Room of Requirement

EE 522 Spring 2020: Final Project
Veena Vijai
vijai@usc.edu

INTRODUCTION

Room impulse response characterizes the behaviour of a room for a particular source and location of microphone/subject. Given a desired frequency response, it is of potential interest to find a room which 'sounds similar,' i.e. modifies the source in a similar way.

Rooms differ in properties such as shape and size, which determine their surface area and volume, and also in the materials they contain, which determines its overall 'absorptivity.' It is an interesting problem to determine if rooms of different shape, size, and absorption can sound similar to each other as well.

For a given room, one can also explore the variation of the frequency response with increasing size of the room and increasing absorption coefficient, keeping all other parameters constant.

OBJECTIVES

1. Create a dataset of realistic rooms with varying size, shape, and absorption coefficient
2. Calculate and store the impulse response in time and frequency domain for the dataset
3. Given a different room, recommend the three most similar rooms from the dataset
4. Analyze the variation of the frequency response with room size and absorption coefficient

TOOLS

Fig 1: "Pyroomacoustics is a package for audio signal processing for indoor applications", developed by EPFL.

pyroomacoustics has a 'room' class which takes room parameters as input, notably coordinates which specify the room corners. Then, the extrude() function can be used to lift up the room by a particular height.

```python
# Create a 3D room

# specify corners
c1 = [0,0]
c2 = [0,3]
c3 = [5,3]
c4 = [5,1]
c5 = [3,1]
c6 = [3,0]
corners = np.array([c1, c2, c3, c4, c5, c6]).T  # [x,y]
print(corners)

# specify source
fs, signal = wavfile.read("arctic_a0010.wav")

# create room
room = pra.Room.from_corners(corners, max_order=8, absorption=0.2)
#room = pra.Room.from_corners(corners, fs=fs, max_order=8, absorption=0.2)

# lift room up by fixed height
room.extrude(2.)

# add source
room.add_source([1, 1, 1.5], signal=signal)

# add mic
R = np.array([[1], [1], [0.5]])  # [[x], [y], [z]]
room.add_microphone_array(pra.MicrophoneArray(R, room.fs))

# display room
fig, ax = room.plot()
ax.set_xlim([0, 5])
ax.set_ylim([0, 3])
ax.set_zlim([0, 2]);
```

Fig 2: Creating a room and adding source and mic

In order to calculate the room impulse response (RIR), pyroomacoustics also takes an absorption coefficient as parameter, along with a source file and its coordinates, and the coordinates of a microphone/microphone array. It uses an image source model (ISM), for which the maximum order can be specified as well. In the experiments below, the order was chosen to be the default, which is 8 reflections.

