consonare

Automatic Overtone Analysis and Dissonance Profiling for Single-Note Recordings

Kritchanat Thanapiphatsiri

Department of Computer Engineering
Kasetsart University

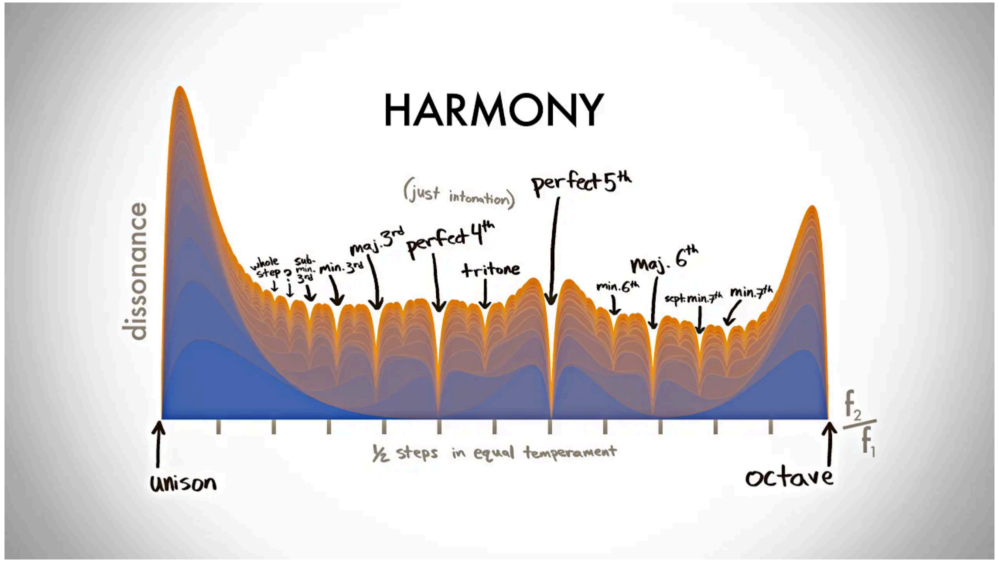
2025-10-06

Outline

1. Introduction

1.1 Background

This project is heavily inspired by a YouTube video:



minutephysics - The Physics Of Dissonance

https://youtu.be/tCsl6ZcY9ag

Why do we have music?

1.1 Background

Why do we have music? Think about it.

1.1 Background

Why do we have music? Think about it.

You might then answer:

“Well, because thousands of years ago, people began to realize that some sounds felt good together, and they never stopped exploring that.”

1.1 Background

Why do we have music? Think about it.

You might then answer:

“Well, because thousands of years ago, people began to realize that some sounds felt good together, and they never stopped exploring that.”

Okay, but then… why *do* some sounds “felt good” together? And what does that even mean?

1.1 Background

For those who've taken any kinds of music class, you sure would have learned the concept of "Intervals":

- Western music has given each "pair of sounds" usually from the same source a unique name, depending on how "far apart" they are.
- Common intervals include the *perfect octave*, the *perfect fifth*, the *perfect forth* and so on.
- Usually, it is simply taught that *"If a pair of sounds is an octave, a fifth, or a forth apart from each other, then they will* ***feel good*** *to be listened to together."*

1.1 Background

For those who've taken any kinds of music class, you sure would have learned the concept of "Intervals":

- Western music has given each "pair of sounds" usually from the same source a unique name, depending on how "far apart" they are.
- Common intervals include the *perfect octave*, the *perfect fifth*, the *perfect forth* and so on.
- Usually, it is simply taught that *"If a pair of sounds is an octave, a fifth, or a forth apart from each other, then they will* ***feel good*** *to be listened to together."*

But… why? and does it even applies to *all* sounds?

1.1 Background

A lot of people like to simply brush it off as:

“Oh, our ears simply like nice, round numbers of frequency ratios between a pair of sounds - like 2:1 for an octave, 3:2 for a perfect fifth, and so on…”.

