Janet Li
Group 14

Performance Visualizer

Result

Notes
- Sentiment
- Quantitative characteristics

EMOPIA
- Valence
- Arousal

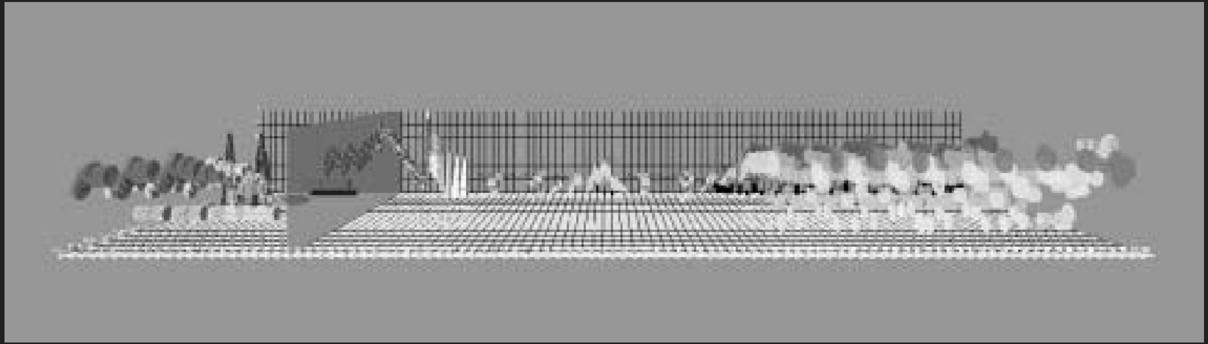
Shape
- Size
- Speed
- Color
- Shape
- How often a shape is generated

Original concept: Visual overlay on performing musician based on sentiment and quantitative analysis

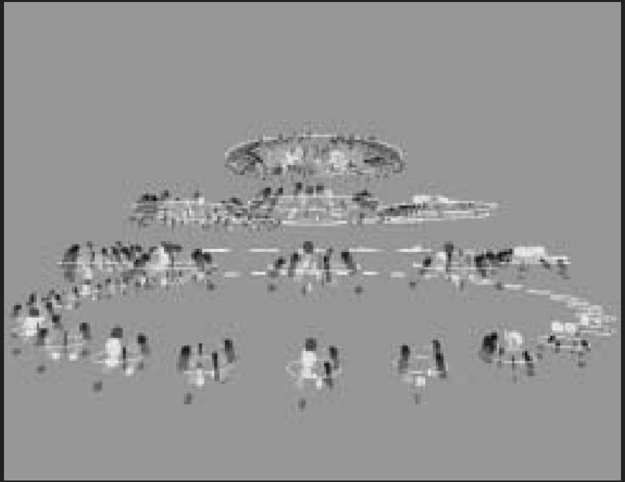
Mappings

Input: music	Output: shapes
Amplitude	Size of shape
Note duration/inter-onset interval	Length of shape (more or less continuous)
Sentiment: Positive	Warmer, brighter colors
Sentiment: Negative	Cooler, darker colors
Time since note onset	Distance from performer

Current* work in performance visualization



Comprehensive MIDI Player-Interactive virtual space; visualizes multiple channel layers of Valse des Fleurs by Tchaikovsky



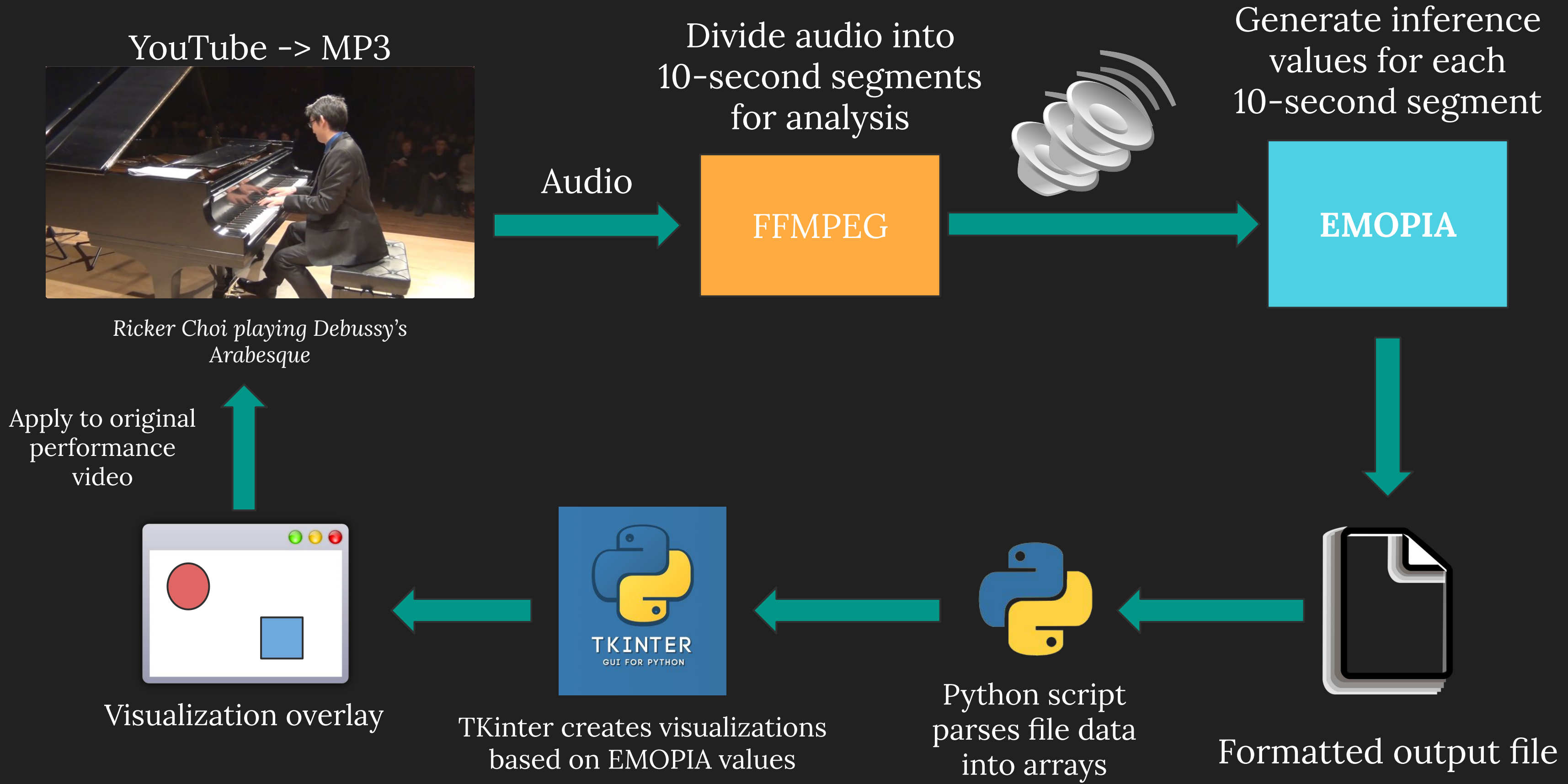
Visualizes musical structure of recursive hierarchy