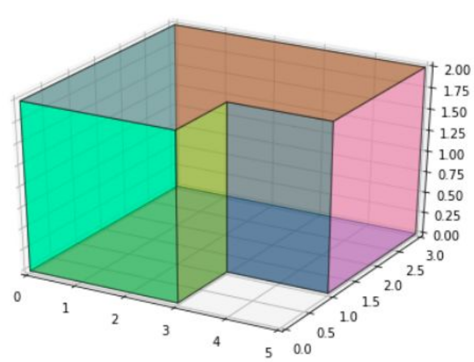


Fig 3: A room with 6 specified corners and height of 2 units

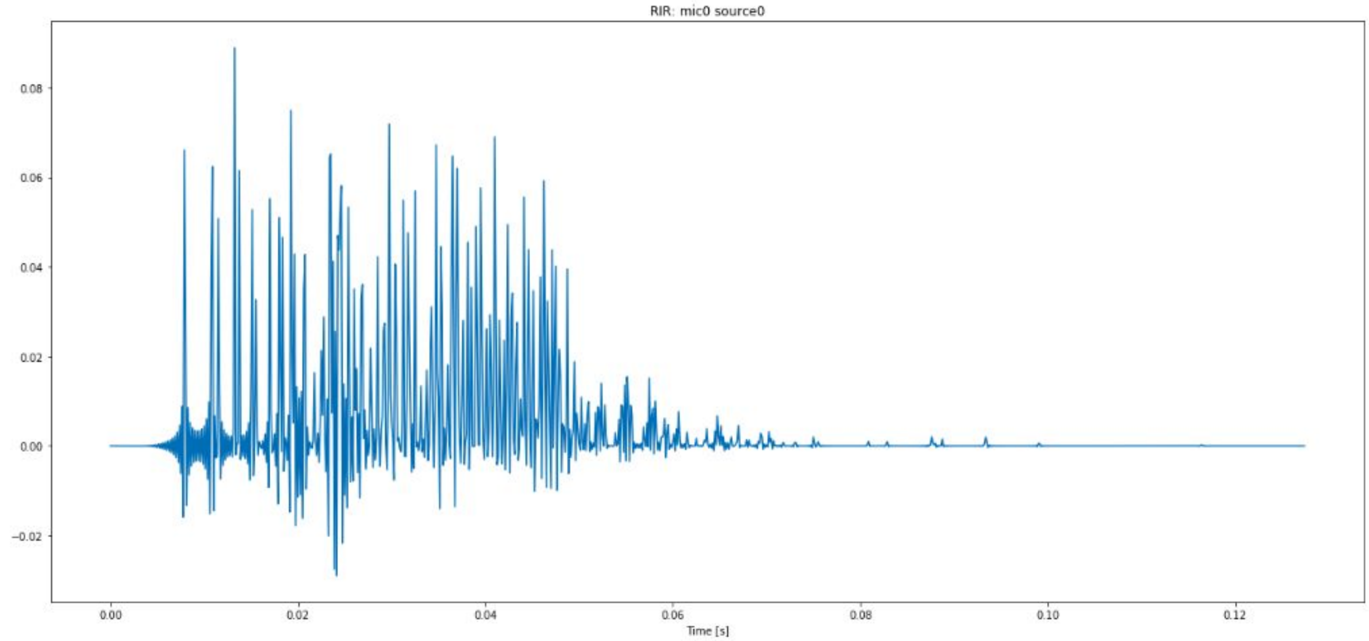


Fig 4: RIR in time domain, plotted within pyroomacoustics

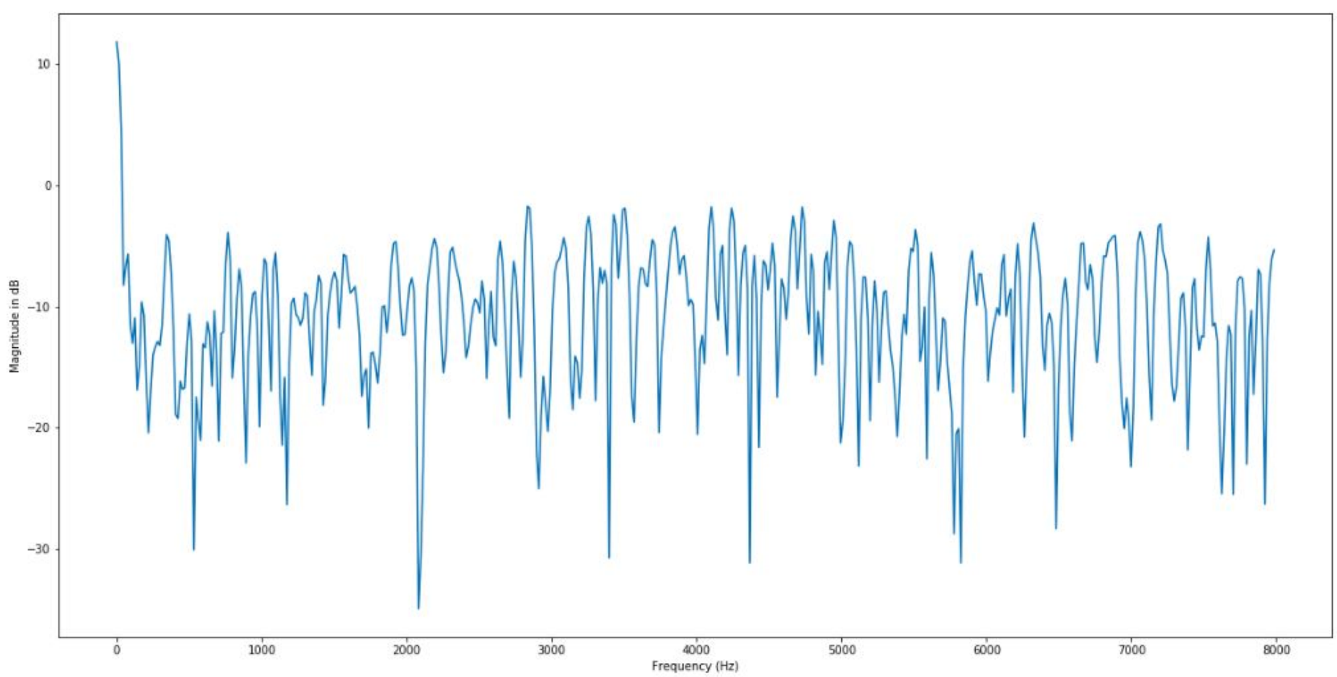


Fig 5: RIR in frequency domain

DATASET

In order to design realistic rooms, realistic designs were required. For this purpose, floor plans and designs from the following sources were considered: USC Housing, USC ITS Room Finder, and pinterest.

Fig 6: Sample USC Housing floor plan

Classroom/house room size:

- 6 floor plans were shortlisted from the above sources
- Each shape could have 5 possible sizes, designed to be proportional
- Ceiling height was fixed as 10 units (standard ceiling height is about 10 ft)
- Their absorption coefficients varied between 0.1 to 0.8 in increments of 0.05
- No. of rooms = 6 designs * 5 sizes * 15 coeffs = 450 rooms
- Source (x, y, z) in front left corner at 5 ft height and mic (x2, y2, z2) in the middle of the room at 2 ft height
- Surface area and vol calculated for each room, approximated for irregular shapes

ROOM 1

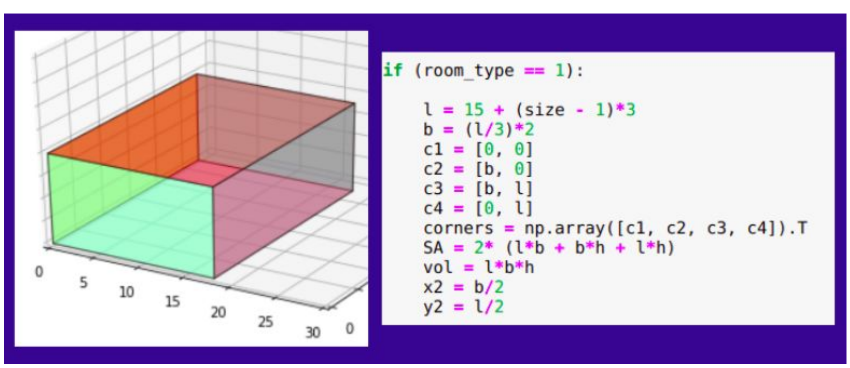
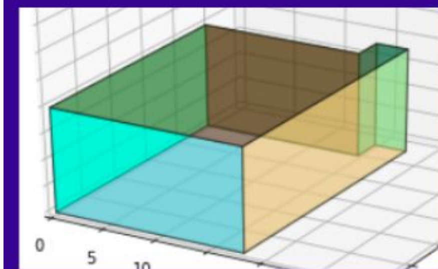


Fig 7: Room 1 design and parameters

ROOM 2

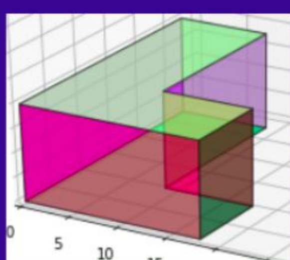



```python
elif (room_type == 2):

    l = 15 + (size - 1)*3
    b = (l/3)*2
    c1 = [0, 0]
    c2 = [b, 0]
    c3 = [b, l]
    c4 = [b-3, l]
    c5 = [b-3, l-3]
    c6 = [0, l-3]
    corners = np.array([c1, c2, c3, c4, c5, c6]).T
    SA = b*h + l*h + 6*h + h* (b-3 + l-3) + (l-3)*b + 9
    vol = ((l-3)*b + 9)*h
    x2 = b/2
    y2 = l/2
```

Fig 8: Room 2 design and parameters

ROOM 3

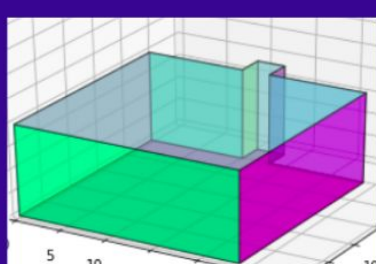



```python
elif (room_type == 3):

    l = 15 + (size - 1)*3
    b = (l/3)*2
    b2 = l/3
    c1 = [0, 0]
    c2 = [b, 0]
    c3 = [b, b2]
    c4 = [b - b2, b2]
    c5 = [b - b2, l]
    c6 = [0, l]
    corners = np.array([c1, c2, c3, c4, c5, c6]).T
    SA = b*h + l*h + 2*b2*h + h* (b-b2 + l-b2) + b*b2 + (l-b2)*(b-b2)
    vol = h*((l-b2)*(b-b2) + b*b2)
    x2 = b/3
    y2 = l/2
```

Fig 9: Room 3 design and parameters

ROOM 4

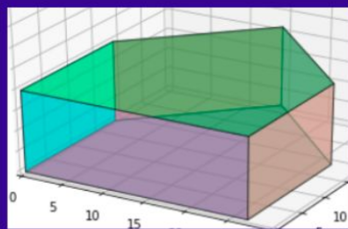



```python
elif (room_type == 4):

    l = 12 + (size - 1)*3
    x = (l - 3)/2
    c1 = [0, 0]
    c2 = [l, 0]
    c3 = [l, l]
    c4 = [l - x, l]
    c5 = [l - x, l + 3]
    c6 = [l - x - 3, l + 3]
    c7 = [l - x - 3, l]
    c8 = [0, l]
    corners = np.array([c1, c2, c3, c4, c5, c6, c7, c8]).T
    SA = 3*l*h + 2*x*h + 3*3*h + l*l + 9
    vol = h*(l*l + 9)
    x2 = l/2
    y2 = l/2
```

