# \*\*Documentation for Piano and Electronics Composition Project\*\*

## \*\*Overview\*\*

This project focuses on creating a timbre-based composition for piano and electronics, avoiding symbolic approaches. The workflow involves generating "unwanted" piano music, fragmenting it using content-aware methods, enriching it with sounds retrieved via the Freesound API, and developing spectral techniques to integrate piano materials with electronic textures.

---

## \*\*Workflow\*\*

### \*\*1. Generating the "Unwanted" Piano Piece\*\*

- \*\*Objective\*\*: Use AI to generate a cheesy, pop-style piano piece as raw material.

- \*\*Tool\*\*: Suno (or any AI music generator).

- \*\*Prompt\*\*:

```

[Bright Piano Melody], [Cheerful Pop Style], [Simple Chord Progressions], [120 BPM], [Feel-Good Vibes], [Catchy Repetitive Motifs]

```

- Keep the prompt simple and tag-based to ensure compatibility with Suno.

- Generate multiple versions if necessary to find the most "cheesy" result.

---

### \*\*2. Splitting the Piano Music into Fragments\*\*

- \*\*Objective\*\*: Segment the generated piano music into meaningful fragments using content-aware methods.

- \*\*Criteria for Splitting\*\*:

1. \*\*Onset Detection\*\*: Split at note or chord changes (e.g., using `librosa.onset.onset\_detect`).

2. \*\*Novelty Detection\*\*: Identify transitions between contrasting sections (e.g., via self-similarity matrices).

3. \*\*Spectral Features\*\*: Split based on timbral changes (e.g., spectral centroid or flatness).

4. \*\*Energy-Based Segmentation\*\*: Divide based on loudness levels (e.g., RMS energy).

5. \*\*Harmonic vs. Percussive Decomposition\*\*: Separate harmonic and percussive components (e.g., HPSS in `librosa`).

6. \*\*Silence Detection\*\*: Split at pauses or low-energy regions (e.g., `pydub.silence.detect\_nonsilent`).

7. \*\*Beat and Tempo Analysis\*\*: Align splits with rhythmic patterns or tempo changes.

- \*\*Tools\*\*:

- Python libraries: `Librosa`, `Essentia`, `madmom`, or `pydub`.

- Spectral analysis tools like Sonic Visualiser for manual refinement.

---

### \*\*3. Filtering the Fragments\*\*

- \*\*Objective\*\*: Reduce redundancy and select fragments that align with your aesthetic goals.

- \*\*Filtering Criteria\*\*:

1. \*\*Timbre-Based Filtering\*\*: Use MFCCs or spectral centroid to group fragments by timbre.

2. \*\*Spectral Complexity\*\*: Retain fragments with high or low harmonic richness.

3. \*\*Dynamic Range\*\*: Focus on fragments with strong dynamic contrasts.

4. \*\*Rhythmic Patterns\*\*: Select fragments with regular or irregular rhythmic structures.

5. \*\*Pitch Content\*\*: Choose fragments based on tonal stability, dissonance, or microtonality.

6. \*\*Novelty Scores\*\*: Retain unique fragments with high novelty values.

- \*\*Clustering for Filtering\*\*:

- Use clustering algorithms to organize fragments by feature similarity:

- K-Means for grouping by timbre or rhythm.

- DBSCAN for identifying outliers or unique fragments.

- Hierarchical clustering for nested relationships between fragments.

- Spectral clustering for subtle textural differences.

- \*\*Tools for Clustering\*\*:

- Python libraries: `sklearn.cluster` (K-Means, DBSCAN), `Librosa` (feature extraction), `matplotlib`/`seaborn` (visualization).

- Dimensionality reduction techniques like t-SNE or UMAP for visualizing clusters.

---

### \*\*4. Enriching Fragments Using Freesound API\*\*

- \*\*Objective\*\*: Retrieve similar sounds from Freesound.org to layer with your piano fragments.

- \*\*Steps\*\*:

- Query Freesound API using features extracted from each fragment:

- Spectral descriptors (e.g., pitch, timbre, spectral centroid).

- Tags related to emotional quality or texture (e.g., "bright," "metallic").

- Duration constraints to match fragment lengths.

- \*\*Tools\*\*:

- Freesound API documentation and Python SDK for querying sounds programmatically.

- \*\*Combining Sounds\*\*:

- Layer retrieved sounds with piano fragments using time-stretching, pitch-shifting, or spectral morphing techniques.

---

### \*\*5. Morphing Sounds\*\*

- \*\*Objective\*\*: Blend piano fragments with retrieved sounds to create hybrid textures.

- \*\*Recommended Tools for Morphing\*\*:

- Zynaptiq MORPH (plugin): Real-time audio morphing with multiple algorithms.

- MeldaProduction MMorph (plugin): Spectral morphing based on harmonic features.

- CDP (Composers' Desktop Project): Command-line tools for experimental sound transformations.

- iZotope Iris: Spectral editing and layering of audio components.

---

### \*\*6. Developing Piano Materials Using Spectral Techniques\*\*

- Analyze combined sounds to extract spectral features such as harmonic content, inharmonicity, or formant structures.

- Translate these features into playable piano material:

- Microtonal clusters derived from partials.

- Rhythmic patterns inspired by transient behavior in electronic textures.

- Tools:

- SPEAR (Sinusoidal Partial Editing Analysis and Resynthesis) for detailed spectral analysis.

- Python libraries (`Librosa`, `Essentia`) for extracting spectral data programmatically.

---

### \*\*7. Structuring the Composition\*\*

- Develop a structure where electronic textures interact dynamically with live piano material:

- Start with fragmented electronic textures that gradually merge with live piano playing.

- Alternate between sections dominated by electronics and those focusing on acoustic piano.

- Consider spatialization techniques to enhance interaction between acoustic and electronic elements:

- Assign electronic sounds to different speakers in a multi-channel setup.

---

## Tools Summary

| Task | Recommended Tools |

|-----------------------------|----------------------------------------------------|

| Generating Piano Music | Suno |

| Splitting Audio | Librosa, Essentia, madmom |

| Filtering Fragments | sklearn.cluster, t-SNE/UMAP |

| Querying Sounds | Freesound API |

| Morphing Sounds | Zynaptiq MORPH, Melda MMorph, CDP |

| Spectral Analysis | SPEAR, iZotope RX, Python Libraries |

| Spatialization | Ambisonics tools (Reaper plugins like IEM Suite) |

---

## Final Notes

This documentation provides a detailed roadmap to guide your creative process while leaving room for experimentation at each stage. By following these steps, you'll be able to craft a cohesive yet exploratory composition that merges acoustic and electronic elements seamlessly.

Let me know if you'd like additional details on any section!