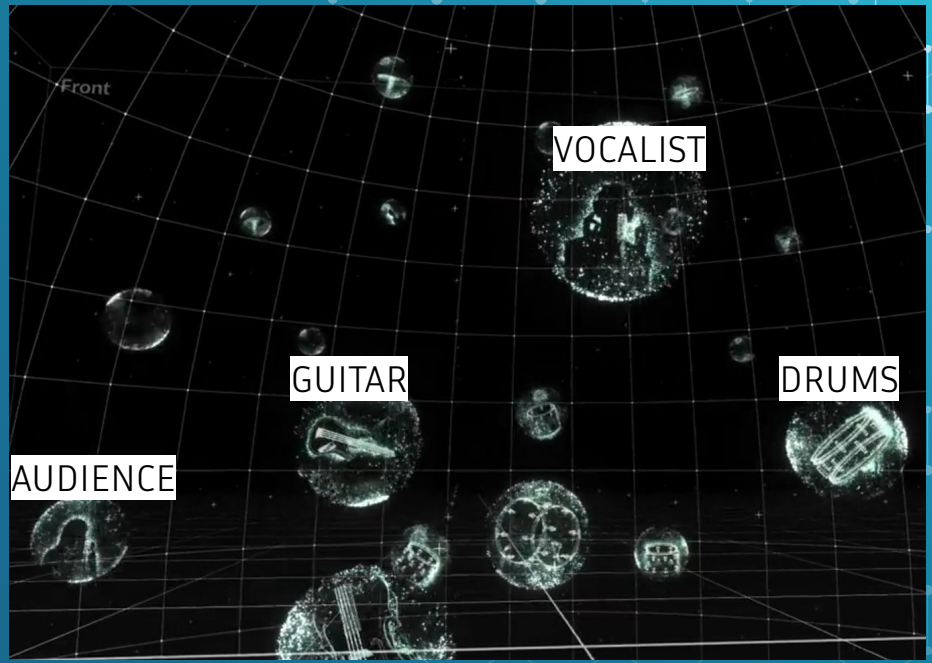
Spatial Audio Demo

Mentor: Prof. Truong Nguyen

Miriam Hamidi, Songyu Lu, Branson Beihl, Luke Yu, Scott He

Inspiration: Spatial Audio Applications

- In Music
  - Sony 360 Reality Audio: "**the future of music**"
  - AirPods pros newest update
- In Gaming
  - Playstation 5: 3D Soundscapes for most realistic video game experience yet

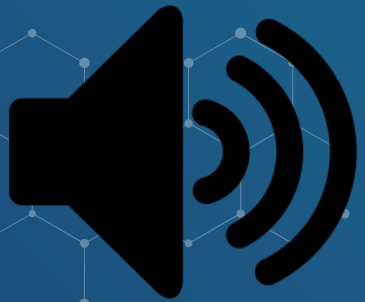


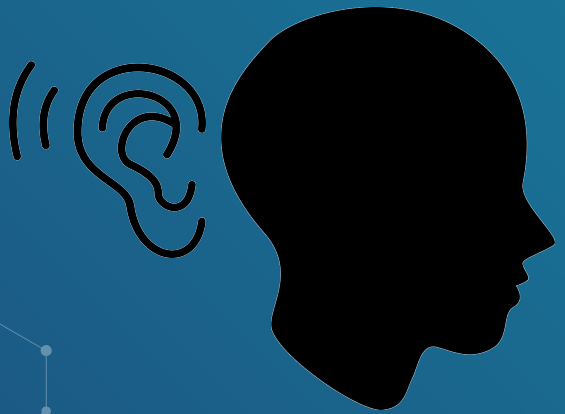
Sony's animation of their 360 Reality Audio experience

How it Works: HRTF

The Head Related Transfer Function: describes how a sound wave changes from its source to the inner ear

Sound leaves source → Sound wave is reflected/ refracted by anatomy of individual → Unique signal for unique angle of sound source, brain uses these cues for sound localization in 3D