Fig 10: Room 4 design and parameters

ROOM 5

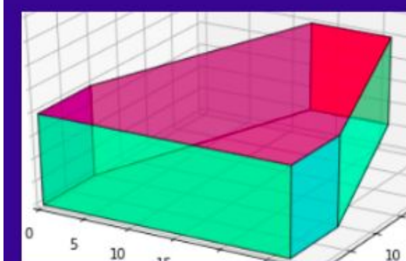


```
elif (room_type == 5):

    l = 15 + (size - 1)*3
    b = (l/3)*2
    b2 = l/2
    c1 = [0,0]
    c2 = [l, 0]
    c3 = [l, b]
    c4 = [l/2, b + b2]
    c5 = [0, b]
    corners = np.array([c1, c2, c3, c4, c5]).T
    SA = 2*b*h + l*h + 2*(l/math.sqrt(2))*h + (l*b + 0.5*l*b2)
    vol = l*b*h + 0.5*l*b2*h
    x2 = l/2
    y2 = b/2
```

Fig 11: Room 5 design and parameters

ROOM 6



```
elif (room_type == 6):

    l = 15 + (size - 1)*3
    b = l/3
    c1 = [0,0]
    c2 = [l, 0]
    c3 = [l, b]
    c4 = [l - b, l + b]
    c5 = [b, l + b]
    c6 = [0, b]
    corners = np.array([c1, c2, c3, c4, c5, c6]).T
    SA = 2*b*h + l*h + (l -2*b)*h + 2*(b/math.sqrt(2))*h + l*b + 0.5*b*(l + l - 2*b)
    vol = h*(l*b + 0.5*b*(l + l - 2*b))
    x2 = l/2
    y2 = b
```

Fig 12: Room 6 design and parameters

Auditorium hall size:

- 3 floor plans were added, each with only one size, and ceiling height at 20 units
- Their absorption coefficients varied between 0.1 to 0.8 in increments of 0.05
- No. of rooms = 3 designs * 1 size * 15 coeffs = 45 rooms
- Source (x, y, z) in the middle of the stage at 7 ft height and mic (x2, y2, z2) in the middle of the audience at 10 ft height
- Surface area and vol calculated for each room, approximated for irregular shapes

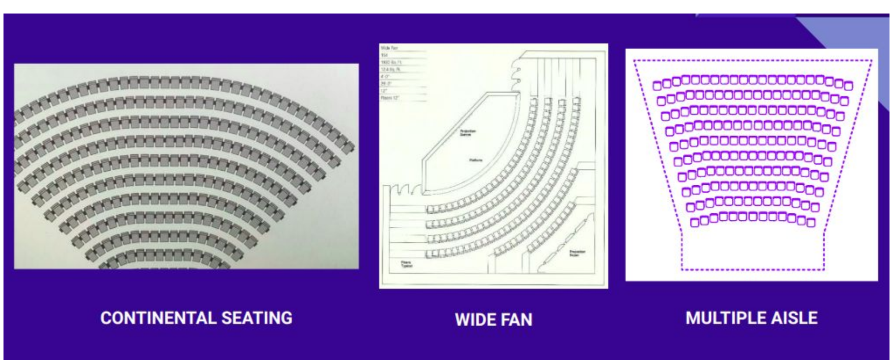
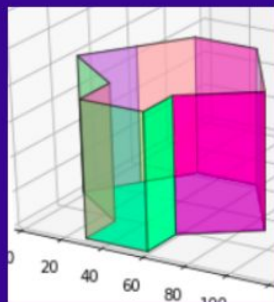


Fig 13: Auditorium designs

<u>ROOM 7</u>

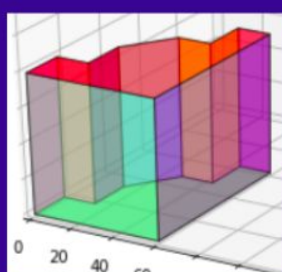



```
elif (room_type == 7):

    l = 30
    b = l/2
    c1 = [l, 0]
    c2 = [2*l, 0]
    c3 = [2*l, b]
    c4 = [3*l, 2*b]
    c5 = [2.5*l, 3*b]
    c6 = [2*l, 4*b]
    c7 = [l, 4*b]
    c8 = [0.5*l, 3*b]
    c9 = [0, 2*b]
    c10 = [l, b]
    corners = np.array([c1, c2, c3, c4, c5, c6, c7, c8, c9, c10]).T
    r = 3*l/math.sqrt(2)
    SA = (math.pi*r/2)*h + 2*r*h + math.pi*r*r/4
    vol = h * math.pi*r*r/4  # approx
    x = 1.5*l
    y = 0.5*b
    z = 7
    x2 = 1.5*l
    y2 = 2*b
    z2 = 10
```

Fig 14: Room 7 design and parameters

<u>ROOM 8</u>

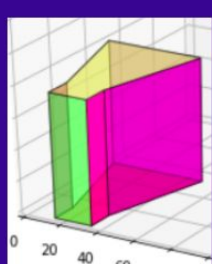



```
elif (room_type == 8):

    l = 60
    b = l/4
    c1 = [0, 0]
    c2 = [l, 0]
    c3 = [l, l]
    c4 = [3*b, l]
    c5 = [3*b, 3*b]
    c6 = [2*b, 3*b]
    c7 = [b, 2*b]
    c8 = [b, b]
    c9 = [0, b]
    corners = np.array([c1, c2, c3, c4, c5, c6, c7, c8, c9]).T
    SA = h*(2*l + 7*b) + l*l/2   #approx
    vol = h*l*l/2  #approx
    x = 2*b
    y = l/3
    z = 7
    x2 = 3*b
    y2 = l/3
    z2 = 10
```

Fig 15: Room 8 design and parameters

<u>ROOM 9</u>




```
elif (room_type == 9):

    l = 20
    b1 = 10
    b2 = 15
    c1 = [b2, 0]
    c2 = [b2 + l, 0]
    c3 = [b2 + l, b1]
    c4 = [b2 + l + b2, b2 + l + b2]
    c5 = [0, b2 + l + b2]
    c6 = [b2, b1]
    corners = np.array([c1, c2, c3, c4, c5, c6]).T
    SA = l*h + 2*b1*h + h*(b2 + l + b2) + 2*52*h + l*b1 + 0.5*(l + 2*b2 - b1)*(l + l + 2*b2)
    vol = h*(l*b1 + 0.5*(l + 2*b2 - b1)*(l + l + 2*b2))
    x = b2 + l/2
    y = b1/2
    z = 7
    x2 = b2 + l/2
    y2 = 30
    z2 = 10
```

Fig 16: Room 9 design and parameters

In total, there are 495 rooms in the dataset.

SIMILARITY CALCULATION

The aim is to compare correlation of the 'query room' to the rooms in the dataset, to find the three most similar rooms.

The 'query rooms' will be designed such that they have the same shape as a room in the dataset, but different size and absorption, so there is no exact match.