Rumi Hiraga, Reiko Mizaki, and Issei Fujishiro. 2002. Performance visualization: a new challenge to music through visualization. In Proceedings of the tenth ACM international conference on Multimedia (MULTIMEDIA '02). Association for Computing Machinery, New York, NY, USA, 239–242. https://doi.org/10.1145/641007.641054

Motivation

- Classical music is often hard to approach as a listener
  - Particularly for kids/young children
- Provide a simple visualizer for recordings of classical performances based on the notes currently being played
  - Specifically, use a combination of sentiment and quantitative analysis on the notes being played to influence what the generated visuals look like

Technique


YouTube -> MP3
Ricker Choi playing Debussy's Arabesque
Audio
Divide audio into 10-second segments for analysis
FFMPEG
Generate inference values for each 10-second segment
EMOPIA
Formatted output file
Python script parses file data into arrays
TKINTER
GUI FOR PYTHON
TKinter creates visualizations based on EMOPIA values
Visualization overlay
Apply to original performance video

Process: EMOPIA output

Inferences for every 10 seconds of Ricker Choi playing Debussy's Arabesque No. 1

	Low Valence	High Valence
High Arousal	Q2	Q1
Low Arousal	Q3	Q4

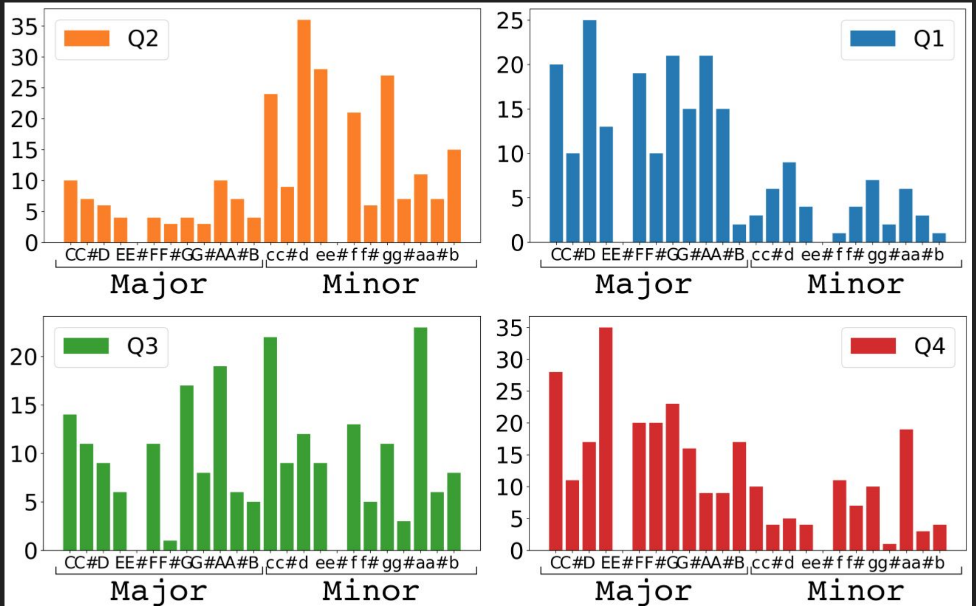
outputs

```
abs_0.wav  is emotion Q4, Inference values:  [0.05236754 0.03710493 0.05777654 0.9899472 ]]
abs_10.wav  is emotion Q4, Inference values:  [0.02216598 0.0195886  0.02014354 0.99775684]]
abs_20.wav  is emotion Q4, Inference values:  [0.01405534 0.03163977 0.02138974 0.99696064]]
abs_30.wav  is emotion Q1, Inference values:  [0.6617407  0.01601456 0.01004033 0.50370824]]
abs_40.wav  is emotion Q3, Inference values:  [0.25627008 0.39471266 0.65608525 0.3135395 ]]
abs_50.wav  is emotion Q4, Inference values:  [0.18802476 0.2107108  0.0188778  0.96072435]]
abs_60.wav  is emotion Q4, Inference values:  [0.03207375 0.07148175 0.0592729  0.9922824 ]]
abs_70.wav  is emotion Q4, Inference values:  [0.18821049 0.04989502 0.28378034 0.9648091 ]]
abs_80.wav  is emotion Q4, Inference values:  [0.46439102 0.32005224 0.02169288 0.61169475]]
abs_90.wav  is emotion Q4, Inference values:  [0.0207687  0.0260953  0.51218325 0.60367954]]
abs_100.wav  is emotion Q4, Inference values:  [0.02896808 0.0095521  0.02661344 0.9969424 ]]
abs_110.wav  is emotion Q4, Inference values:  [0.43077004 0.09250324 0.05918495 0.85090196]]
abs_120.wav  is emotion Q1, Inference values:  [0.52503    0.01079857 0.01055894 0.50426686]]