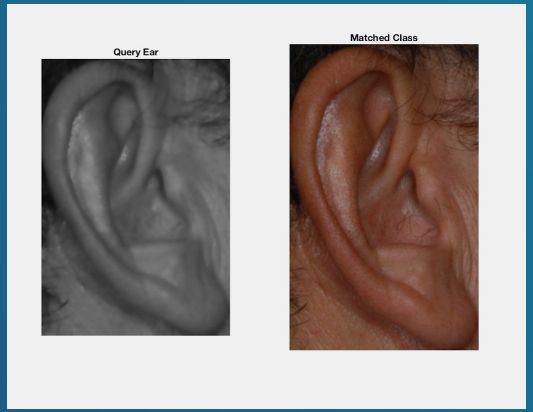


Structure: Educational Module

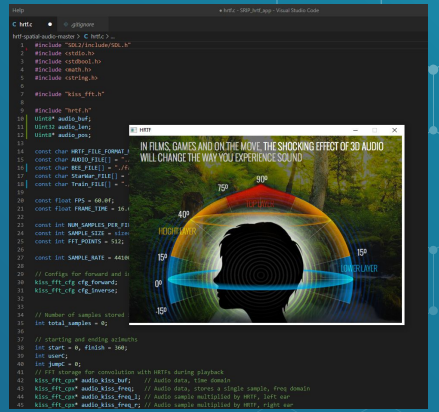
- Purpose: create an educational tool for students to
  - experiment with spatial audio personalized to their anthropometric features
  - understand the math/ECE concepts behind the process

User inputs image of ear

Ear/HRTF match identified (Matlab)

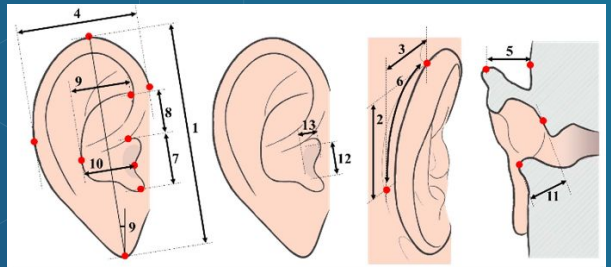


Personalized spatial audio is rendered

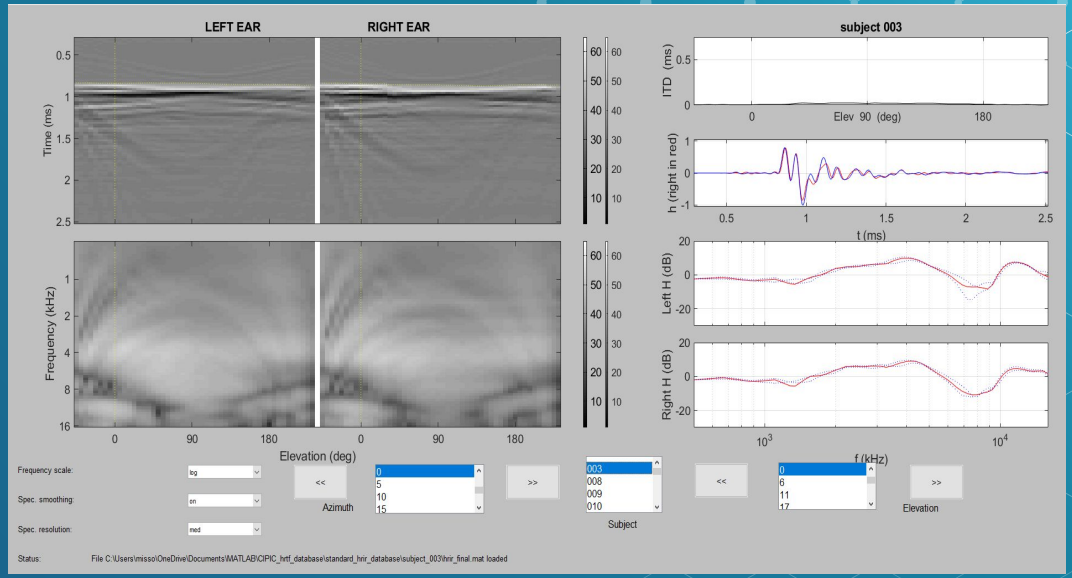


Still in progress!

Why Ear Matching?