SIMILARITY: FREQUENCY DOMAIN

In order to compare frequency response, a problem arises: the frequency responses have varying lengths. In order to account for this, we can group the log energy into bins by summing them. The details are:

- Our sampling frequency is 16 kHz, therefore Nyquist frequency is 8 kHz
- If we choose bins of width 100 Hz, we have 80 bins
- We sum log energy for each of the 80 bins
- Now, the varying-length frequency response is an 80-dim vector
- We can calculate Pearson's correlation coefficient between the query room's 80-dim frequency response and that of the dataset

```python
# helper function to compute energy in bins

def get_fr_bin_indices(FR, fs, num_bins, multiples=100):


    freq_bins = fs*np.arange(FR.shape[-1])/float(2*FR.shape[-1])
    for f in freq_bins:
        f = round(f)

    #find the frequency index numbers to group in bins of 10 kHz width
    # from 0-100 Hz, 100-200 Hz, up to 8 kHz
    stop_freq_arr = np.zeros((num_bins, ))
    start_freq_arr = np.zeros((num_bins, ))

    for i in range(num_bins):

        stop_freq = multiples*(i + 1)
        stop_freq_arr[i] = freq_bins.searchsorted(stop_freq, side='right') - 1

        if (i < num_bins - 1):
            start_freq_arr[i + 1] = stop_freq_arr[i] + 1

    return start_freq_arr.astype(int), stop_freq_arr.astype(int)
```

Fig 17: Code to calculate starting and ending indices to sum up energy in 100 Hz bins

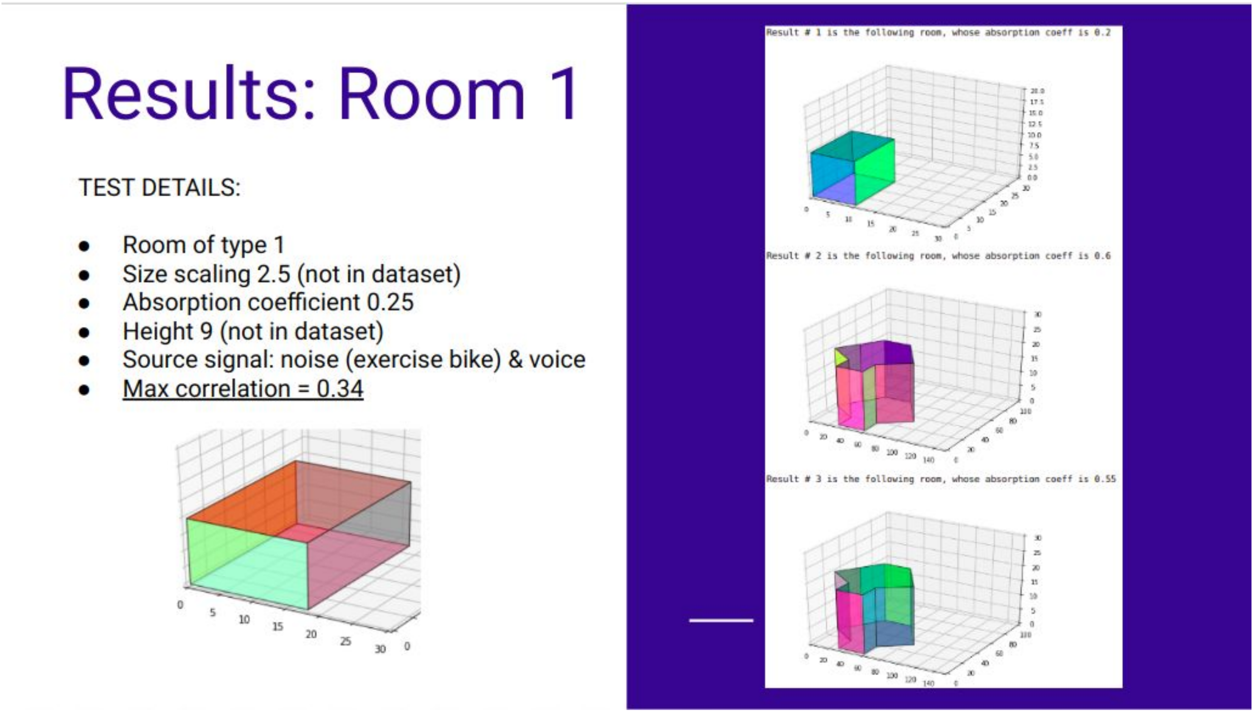


Fig 18: Top 3 rooms similar to a room of shape Room 1

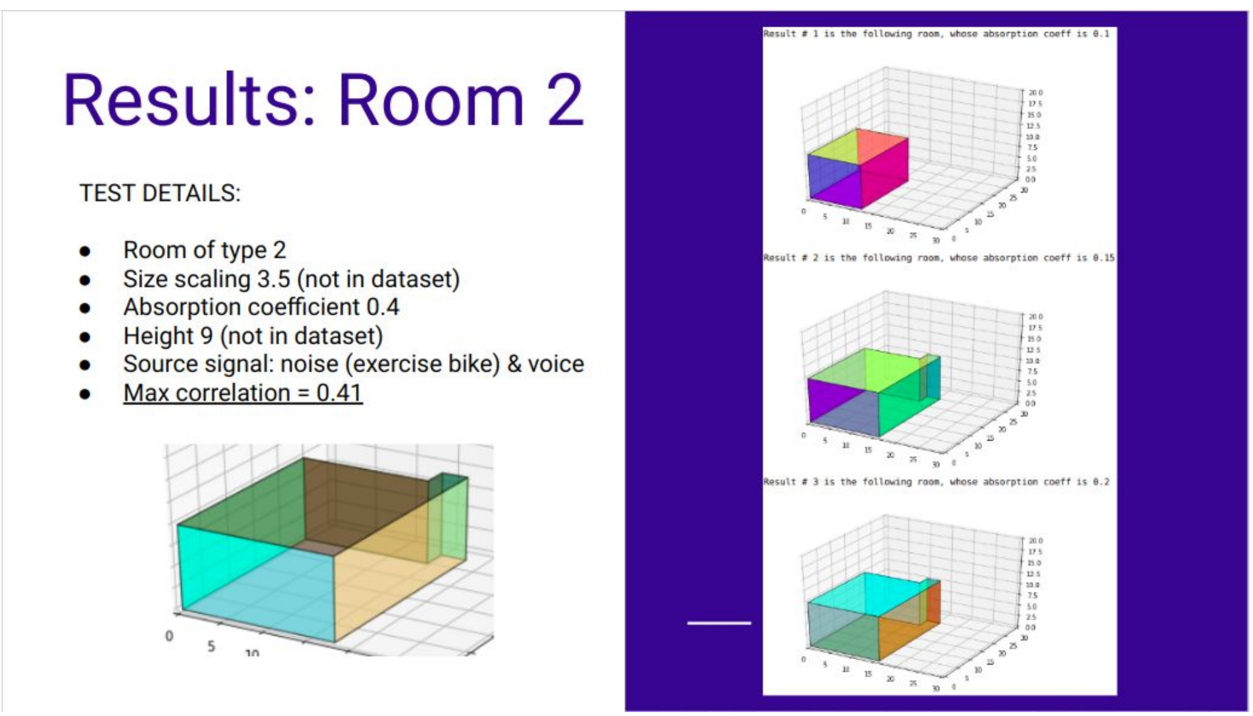
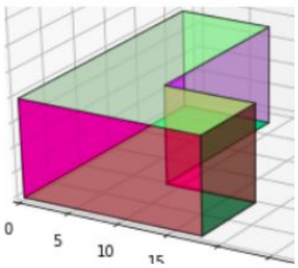