abs_130.wav  is emotion Q4, Inference values:  [0.4978638  0.03958806 0.16938588 0.5492348 ]]
abs_140.wav  is emotion Q4, Inference values:  [0.0119505  0.01146556 0.49717066 0.7328177 ]]
abs_150.wav  is emotion Q4, Inference values:  [0.08625619 0.08709763 0.19441879 0.9777598 ]]
abs_160.wav  is emotion Q2, Inference values:  [0.6067872  0.6470002  0.01204914 0.04304875]]
abs_170.wav  is emotion Q3, Inference values:  [0.02242045 0.02241473 0.9891635  0.11356956]]
abs_180.wav  is emotion Q3, Inference values:  [0.01853796 0.02256764 0.63111085 0.5197379 ]]
abs_190.wav  is emotion Q4, Inference values:  [0.02922424 0.01407809 0.01523031 0.99783885]]
abs_200.wav  is emotion Q4, Inference values:  [0.01203746 0.05413383 0.04454004 0.9952735 ]]
abs_210.wav  is emotion Q1, Inference values:  [0.99676216 0.02690898 0.00879209 0.04750074]]
abs_220.wav  is emotion Q1, Inference values:  [0.7161766  0.5723025  0.0310793  0.32788372]]
abs_230.wav  is emotion Q4, Inference values:  [0.15028392 0.01569371 0.02630714 0.99106795]]
abs_240.wav  is emotion Q4, Inference values:  [0.01715324 0.0232556  0.02378375 0.997772  ]]
abs_250.wav  is emotion Q4, Inference values:  [0.15640613 0.03024157 0.16956954 0.949735  ]]
```

Process: Mapping valence/arousal values to visuals

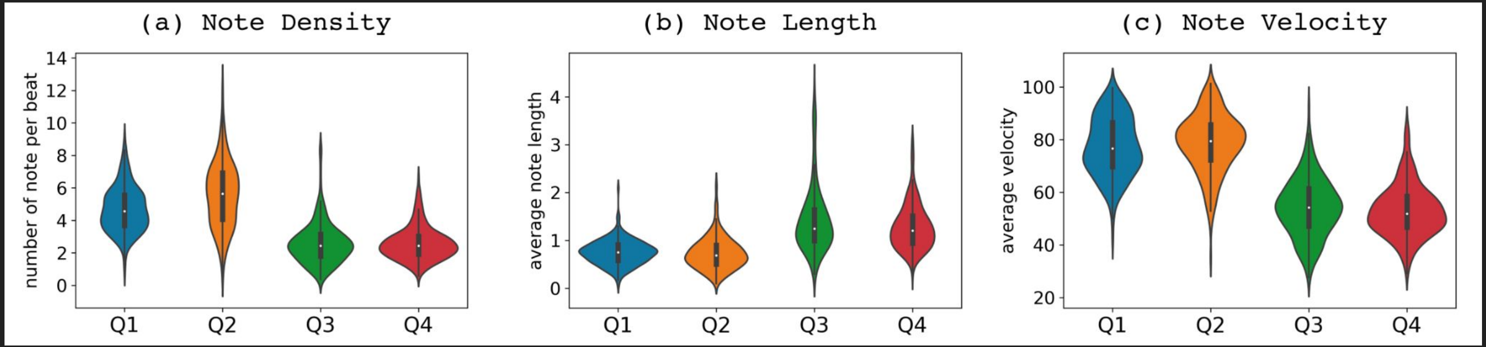
Valence: is the audio positive or negative?

- High Valence -> Major tonalities -> Positive sentiment -> Warm colors
- Low Valence -> Minor tonalities -> Negative sentiment -> Cool colors



Process: Mapping valence/arousal values to visuals

Arousal: "...approximately maps to frequency of note occurrences and their strength...we measure them by note density, length, and velocity"



Process: Mapping valence/arousal values to visuals

Arousal: "...approximately maps to frequency of note occurrences and their strength...we measure them by note density, length, and velocity"

- High Arousal: high note density, short note lengths, high note velocities
  - High note density -> Shapes come out more often
  - Short note lengths -> Shapes move away very fast from the performer
  - High note velocities -> Shapes are larger
  - Generally higher arousal -> Shape is pointier
- Low Arousal: low note density, longer note lengths, smaller note velocities
  - Low note density -> Shapes come out less often
  - Longer note lengths -> Shapes are slower to move away from the performer
  - Smaller note velocities -> Shapes are smaller
  - Generally lower arousal -> Shape is rounder