Which raises even more question - why *do* our ears prefer nice ratios of frequencies?

1.2 Objectives

As it turns out, that explanation was **wrong**. There is a much more logical explanation to this phenomenon, which this project will cover. We will also go over how the western music concept of intervals only applies to *specific* instruments.

1.2 Objectives

As it turns out, that explanation was **wrong**. There is a much more logical explanation to this phenomenon, which this project will cover. We will also go over how the western music concept of intervals only applies to *specific* instruments.

This includes modelling human perception, analysing the unique characteristics that make a sound identifiable, and how that relates to the specific patterns of how “far apart” a pair of sounds should be to “feel good to listen to”. If that specific pattern doesn’t line up with western music intervals, then **the procedure should be able to precisely output the correct patterns themselves for any sound.**

1.3 Scope

This project will strictly be using a noiseless (or close to) mono WAV file to store **single note** recordings.

We will also only be **analysing one audio source at a time**. It will not analyse the intervals from two different instruments being played two different notes, for example.

2. Methods

2.1 Theoretical Background

Let’s start from the very basics.

$$y(t) = A\sin(\omega t + \varphi)$$

This is a general form of an arbitrary sinusoid.

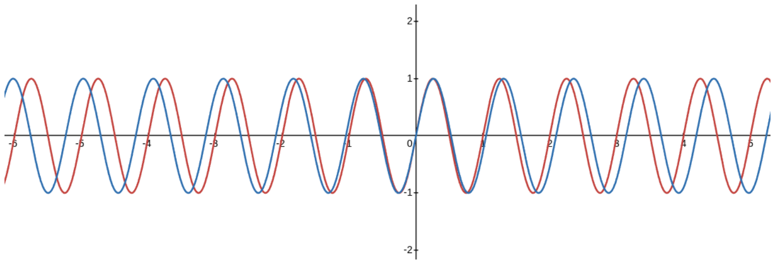
2.1 Theoretical Background

Let’s start from the very basics.

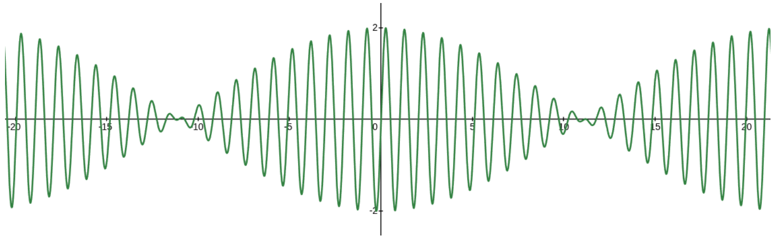
$$y(t) = A\sin(\omega t + \varphi)$$

This is a general form of an arbitrary sinusoid.

When two sinusoids of very similar frequencies are summed together, it will interfere with each other periodically. This is called **beating**, and it will beat at exactly $f_{\text{beat}} = f_1 - f_2$.