Fig 19: Top 3 rooms similar to a room of shape Room 2

Results: Room 3

TEST DETAILS:

- Room of type 3
- Size scaling 1.5 (not in dataset)
- Absorption coefficient 0.5
- Height 9 (not in dataset)
- Source signal: noise
- <u>Max correlation = 0.497</u>



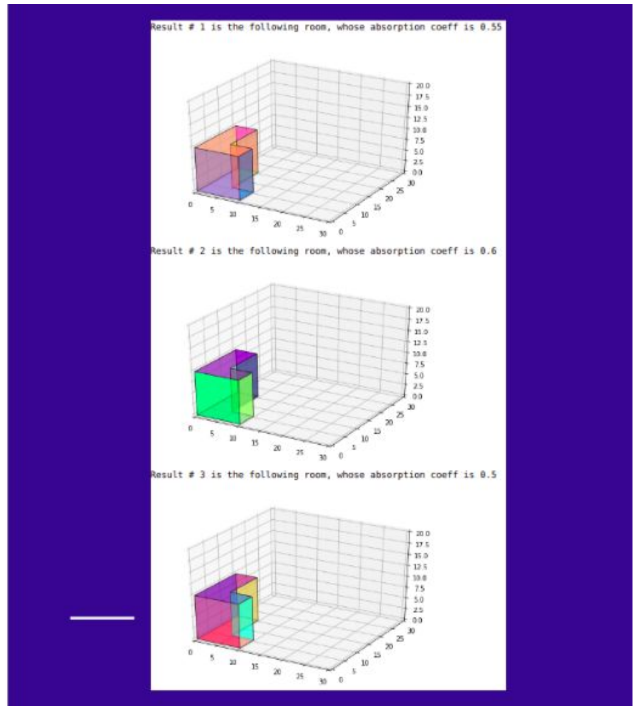
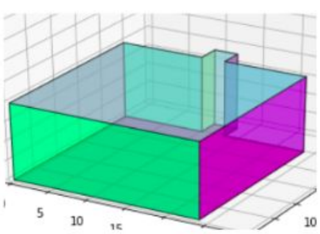


Fig 20: Top 3 rooms similar to a room of shape Room 3

Results: Room 4

TEST DETAILS:

- Room of type 4
- Size scaling 1.75 (not in dataset)
- Absorption coefficient 0.6
- Height 9 (not in dataset)
- Source signal: noise
- <u>Max correlation = 0.36</u>



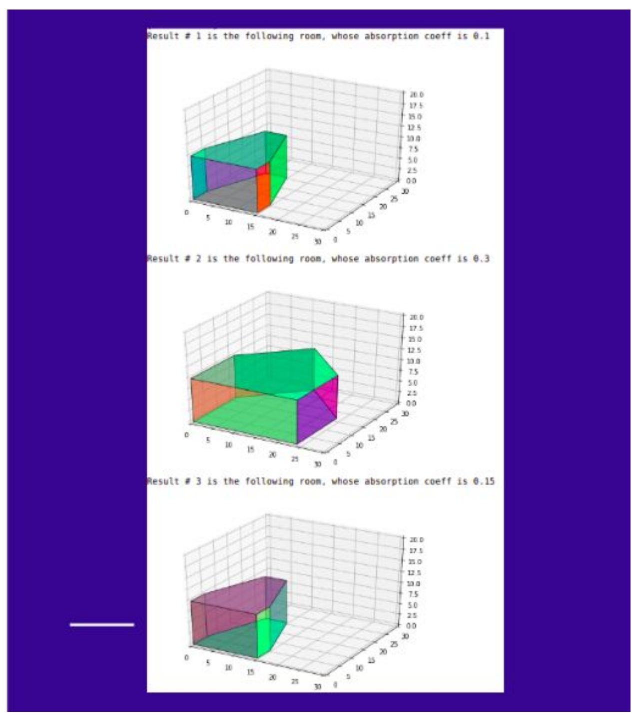
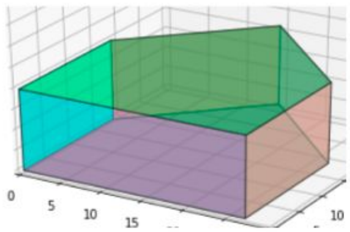


Fig 21: Top 3 rooms similar to a room of shape Room 4

Results: Room 5

TEST DETAILS:

- Room of type 5
- Size scaling 3.25 (not in dataset)
- Absorption coefficient 0.65
- Height 9 (not in dataset)
- Source signal: noise
- <u>Max correlation = 0.33</u>



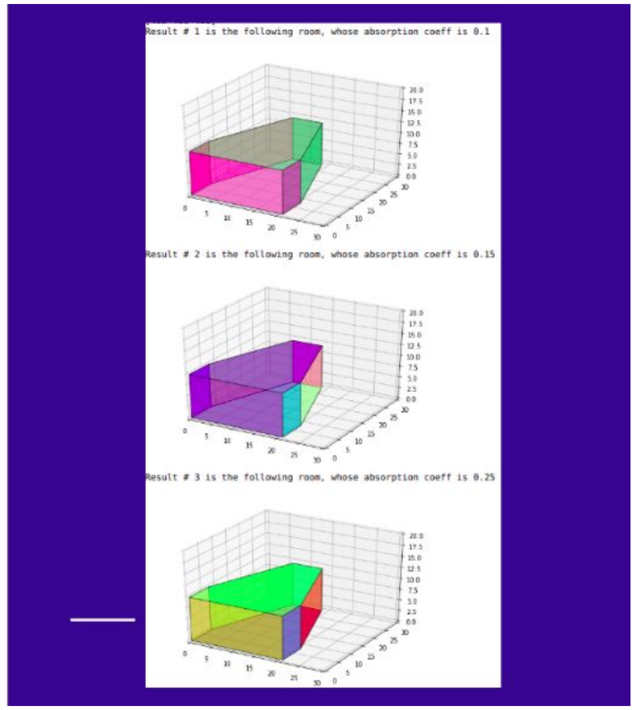
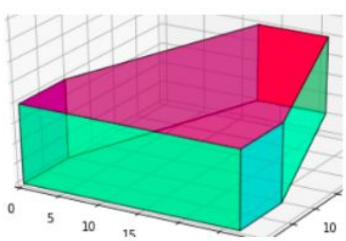


Fig 22: Top 3 rooms similar to a room of shape Room 5

Results: Room 6

TEST DETAILS:

- Room of type 6
- Size scaling 3.25 (not in dataset)
- Absorption coefficient 0.7
- Height 9 (not in dataset)
- Source signal: noise
- <u>Max correlation = 0.33</u>