Process: Mapping valence/arousal values to visuals

	Quadrant	Q1: High Valence, High Arousal	Q2: Low Valence, High Arousal	Q3: Low Valence, Low Arousal	Q4: High Valence, Low Arousal
Shape Characteristics					
Size		Larger	Larger	Smaller	Smaller
Speed		Faster	Faster	Slower	Slower
Color		Warmer	Cooler	Cooler	Warmer
How often shapes are generated		More often	More often	Less often	Less often
What kind of shape?		Pointier (line or rectangle)	Pointier (line or rectangle)	Rounder (circle)	Rounder (circle)

Process: Mapping valence/arousal values to visuals

```python
# Q1: High Valence, High Arousal
# Shapes should be BIGGER, FASTER, WARM-COLORED and should come out MORE OFTEN; shapes should be POINTIER
if Q_array[data.infIndex] == 'Q1':
    base_radius = base_radius + 20*quad
    base_speed = base_speed + 20*quad
    base_color = color_gen("warm", quad)
    data.circleTime = round((5 + round(10*quad)) / 2)
    base_shape = random.choice([MyLine, MyRectangle])
# Q2: Low Valence, High Arousal
# Shapes should be BIGGER, FASTER, COOL-COLORED and should come out MORE OFTEN; [illegible]ould be POINTIER
elif Q_array[data.infIndex] == 'Q2':
    base_radius = base_radius + 20*quad
    base_speed = base_speed + 20*quad
    base_color = color_gen("cool", quad)
    data.circleTime = round((5 + round(10*quad)) / 2)
    base_shape = random.choice([MyLine, MyRectangle])
# Q3: Low Valence, Low Arousal
# Shapes should be SMALLER, SLOWER, COOL-COLORED and should come out LESS OFTEN; shapes should be ROUNDER
elif Q_array[data.infIndex] == 'Q3':
    base_radius = base_radius - 20*quad
    base_speed = base_speed - 20*quad
    base_color = color_gen("cool", quad)
    data.circleTime = round((15 + round(10*quad)) / 2)
    base_shape = Circle
# Q4: High Valence, Low Arousal
# Shapes should be SMALLER, SLOWER, WARM-COLORED and should come out LESS OFTEN; shapes should be ROUNDER
elif Q_array[data.infIndex] == 'Q4':
    base_radius = base_radius - 20*quad
    base_speed = base_speed - 20*quad
    base_color = color_gen("warm", quad)
    data.circleTime = round((15 + round(10*quad)) / 2)
    base_shape = Circle
```

Shape characteristics are NOT binary! The inference value acts as a scaling factor to determine *how* fast, *how* big, *how* warm/cool, etc. of the shape

Process: Mapping valence/arousal values to visuals

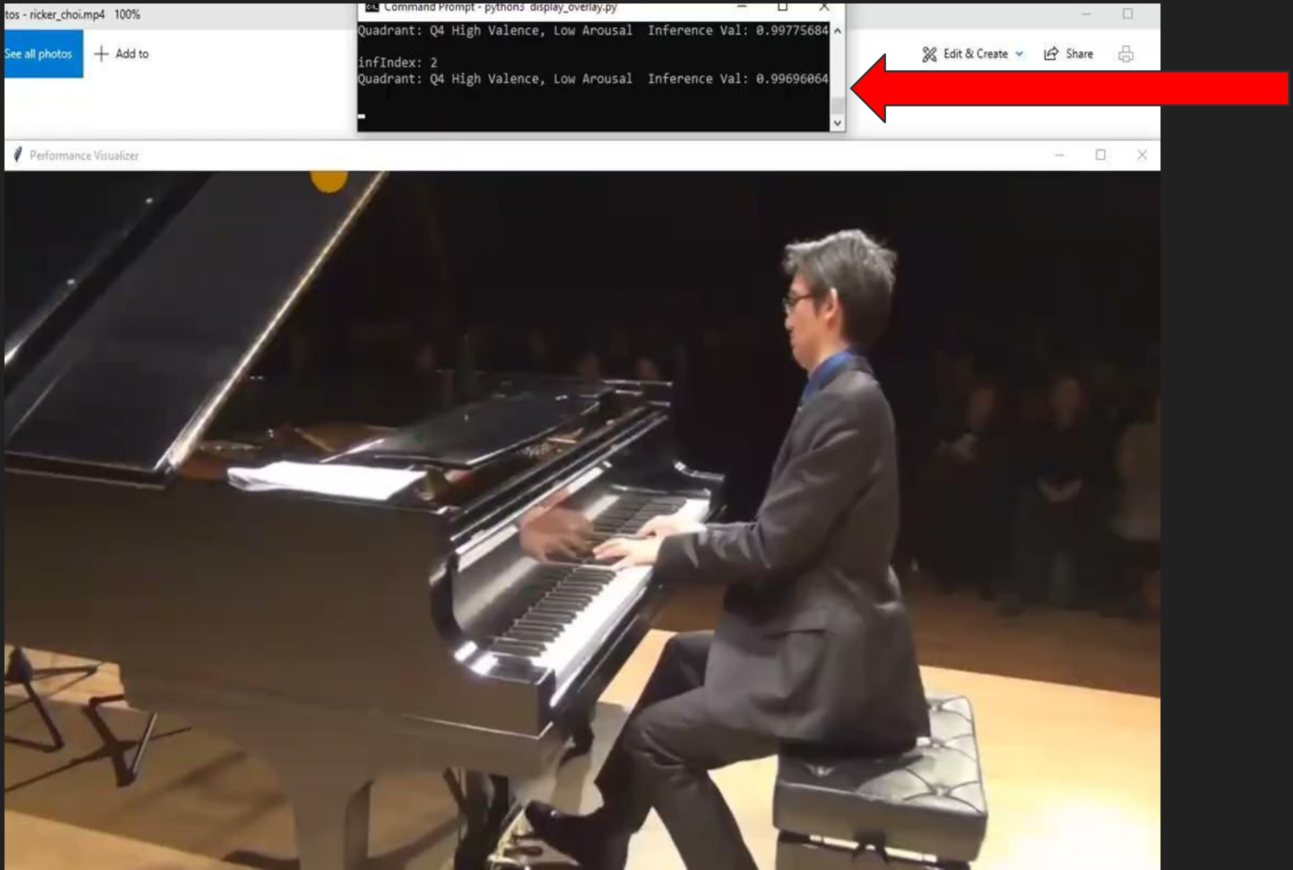
Color

```python
# Helper function: given a mode and a scaling number, generates a hex color for the shape
def color_gen(mode, scale):
    if mode == "warm":
        r_lim = round(255*scale)
        b_lim = 255 - round(255*scale)
        g_lim = round((r_lim + b_lim) / 2)
        red = lambda: random.randint(0,r_lim)
        green = lambda: random.randint(0,g_lim)
        blue = lambda: random.randint(0,b_lim)
        return '#%02X%02X%02X' % (red(),green(),blue())
    elif mode == "cool":
        r_lim = 255 - round(255*scale)
        b_lim = round(255*scale)
        g_lim = round((r_lim + b_lim) / 2)
        red = lambda: random.randint(0,r_lim)
        green = lambda: random.randint(0,g_lim)
        blue = lambda: random.randint(0,b_lim)
        return '#%02X%02X%02X' % (red(),green(),blue())
```