$$\overset{+}{\Rightarrow}$$



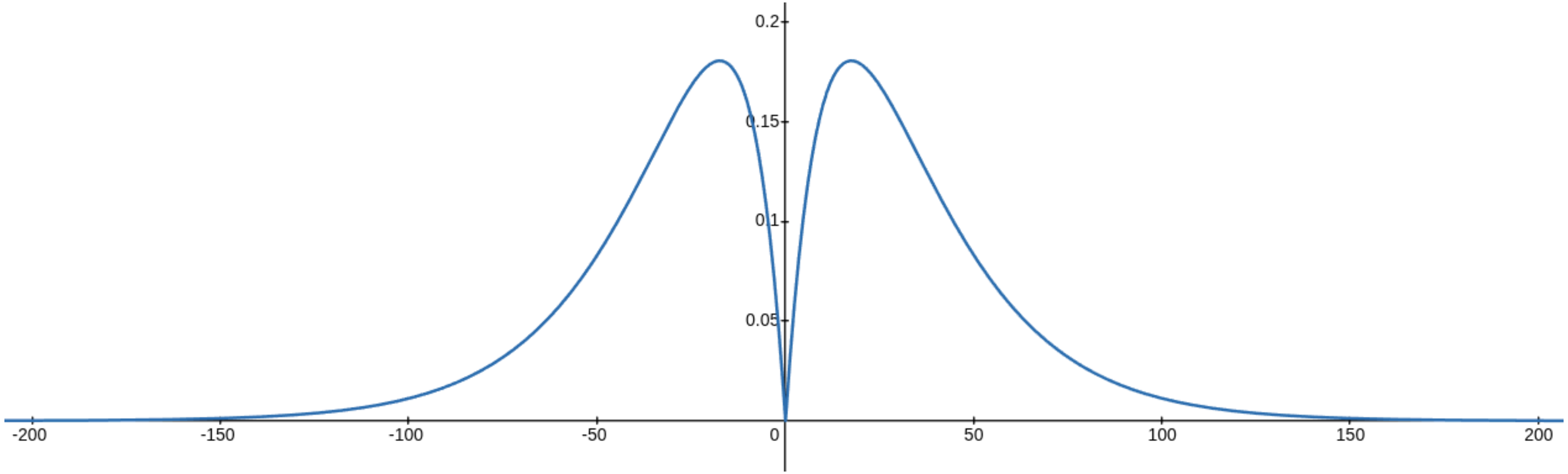
2.1 Theoretical Background

When f_{beat} is zero, there is no beating. When it's sufficiently low, the beating period is long enough to be heard. Our auditory system is extremely sensitive to the amplitude fluctuations, which we naturally interpret as unpleasant or danger. But as the frequency gets larger, we start to perceive the sound as continuous yet "rough".

2.1 Theoretical Background

When f_{beat} is zero, there is no beating. When it's sufficiently low, the beating period is long enough to be heard. Our auditory system is extremely sensitive to the amplitude fluctuations, which we naturally interpret as unpleasant or danger. But as the frequency gets larger, we start to perceive the sound as continuous yet "rough".

We can plot the roughness over Δf like so:



2.1 Theoretical Background

That specific graph shape describes exactly what we described earlier, but we can make it more nuanced. The ability of our auditory system to differentiate between two sinusoidals decline as the reference frequency goes up. This is called the **critical bandwidth** of that f_{ref}.

2.1 Theoretical Background

That specific graph shape describes exactly what we described earlier, but we can make it more nuanced. The ability of our auditory system to differentiate between two sinusoidals decline as the reference frequency goes up. This is called the **critical bandwidth** of that f_{ref}.

A simplified implementation of a CBW is the **equivalent rectangular bandwidth** function:

$$\text{ERB}(f) = 24.7\left(4.37\frac{f}{1000} + 1\right)$$

where it simply acts like a narrow band-pass filter tuned to a center frequency f.

2.1 Theoretical Background

Going back to the pairwise roughness function, a popular implementation is the **Sethares/Plomp–Levelt kernel**:

$$D(\Delta) = e^{-3.5\ \mathrm{ERB}(\Delta)|\Delta|} - e^{-5.75\ \mathrm{ERB}(\Delta)|\Delta|}$$

That covers the sinusoidals. But what about other audio signals?

2.1 Theoretical Background

Going back to the pairwise roughness function, a popular implementation is the **Sethares/Plomp–Levelt kernel**:

$$D(\Delta) = e^{-3.5\ \mathrm{ERB}(\Delta)|\Delta|} - e^{-5.75\ \mathrm{ERB}(\Delta)|\Delta|}$$

That covers the sinusoidals. But what about other audio signals? Well, recall the **Fourier series**:

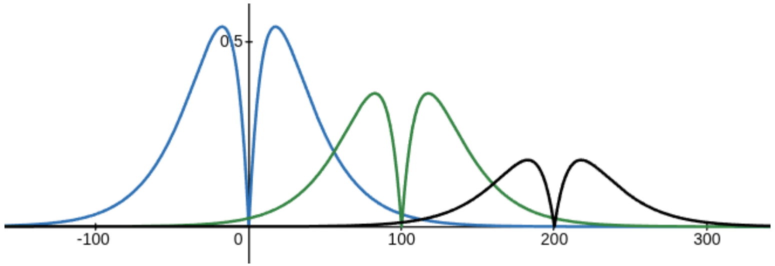
$$s_{N(x)} = \sum_{n=-N}^{N} c_n e^{i2\pi\frac{n}{P}x}$$

Sethares assumes that *total sensory roughness can be modelled as the sum of independent pairwise roughness*, so let's model this.

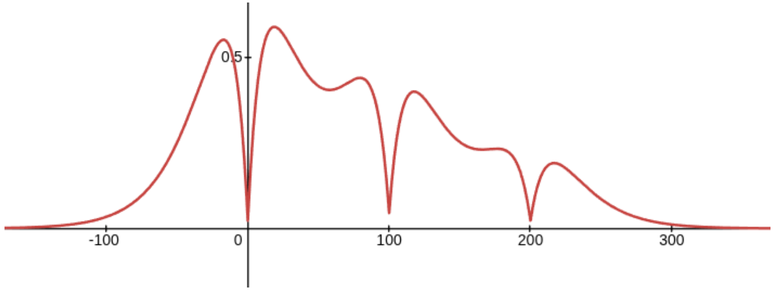
2.1 Theoretical Background

Since any audio signals comprise of sums of sinusoids, the dominant set being called their **partials** or **overtones**, we can linearly also *sum up the pairwise roughness for each sinusoid in the frequency domain, each frequently-shifted appropriately*. This total roughness is called **dissonance**, and the lack there of is called **consonance**.

$$D_{\text{total}} = \sum_{i<j} w_i w_j D\left(\left|f_i - f_j\right|\right)$$



$\overset{+}{\Rightarrow}$



2.1 Theoretical Background

To finish the model, we need to consider the weighting of each pairwise roughness. Since our auditory systems sense different perceived loudness on different frequencies, an equal-loudness weighting must be used. One widely used implementation is **A-weighting**:

$$R_A(f) = \frac{12194^2 f^4}{(f^2 + 20.6^2)\sqrt{(f^2 + 107.7^2)(f^2 + 737.9^2)}(f^2 + 12194^2)}$$

Where $A(f) \approx 20 \log_{10}(R_A(f)) + 2.00$.

2.1 Theoretical Background

Lastly, when two frequency components are close and one is much stronger, the weaker one may become inaudible because the stronger component masks nearby frequencies within a critical band. This is called **auditory masking**. Masking depends on both frequency distance and amplitude ratio.

And now, we have all of the theoretical background to model human hearing and dissonance perception.

2.2 Procedure

1. Normalise and trim silence from input file. Flag if not steady-pitch (via sliding window median f_0)
2. Band-pass at hearing range, median-filter to emphasise partials while STFT with Hann window & suppress silence via percentile noise floor estimation.
3. Compute f_0 candidates with YIN, autocorrelation and cepstral. Then pick by RANSAC over time frames.
4. Peak pick top N partials.
5. Try fitting the spectrum to a stiff string model.
6. Apply equal-loudness weighing via A-weighting and mask partials within the ERB.

2.2 Procedure

7. Place the weighted pairwise roughness curve at every partials and sum all of them up.
8. Find local minima and suggest simple fraction approximations.
9. Extend to 3 notes, also implement 7 and 8.
10. Flag if low confidence results, provide diagnostics about the procedure.
11. Provide overtone table, dissonance curve and tuning suggestions for musicians.

3. Experimental Results

3.1 Input and Output Data

30 1-second single note recordings were used as test data. They are mostly real recordings with some being synthetic for control. Here is a spectogram of a selected recording.