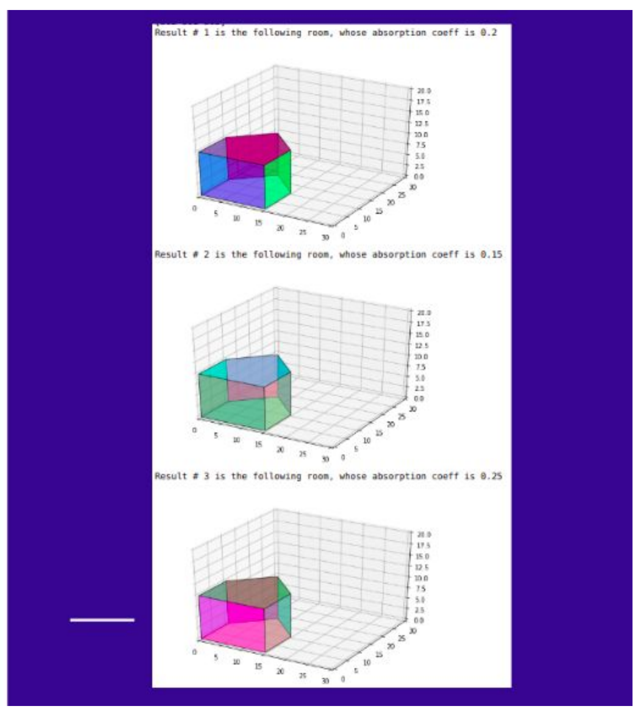


Fig 23: Top 3 rooms similar to a room of shape Room 6

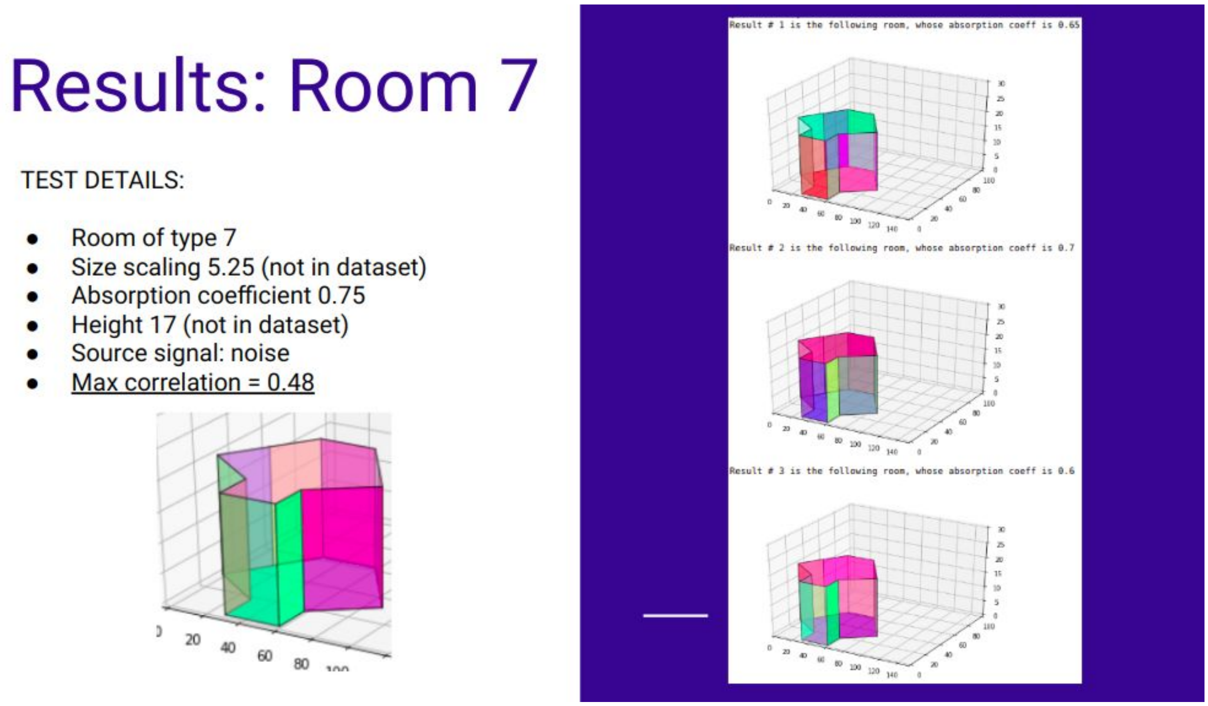


Fig 24: Top 3 rooms similar to a room of shape Room 7

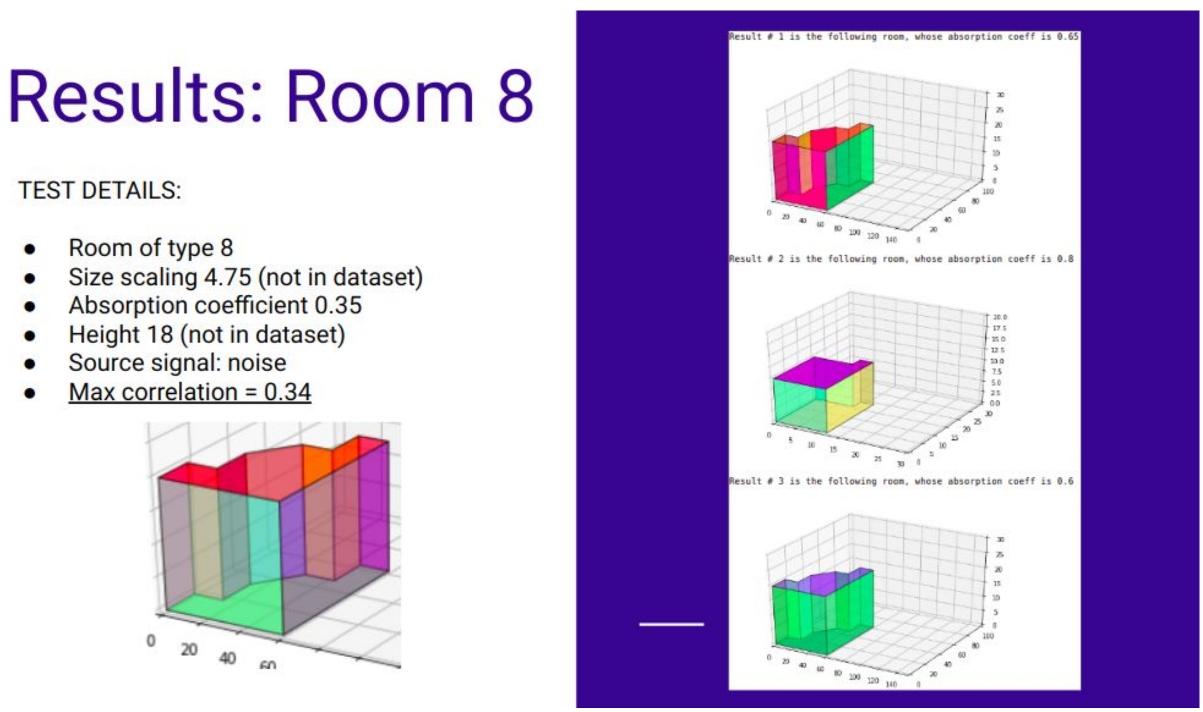


Fig 25: Top 3 rooms similar to a room of shape Room 8

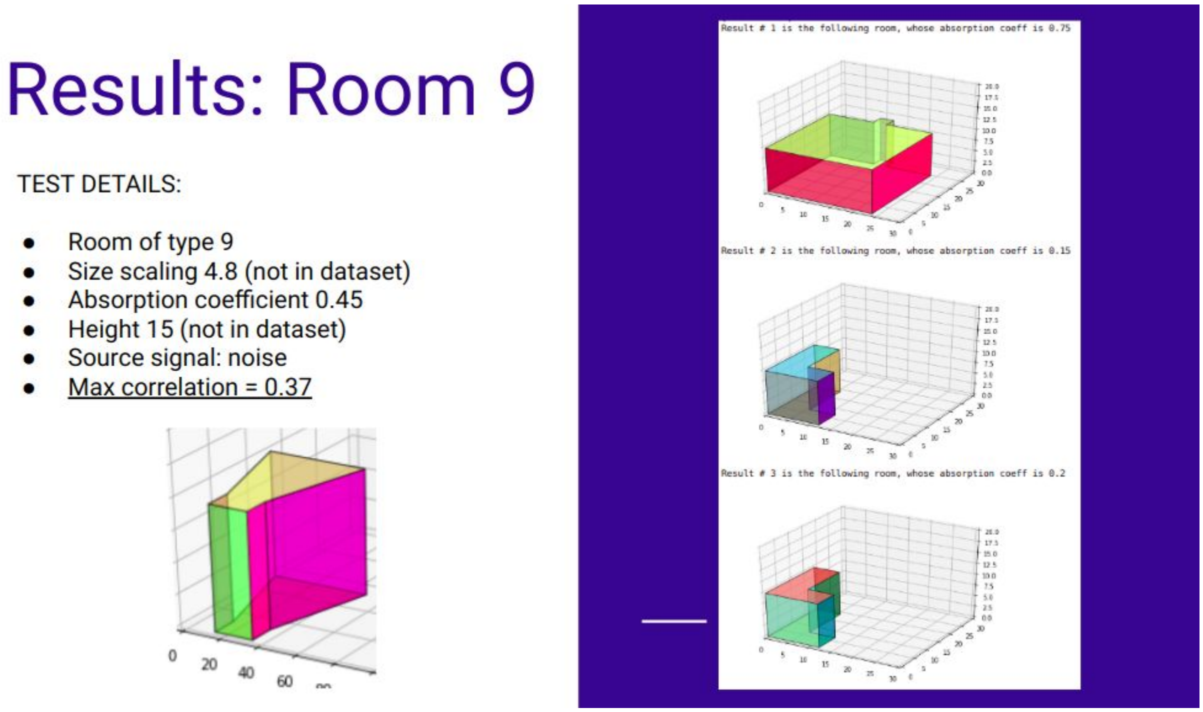


Fig 26: Top 3 rooms similar to a room of shape Room 9

INFERENCES

We see that for some shapes, such as 1, 2, 3, and 7, the returned ‘similar rooms’ have the same shapes as the query room. For 2, 3, and 7, this may be because of the unique shape of the room. For room 4, we could conclude that the enclosure for the door may not result in a distinctive impulse response since all its results have different shapes. As for rooms 5 and 6, they are confused with one another - however, the maximum correlation is about 0.33, which may not be very reliable. As for room 9, the result is surprising because it implies that an auditorium with a unique shape and position of source and mic sounds similar to a regular-sized