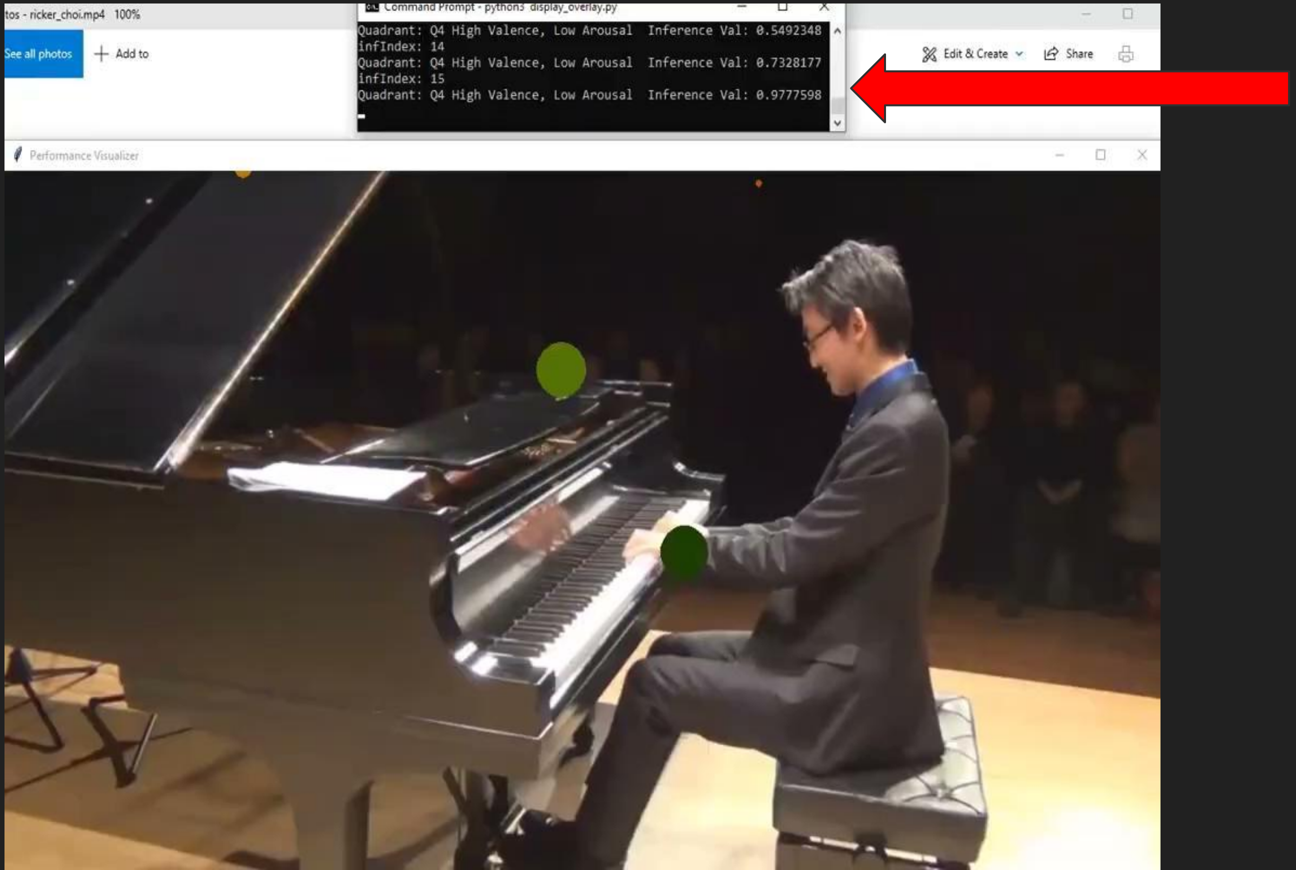
Scale positively affects red value and negatively affects blue value