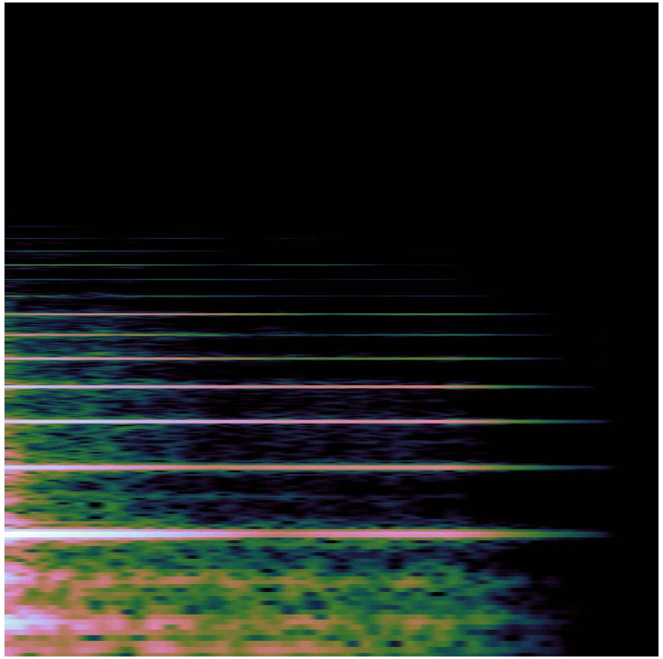


Figure 1: An E4 recording from an upright piano.

3.1 Input and Output Data

Output data should look like the following (shortened for presentation):

Summary Highlights:

- Recording: `keys_upright-piano_e4.wav` – median $f_0 \approx 330.9$ Hz (steady pitch)
- Model error ≈ 0.96 cents, $B = 5.28 \times 10^{-4} (R^2 = 0.955)$
- Stable partials = 20 (SNR ≥ 0 dB)
- f_0 estimators disagree ≈ 26 cents (HIGH)

Key Findings:

- Dissonance curve depth $\approx 0.95 \rightarrow$ good dynamic range
- Overtone peaks match theoretical harmonic ratios
- Best interval: **3 : 1 (perfect 12th)** at 1902 cents

3.1 Input and Output Data

- Suggest tuning upper note ≈ 0 cents sharp of 3:1 minimum

Diagnostics:

- Inharmonicity consistent with upright-piano profile
- 3-note dissonance surface confirms consonant alignment
- Minor pitch estimator disagreement flagged

Output Files:

- `dissonance_curve.csv`, `overtone_table.csv`

3.1 Input and Output Data

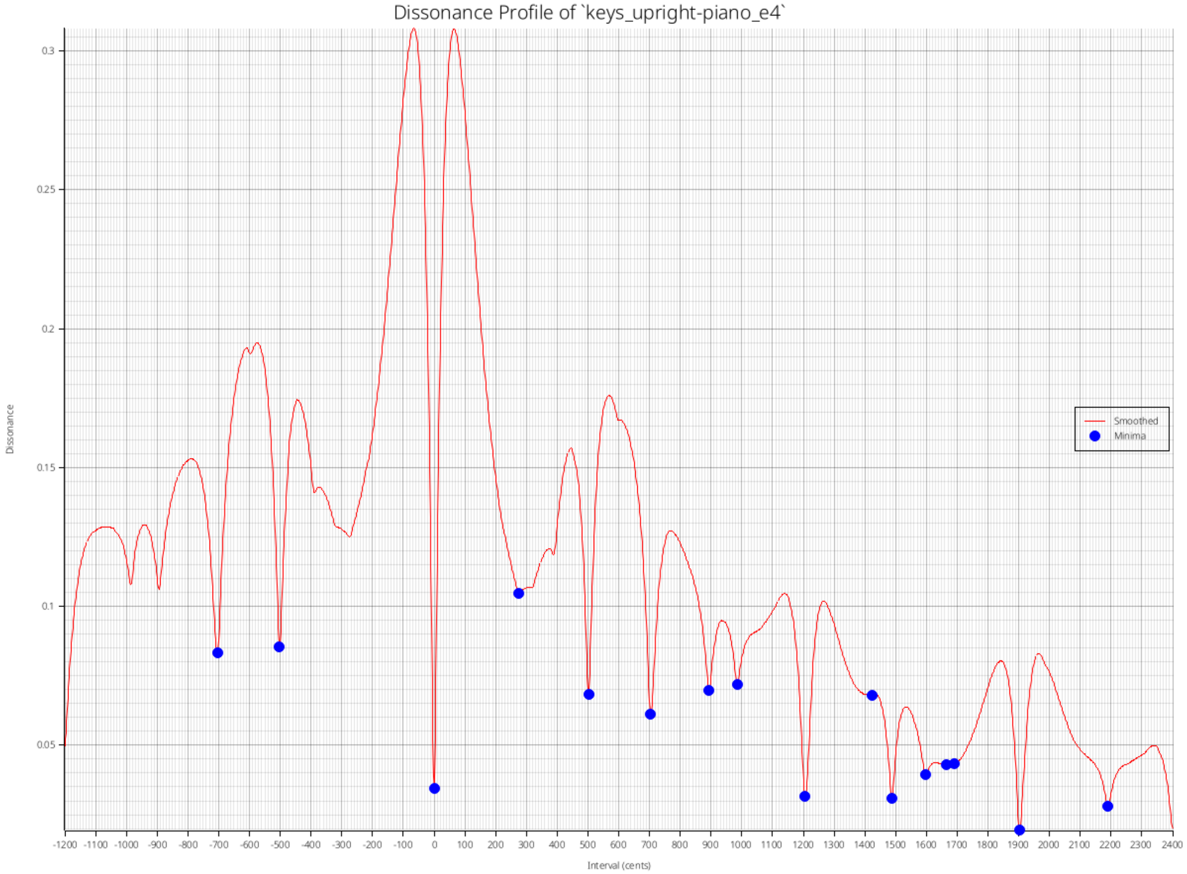


Figure 2: Dissonance Profile of recording

3.1 Input and Output Data

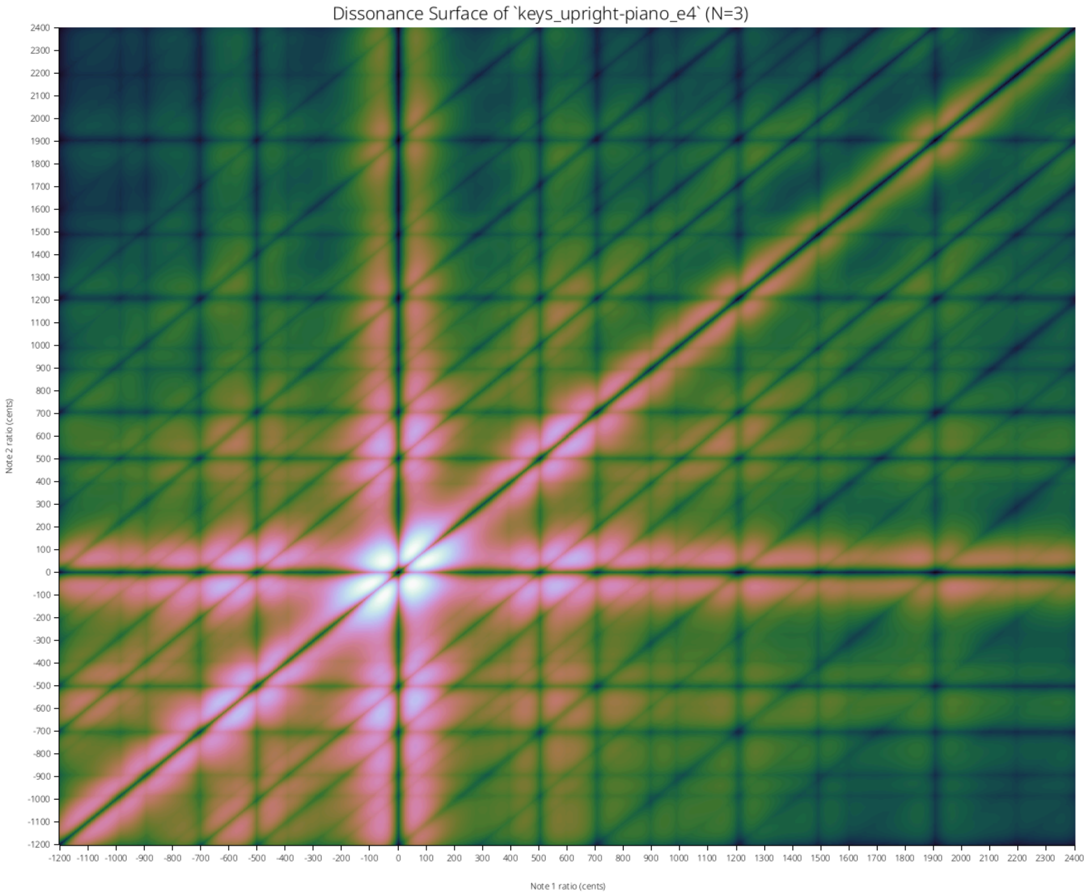


Figure 3: Dissonance Surface of recording ($N = 3$)

3.2 Evaluation of Method Performance

1. Fundamental Accuracy
   - Median error = 2.91 cents; 95th percentile = 7.996 cents → **Complies with $\leq$ 8 cent target.**

2. Partial Fidelity
   - Sawtooth & Square: freq err $\leq 0.005\%$, mag err ≤ 0.1 dB.
   - Sine: meets limits but uncertain tracking.

3. Inharmonicity Fit
   - Model fit $R^2 \geq 0.955$ (piano only); others correctly reject fit.

4. Dissonance vs Perception
   - Kendall $\tau = 0.56$, Spearman $\rho = 0.74$ → **Strong agreement with listener ratings.**

3.2 Evaluation of Method Performance

5. Minimum Localisation & Sharpness
   - All 30 samples within ±7 cents; **100 % compliance.**

6. Naming Accuracy
   - Harmonic: 100 %; Inharmonic: 44 % (< 65 % target) → **Partial compliance.**

7. Robustness to Pitch Drift
   - Consensus estimator handles vibrato / voice instability well.

8. Noise Robustness
   - Stable tracking down to 10 dB SNR; graceful degradation.

9. Ablation Performance

3.2 Evaluation of Method Performance

- Removing ERB or weighting → large degradation ($|g| > 0.7$); masking negligible → partial satisfaction.

10. Computational Efficiency
    - $\sim$ 50 ms (processing only) / 143 ms (with plotting) per 1 s audio.
11. Reproducibility
    - Fully deterministic; identical outputs & checksums across runs.
12. Usability & Reporting
    - Musician-friendly outputs (dissonance plots, tutorials); optional visualization & verbose modes.

4. Conclusion

4.1 Summary

Project achieved its objectives: accurately modeled human hearing and quantified perceived dissonance between two notes of the same timbre.

Results align with theoretical ground truths: recommended intervals are perceived as minimally dissonant (maximally consonant) by both the model and real listeners.

4.2 Limitations & Future Work

Some mathematically consonant intervals (e.g., 3:1 ratio) are less musically practical.

In 3-note dissonance modeling, trivial chords (e.g., unisons) dominate results, reducing variety in interesting chords.

Future improvements could refine selection to better balance mathematical accuracy and musical usefulness.

4.3 Resources

All code, papers, and materials available at:

https://github.com/krtchnt/consonare

Thank you!

Any questions?