bedroom/living room with much smaller dimension and varying absorption coefficient. This gives reason to believe that rooms which differ in every aspect, shape, size, and absorptivity, may still sound similar to each other.

OBSERVED PATTERNS

We can observe the change in the energy contained in frequency bins when all parameters are kept the same except for one. The following figure shows the variation in energy distribution for room 1 (standard cuboid) and a fixed absorption of 0.4 for sizes 1, 3, and 5.

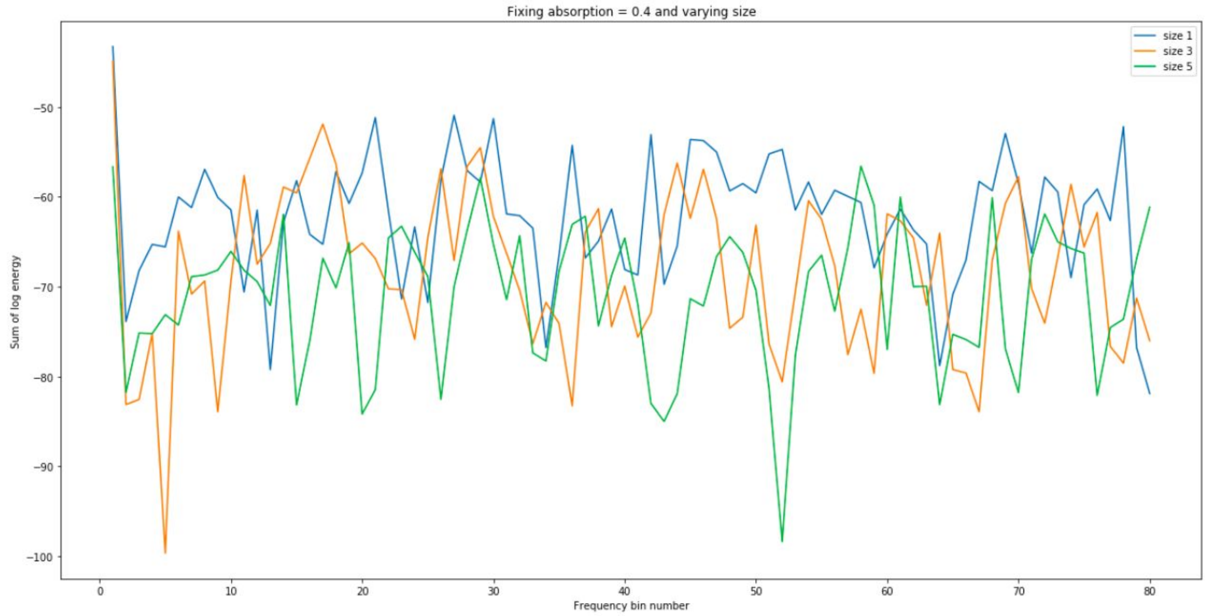


Fig 27: Log energy in freq bins for a fixed absorption coefficient and varying size of room

We see that the overall distribution is roughly the same, except that some peaks become valleys, i.e., changing the size might lead to the room amplifying some frequency ranges much lesser. Also, the overall energy seems to decrease with the size of the room. A similar pattern is noticed for fixed size and increasing absorption as well. Here, lower energy for increasing absorptivity is expected and intuitive.