Scale negatively affects red value and positively affects blue value

Process: Mapping valence/arousal values to visuals



Process: Mapping valence/arousal values to visuals



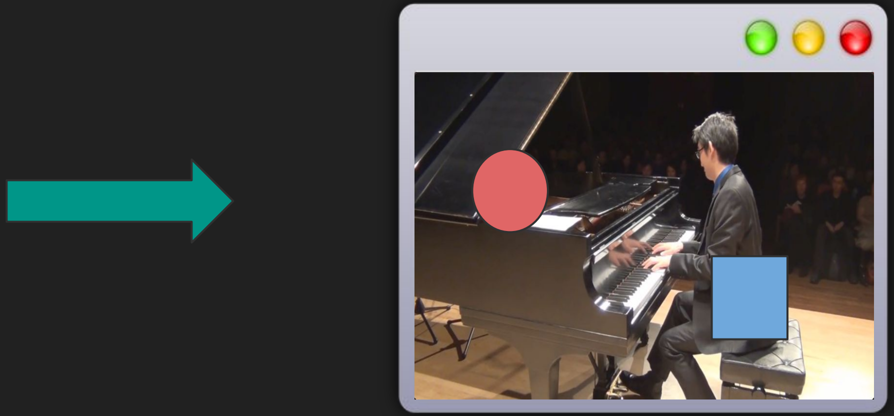
Process/Reflection: TKinter s u c k s

- Method to make the window transparent only exists for Windows, not macOS
  - Had to use CMU's virtual machines
- Very old and bad documentation
- Very limited functionality for common-sense things
  - You can play a video in Tkinter with external libraries
  - But you can't use this video as a background or a frame
  - And you can't draw shapes directly on top of this video

Normal video playing

Process/Reflection: TKinter s u c k s

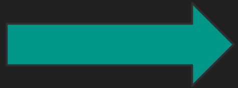
- Method to make the window transparent only exists for Windows, not macOS
  - Had to use CMU's virtual machines
- Very old and bad documentation
- Very limited functionality for common-sense things
  - You can play a video in Tkinter with external libraries
  - But you can't use this video as a background or a frame
  - And you can't draw shapes directly on top of this video

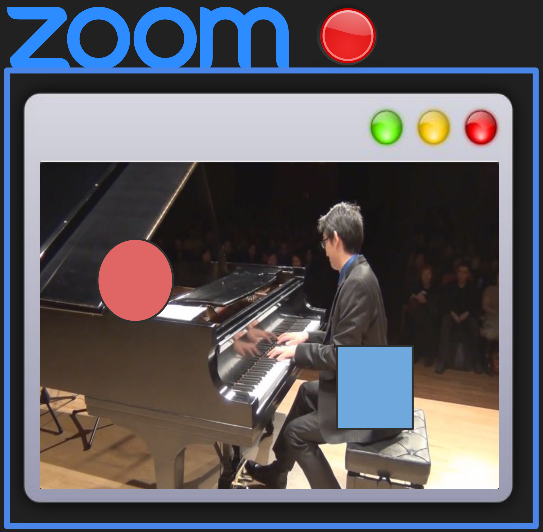


Place transparent TKinter window on top of original video (this only works on Windows OS)

Process/Reflection: TKinter s u c k s

- Method to make the window transparent only exists for Windows, not macOS
  - Had to use CMU's virtual machines
- Very old and bad documentation
- Very limited functionality for common-sense things
  - You can play a video in Tkinter with external libraries
  - But you can't use this video as a background or a frame
  - And you can't draw shapes directly on top of this video





Record this with Zoom

Process/Reflection: TKinter s u c k s

- Method to make the window transparent only exists for Windows, not macOS
  - Had to use CMU's virtual machines
- Very old and bad documentation
- Very limited functionality for common-sense things
  - You can play a video in Tkinter with external libraries
  - But you can't use this video as a background or a frame
  - And you can't draw shapes directly on top of this video

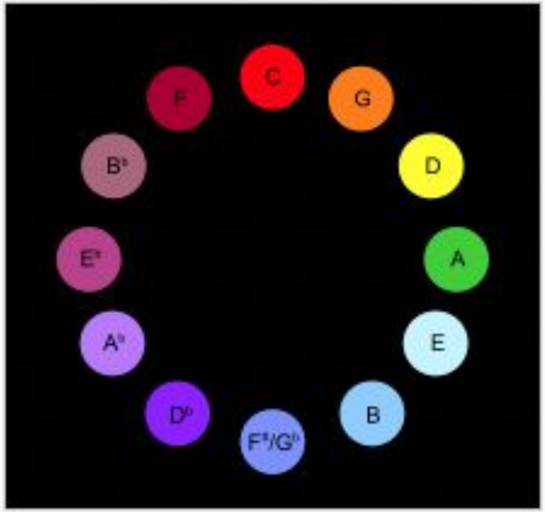
...all on the CMU virtual andrew machine

Reflection

- Non-cross-platform software is really bad!
- Old software is really bad!
- How objective are these visualizations? For example, there could be people who associate low valence/minor tonalities with *warm* colors rather than cooler colors
    - Should these visualizations be customizable, then?
    - Do these visualizations enforce a specific type of viewing specific types of music?
        - *Like mild propaganda*
- Does it make sense to use randomized values to influence the shapes generated as well? Or should all values *only* be determined by analyzed data?
- **How does this compare to chromesthesia?***

Reflection

How does this compare to chromesthesia?