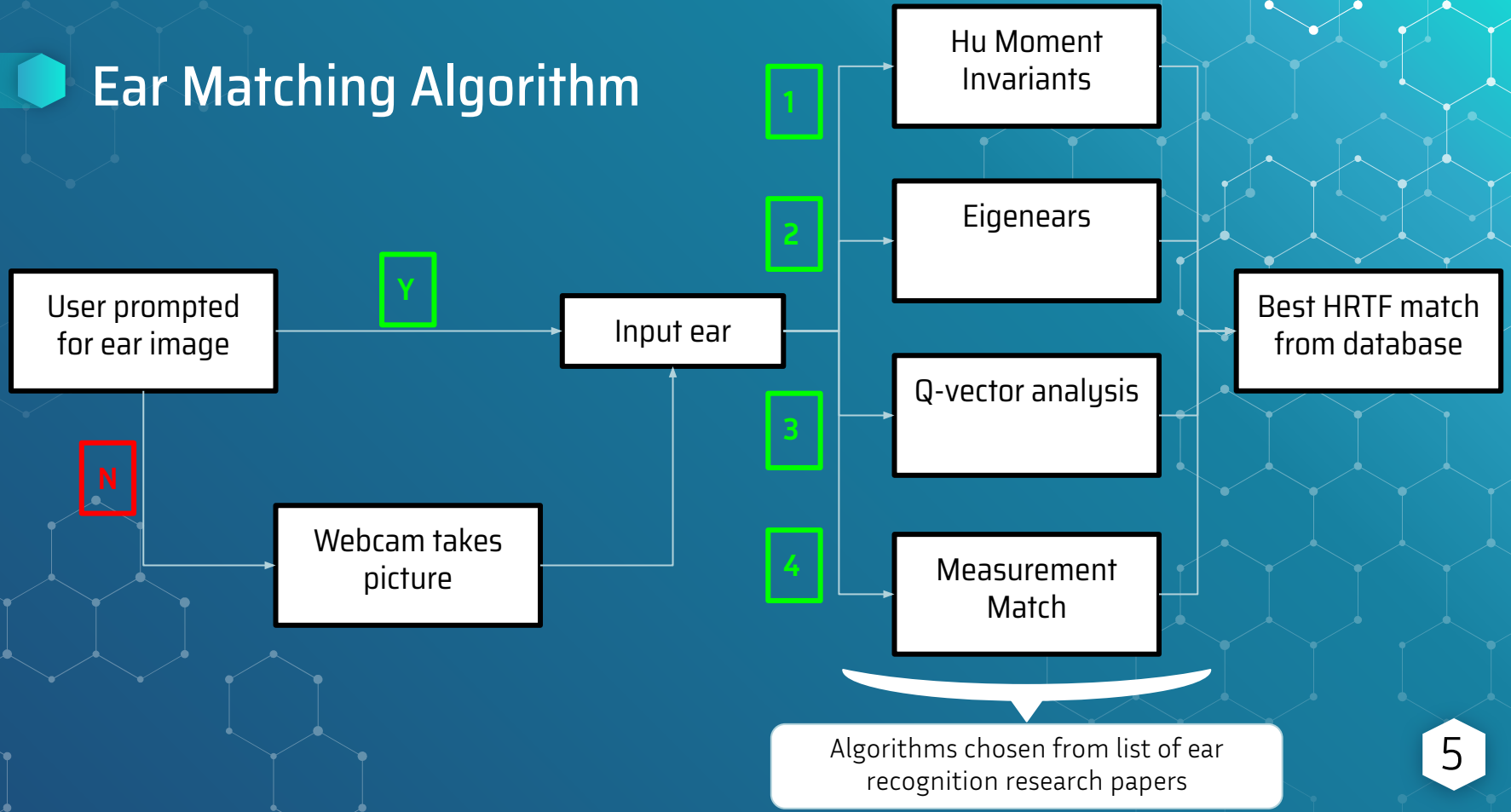
Category		No	Ear dimensions
Ear dimensions	Length	1	ear length
		2	otobasion superius to otobasion posterius horizontal length†
		3	otobasion superius to otobasion posterius vertical length†
	Width	4	ear breadth
		5	ear protrusion
	Arc	6	upper otobasion arc†
Concha dimensions	Length	7	cavum concha length
		8	superior cavum concha to anterior cymba concha length†
		9	posterior concha to anterior cymba concha lengt†
	Width	10	cavum concha width
	Depth	11	cavum concha depth
Ear canal dimensions	Length	12	ear canal length†
	Width	13	ear canal width†



- Measuring process can be complex
- Computing HRTF from measurements is difficult and time-consuming

SOLUTION: identifying a “best match” from measured HRTFs
CIPIC database provides ear measurements and HRTFs [1]

