A Method for Orchestrating from Piano Pieces

Orges Leka

September 24, 2025

Overview

We describe a simple, reproducible procedure that converts a piano score into an orchestration for a "space-like synth" or any chosen ensemble. The method preserves the global duration and structural pacing while imposing a binary, bucketed texture across the ensemble.

Assumptions and Notation

- Let the input be a monophonic or polyphonic *piano* score with absolute onset times and durations.
- Every notated duration is an integer multiple of a smallest unit $\delta > 0$ (the smallest note value occurring in the piece).
- Pitch classes are taken modulo 12. For a MIDI pitch $p \in \mathbb{Z}$, its pitch class is $[p] \in \mathbb{Z}/12\mathbb{Z}$.
- Let the target ensemble have $V \geq 2$ instruments (voices) indexed by $v = 1, \ldots, V$. Each instrument v has a playable MIDI range $I_v = \{p \in \mathbb{Z} \mid p_{\min}(v) \leq p \leq p_{\max}(v)\}$.

Algorithm

Step 1: Choose the ensemble (with $V \geq 2$ instruments).

- Example: General MIDI pads/FX, strings, winds, or any custom set. Record each instrument's playable range I_v.

Step 2: Copy the piano voices cyclically to the ensemble instruments.

- Split the piano texture into *voices* in any consistent way (e.g. top line, inner lines, bass).
- Assign voice j to instrument $v = (j \bmod V) + 1$ in a repeating cycle.

Step 3: Replace chords by their lowest pitch (smallest MIDI number).

- For any simultaneity $\{p_1, \ldots, p_k\}$ at a given onset, keep only $\min\{p_1, \ldots, p_k\}$.

Step 4: Map pitches to each instrument's range by octave, preserving pitch class.

- For a note with pitch p and target instrument v, choose an octave shift $12m$ so that $p' = p + 12m \in I_v$ and $|p' - c_v|$ is minimized for some center $c_v \in I_v$.
- This keeps the pitch class $[p'] = [p]$ unchanged while moving to the nearest playable octave.

Step 5: Divide the entire piece into buckets of size δ.

- Let the piece span time $[0, T)$. Partition into $r = \lceil T/\delta \rceil$ buckets $b_1, \ldots, b_r$, where $b_i = [(i-1)\delta,\, i\delta)$.
- By assumption, every note's start/end align with bucket boundaries or their integer multiples.

Step 6: Label each bucket by a binary code padded to V bits.

- For bucket index $i \in \{1, \ldots, r\}$, write i in binary and left-pad with zeros to V bits:

$$i \mapsto (\beta_{i,1}, \beta_{i,2}, \ldots, \beta_{i,V}) \in \{0,1\}^V,$$

  where $\beta_{i,1}$ is the most significant bit. Bit $\beta_{i,v}$ controls instrument v in bucket b_i.

Step 7: For each note, count zeros z and ones o over its covered buckets.

- For a note assigned to instrument v spanning buckets $b_{i_s}, \ldots, b_{i_e}$, define

$$z = \#\{i \in [i_s, i_e] \mid \beta_{i,v} = 0\}, \qquad o = \#\{i \in [i_s, i_e] \mid \beta_{i,v} = 1\}.$$

- Thus $z + o = i_e - i_s + 1$ (the number of buckets covered by the note).

Step 8: Choose a mute rule based on (z, o) and apply it.

- Examples of rules:

$$\texttt{z>o}: \text{ mute if } z > o; \qquad \texttt{z>0}: \text{ mute if } z > 0; \qquad \texttt{o=0}: \text{ mute if } o = 0.$$

- If the rule triggers, silence the note (e.g. velocity $\to 0$); otherwise keep it.

Result. After these steps, the piano piece is orchestrated for the chosen ensemble. The total duration and high-level structure (entries, rests, phrase lengths) are preserved; the binary bucket mask imposes a repeatable textural logic across parts.

Remarks and Practical Tips

- **Generalization:** This method generalizes to pieces other than for piano, for instance for several voices and a larger ensemble.
- **Range mapping:** If I_v is narrow, prefer the octave closest to the instrument's tessitura to avoid excessive jumps.
- **Rule flavor:** `o=0` preserves notes only when all covered buckets are "1" for that voice (sparse, gated feel). `z>o` keeps notes in majority-1 regions (balanced). `z>0` is the most aggressive muter.
- **Determinism:** The mapping is fully deterministic once the ensemble, ranges, bucket size, and rule are fixed.

Minimal Pseudocode

`for each piano onset time` t `with chord` C`:`
keep $p = \min C$; assign to voice j; map to instrument $v = (j \bmod V) + 1$;
choose octave m so $p' = p + 12m \in I_v$ and closest to c_v;
determine bucket indices $[i_s, i_e]$ covered by the note;
compute (z, o) from bits $(\beta_{i,v})_{i=i_s}^{i_e}$;
if rule(z, o) then mute else keep.

Done! The piece is now orchestrated to the new ensemble while preserving overall duration and structure.