Scriabin's sound-to-color associations arranged into a circle of fifths, demonstrating its spectral quality

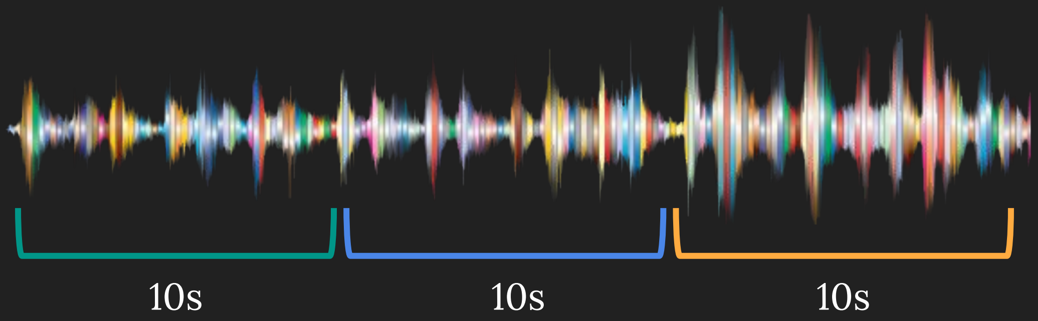
Future Work & Improvements

- All-in-one pipeline to do this with any uploaded video
- More direct analysis functions rather than all-encompassing valence/arousal values
- Should inference values from non-dominant quadrants also influence the visuals?
- **Sliding windows***
- Use better/more modern interfaces than Tkinter to directly show visualizations on top of input performance video
  - OpenCV or even PyGame are alternatives
- Improved visualizations
  - For example: growing/shrinking shapes rather than static ones
  - Customizable visualizations? Users could input what mappings they prefer
- **Real time? Bit tricky, but technically possible!***
  - Requires lots of computer science techniques that I am bad at
  - What implications does this have?
    - Use in performance AR/VR
- Do this with other instruments besides just piano pieces (EMOPIA is tailored for piano pop music specifically)
- Use Detectron to actually detect where the performer is rather than just making shapes come out from near the center
- Brightness of colors?
- More advanced graphics manipulation than just object generation (for example, modifying the color scheme of the entire performance frame)

Future Work & Improvements

Sliding windows

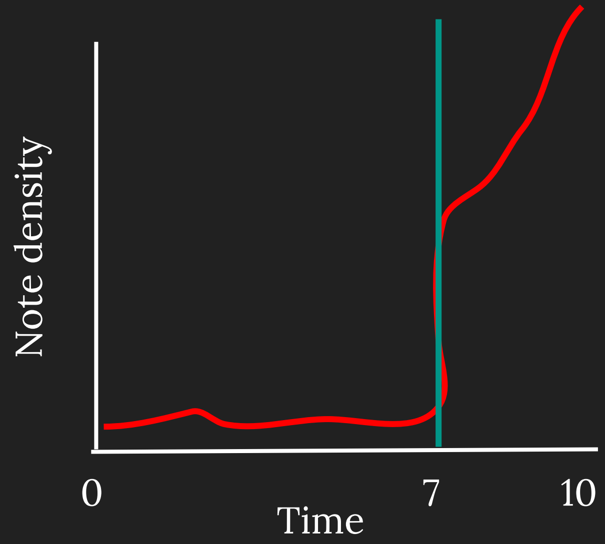
- Currently, within a 10-second segment of the performance audio, drastic changes within a small part in the segment can wildly skew inference results



Future Work & Improvements

Sliding windows

- Currently, within a 10-second segment of the performance audio, drastic changes within a small part in the segment can wildly skew inference results



=

Low density -> Low arousal?

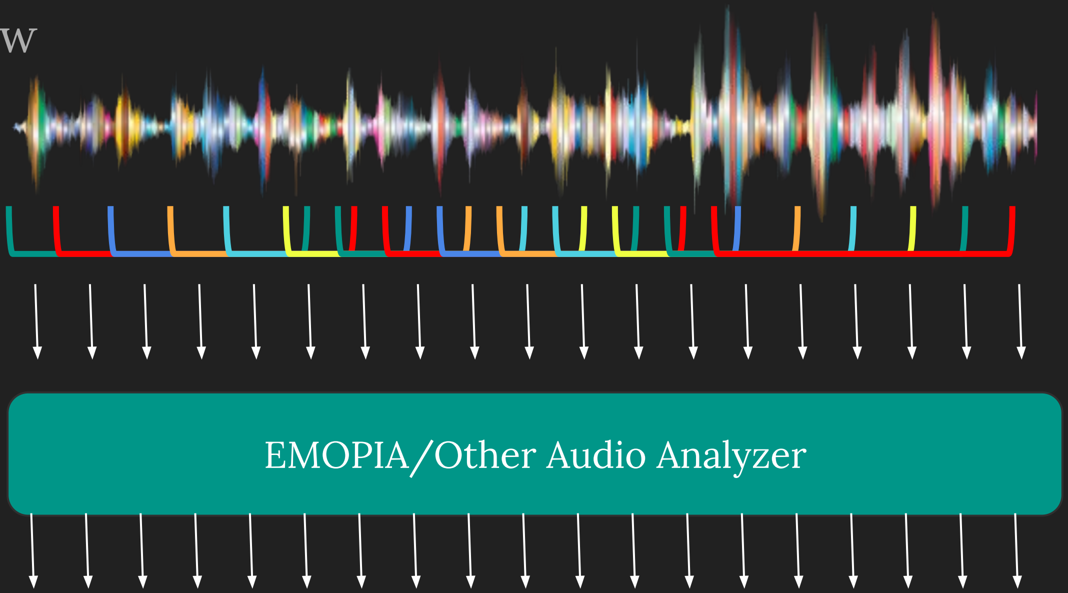
High density -> High arousal?