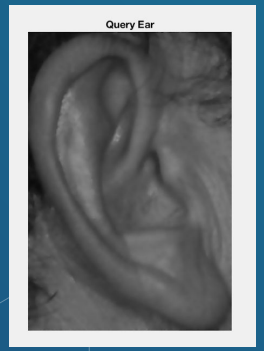
Ear Matching Algorithm

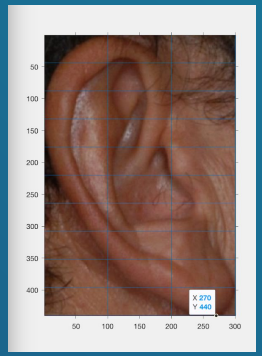


Matching: Hu Moment Invariants[2]

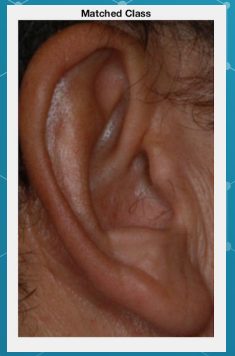
Image Moments: weighted average of pixel intensities

Image Acquired → Normalization Filter Applied → Image Segmented → Moment Features Extracted → Best HRTF match identified

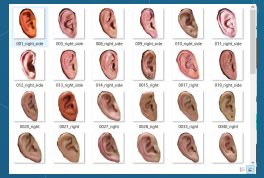




h[0] = 0.00162663
h[1] = 0.00834555

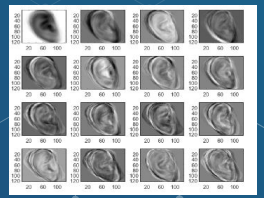


Matching: Eigenears[3]



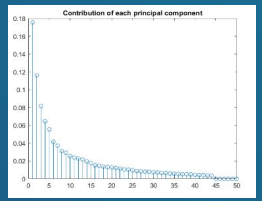
Database of ear images

Input image



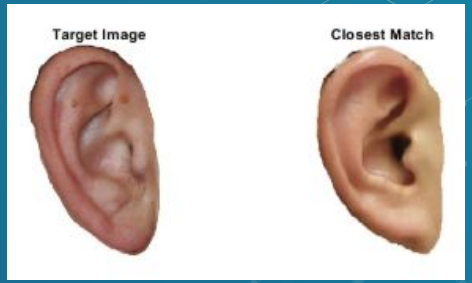
Create a set of Eigenears using PCA

Take the first *k* eigenears with most contribution



Calculate weight vectors

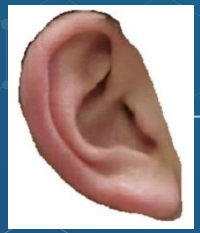
Take closest database ear



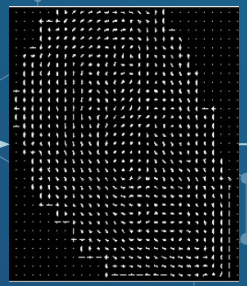
Matching: Q-Vector Analysis[4]

HOG *(Histogram of Oriented Gradient)* is a process that defines the curvature and the depths of the ears according to its shape

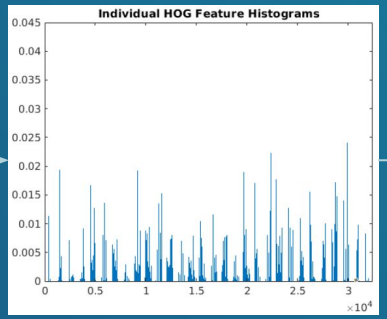
Original Image



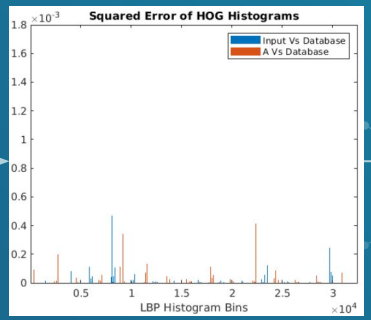
HOG Features



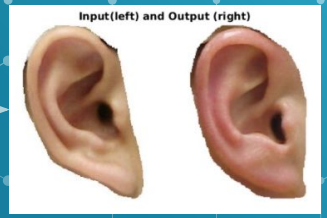
Gather the features into histogram



Comparing the features & pick the better match



Repeating the previous step through database



Best Matching as output

Matching: Measurement[5]