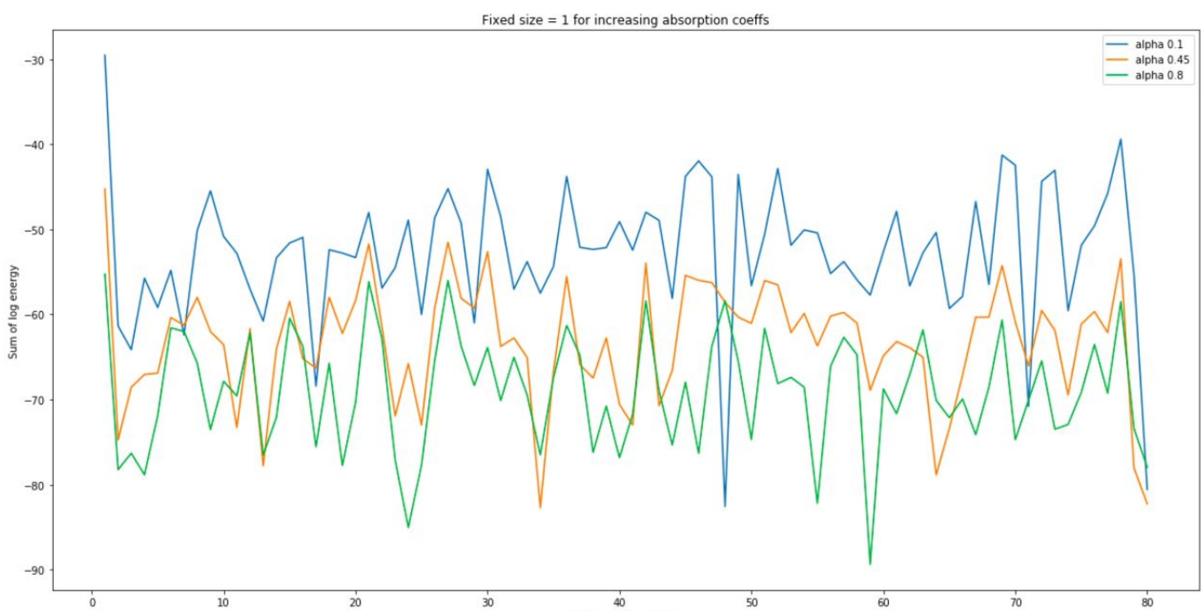


Fig 28: Log energy in freq bins for a fixed room size and varying absorption coefficient

FURTHER EXTENSION

1. Pyroomacoustics currently does not have provisions for varying height, which is common in auditoriums. Also, the surfaces specified by the corners are always assumed to be straight. The following figure shows other commonly seen auditorium/lecture hall designs which could be interesting to evaluate for similarity.

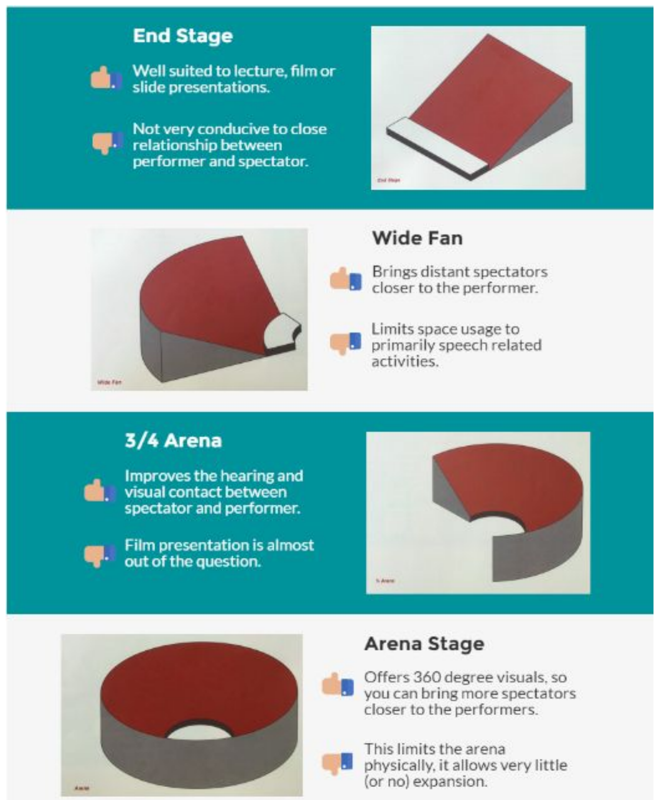


Fig 29: Auditorium designs with curved walls and varying height

2. Using filterbanks other than evenly spaced bins - such as mel, Bark, or gammatone filterbanks, which are more suited to human perceived hearing.

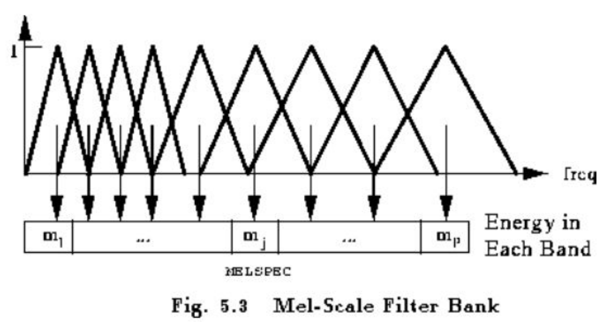


Fig 30: mel-scale filter bank with logarithmically-scaled frequency bins

3. Measuring similarity in the time-domain by using cross-correlation of the room impulse responses and picking the 'top 3 rooms' as those which have the highest peak values of the resulting signal.
4. Convolve the similar rooms' impulse responses with speech and guitar at close range and calculate T_{60}, D_{50}, and C_{80}. According to the three parameters, suggest the best potential application of the room.

Recording Need	Reverb Time (seconds)
Broadcast	Low (below 0.8)
Speech / Vocals	Low (0.7-1.1)
Live instruments	Medium (1-1.4)
Symphony / Drama	High (1.4-2)

Fig 31: Room purpose according to T_{60}

5. Listening test: Conduct a listening test according to MUSHRA, with 5 recordings: (i) the shortlisted top 3 rooms (ii) the original room i.e. the reference (iii) the room least similar to the query i.e. the worst sample. The subjects can be asked, "How similar do they sound?" on a scale of 1 - 5 (1 = not similar at all, 5 = almost exactly the same)
6. Since the approximate surface area and volume have been calculated for each room, they could be used along with the absorption coefficient and frequency response as features to build a supervised machine learning model which, given a desired impulse response, can predict room parameters.

REFERENCES

1. R. Scheibler, E. Bezzam, I. Dokmanić, *Pyroomacoustics: A Python package for audio room simulations and array processing algorithms*, Proc. IEEE ICASSP, Calgary, CA, 2018.
2. *https://blog.capterra.com/9-auditorium-plan-templates-to-inspire-your-next-project/*
3. *https://www.dimensions.guide/collection/theater-auditorium-layouts*
4. *http://www.deglerwhiting.com/auditorium-seating-layout-dimensions-the-complete-guide/*
5. *https://labrosa.ee.columbia.edu/doc/HTKBook21/node54.html*
6. *https://www.soundassured.com/blogs/blog/reverberation-and-its-application-in-recording-studios*
7. Series, B. "Method for the subjective assessment of intermediate quality level of audio systems." *International Telecommunication Union Radiocommunication Assembly* (2014).