Low density, arousal from 0-7 seconds; high density, arousal from 7 to 10 seconds

Future Work & Improvements

Sliding windows

- A better solution would be to have overlapping windows that sample the original audio to be analyzed, where inference values are generated for each window



Future Work & Improvements

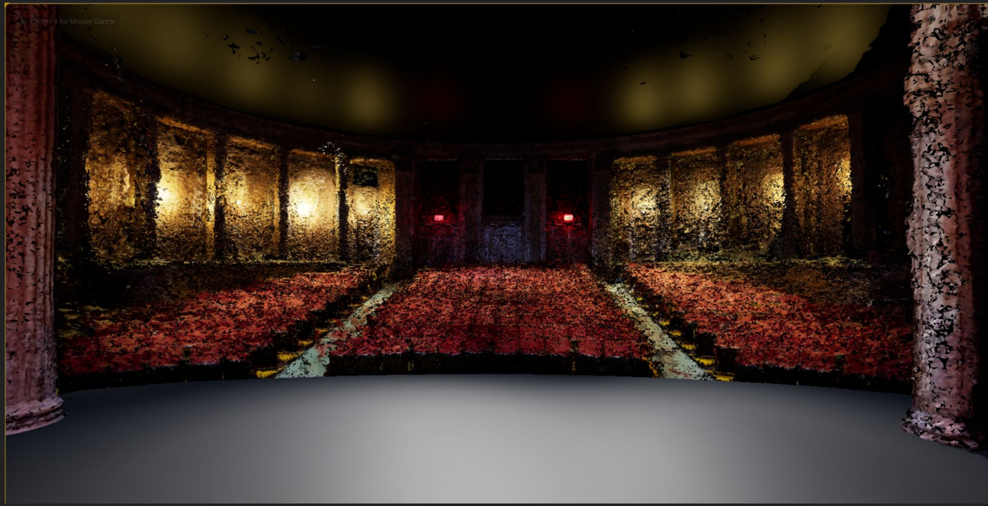
Sliding windows

- Already a very common technique in any audio/signal processing context
- Obviously more expensive computationally and storage-wise
- But would get you much more accurate data & visuals

Future Work & Improvements

Real time? Bit tricky, but technically possible!

- Requires lots of computer science techniques (dynamic programming) that I am bad at
- What implications does this have?
  - Use in performance AR/VR
    - For both the performer and listeners
    - 3D visual objects instead of 2D

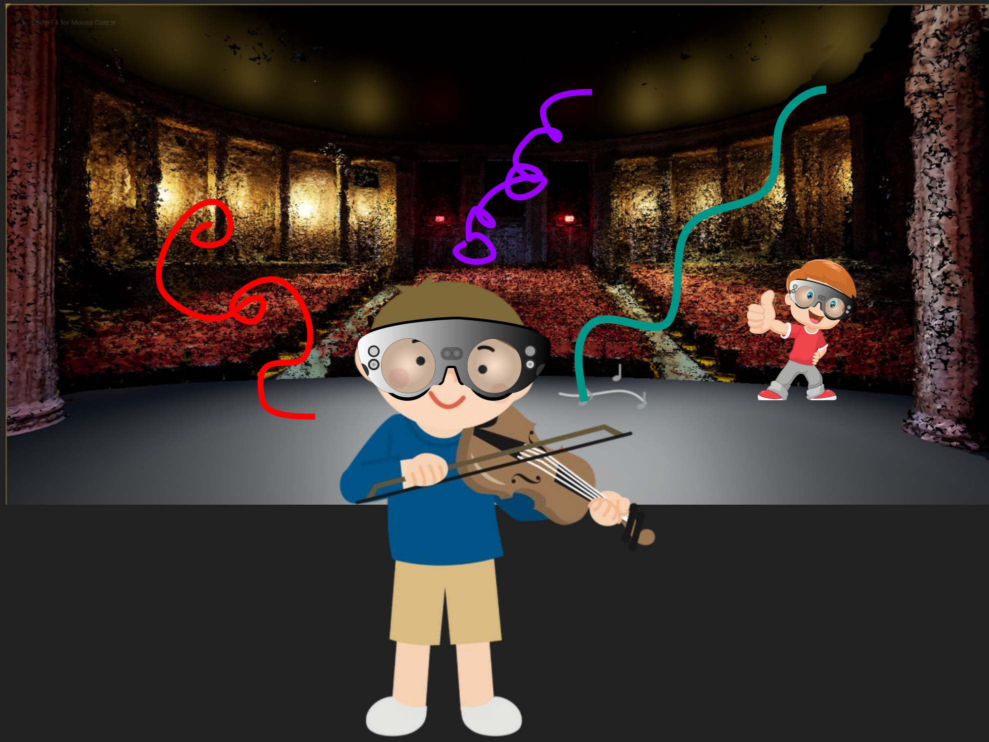


Kresge VR Project @ CMU

Future Work & Improvements

Real time? Bit tricky, but technically possible!

- Requires lots of computer science techniques (dynamic programming) that I am bad at
- What implications does this have?
  - Use in performance AR/VR
    - For both the performer and listeners
    - 3D visual objects instead of 2D



Thank you!