User measures their ear for d1 -d8

User inputs specified ear measurements

Closest database match found

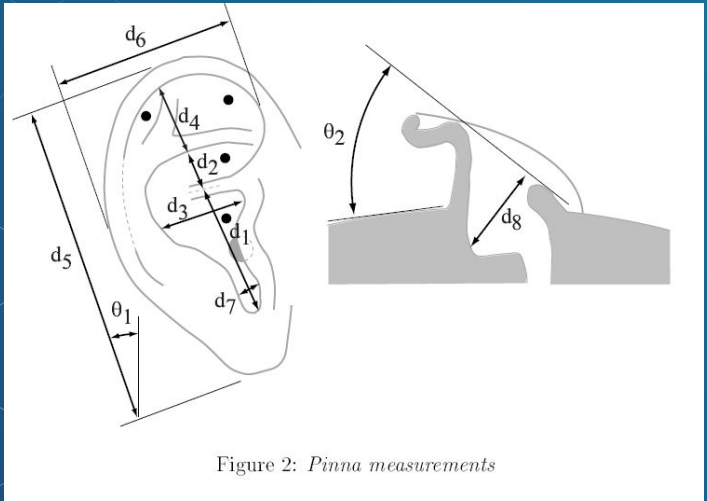
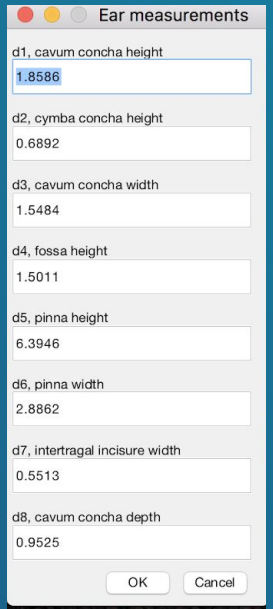
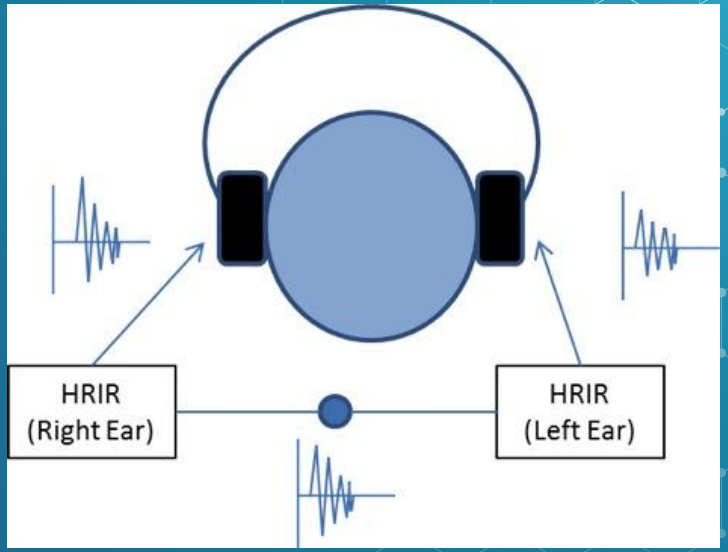


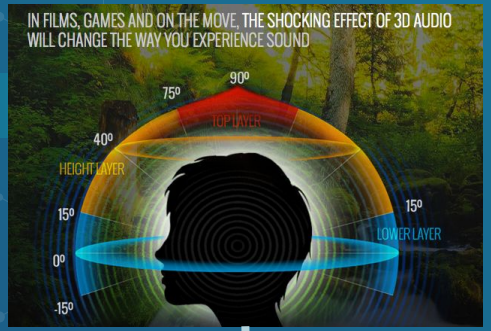
Figure 2: *Pinna measurements*



Audio Rendering: Our Application

- 2-channel
- KISSFFT for FFT computation
- SDL2 for
  - GUI design
  - Audio output





Menu:

Press 1 to choose path

Press 2 to choose audio

Press 3 to choose speed

DEMO

Choose the audio you want to hear:

0 for beep

1 for Star War

2 for train sound

3 for bee

Press 0 to choose standard path

Press 1 to choose customized path

Press enter to input customized azimuth

Enter 3 digits to use as starting azimuth

Press enter to store starting azimuth

Enter 3 digits to use as ending azimuth

Press escape to store ending azimuth and return to menu

Choose Speed Level:

Default - 5 degree increment

1 - 10 degree increment

2 - 20 degree increment

3 - 30 degree increment

4 - 40 degree increment

Plan moving forward

- Finish Audio Rendering
- Test Audio Localization Improvement
- Build Educational Side of Module
- Implement as Workshop

Thanks for Listening!

Special thanks to Professor Nguyen for advising us and for the SRIP program for supporting us!

Sources

1. CIPIC
2. Hu Moment Paper
3. Eigenears
4. Q-vector Analysis Paper
5. Measurement Match Paper
6. Original Github Program