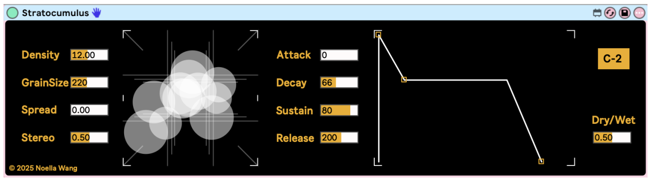
Stratocumulus: A Real-Time Granular Harmonizer with Pitch Tracking and MIDI Polyphony

Noella Wang
Electronic Production and Design
Berklee College of Music, Boston, MA
EP-491 Advanced Project in Electronic Production and Design
noellabwang@gmail.com

November 23, 2025



Density 12.00
GrainSize 220
Spread 0.00
Stereo 0.50
Attack 0
Decay 66
Sustain 80
Release 200
Dry/Wet 0.50

User interface of the Max for Live implementation of *Stratocumulus*.

Abstract

Stratocumulus is a real-time audio plugin that transforms monophonic audio input into a polyphonic granular instrument. The system combines a YIN[1]-inspired fundamental frequency tracker with a granular synthesis engine driven by MIDI input, enabling harmonically consistent, performative control of live vocal or instrumental signals. We describe the system architecture, the design of the pitch detection and granular synthesis stages, and present an evaluation of pitch accuracy and computational cost.

1 Introduction

Real-time vocal harmonization tools are widely used in live performance and studio production, yet most existing systems are conceived as effects processors with limited control over micro-temporal structure and timbre. At the same time, granular synthesis techniques offer rich textural possibilities but are rarely integrated with robust, low-latency pitch tracking and polyphonic control for live input.

Stratocumulus addresses this gap by transforming a monophonic audio source, such as a singing voice or melodic instrument, into a playable granular instrument controlled via MIDI. Incoming audio is continuously buffered; a dedicated analysis path estimates the fundamental frequency, and the resulting pitch information is used to align granular playback with the performer's input and the requested MIDI pitches. This design allows performers—particularly vocalists working with keyboardists or sequencers—to articulate harmonies and textures in real time while preserving a clear relationship between source pitch and synthesized output.

2 Related Work

Stratocumulus builds on prior work in granular processing and pitch-based effects while targeting a different operational role. Devices such as Mutable Instruments Clouds provide real-time granular processing with parameters such as position, grain size, and density, but do not perform continuous fundamental frequency tracking of the input signal[2]. Classic harmonizers such as the Eventide H910 offer keyboard-controlled monophonic pitch shifting using modulated delay lines[3], and contemporary pitch-correction and harmonization plugins typically emphasize formant-preserving transposition to create wider or corrected vocal textures, yet are presented primarily as insert effects rather than as playable granular instruments.

In parallel, a substantial body of research has examined granular synthesis for texture generation and time–frequency manipulation, from foundational treatments by Roads[4] to architectures for real-time granular instruments and performance interfaces[5,6]. However, we are not aware of a system that integrates real-time f_0 tracking, MIDI-driven polyphonic granular synthesis, and an interface explicitly designed to treat the live input as the excitation for a playable instrument. Stratocumulus is proposed to occupy this design space. A structured comparison with representative systems is provided in Table 1.

3 System Overview

A high-level overview of the Stratocumulus signal flow is shown in Fig. 1. The incoming mono signal is written into a circular buffer shared by the analysis and synthesis stages.

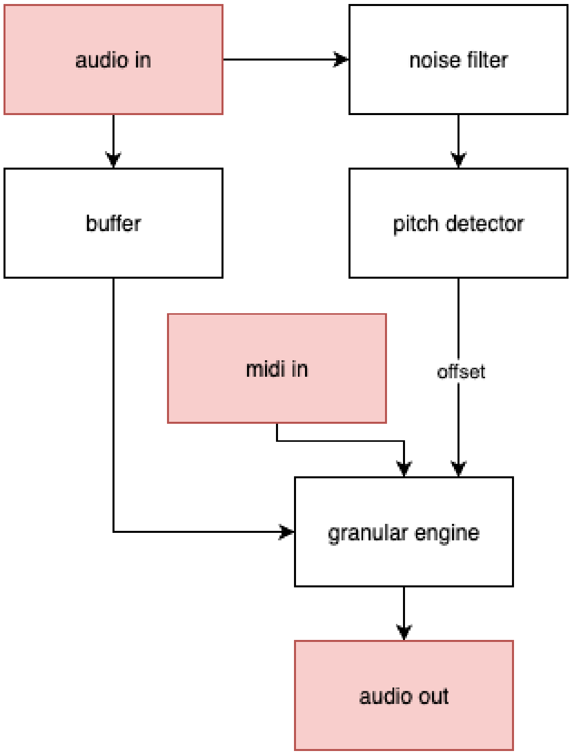


Figure 1: Signal flow of the *Stratocumulus* system. The input audio is analyzed for fundamental frequency, which informs the granular synthesis engine driven by MIDI note events.

In the analysis path, the buffered audio is passed to a YIN-inspired pitch detector, which produces a continuous estimate of the fundamental frequency $f_0[n]$ and an associated confidence value. In parallel, MIDI note events specify target pitches; these are converted to transposition ratios relative to the most recent reliable f_0 estimate. The granular engine then reads grains from the circular buffer and retunes them according to these ratios, producing a polyphonic output that remains aligned with the performer's live input.

4 Pitch Tracking and Note Mapping

4.1 Pitch Detection Approach

Stratocumulus employs a YIN-based fundamental frequency estimator[1], chosen for its robustness on monophonic, quasi-harmonic sources such as the human voice. We follow de Cheveigné and Kawahara's definitions of the difference function $d(\tau)$ and the cumulative mean normalized difference function $d'(\tau)$, and search over lags within a constrained pitch range (e.g., 65–2000 Hz).

The fundamental period is selected as the first lag τ_0 at which $d'(\tau)$ falls below a fixed threshold and attains a local minimum, favoring the true fundamental over its harmonics. We then apply parabolic interpolation around τ_0 to obtain a sub-sample period estimate $\hat{\tau}$ and compute the instantaneous fundamental frequency as

$$\hat{f}_0 = \frac{F_s}{\hat{\tau}}. \tag{1}$$

Our implementation is informed by an existing offline C version of YIN on GitHub[7] that processes fixed, non-overlapping buffers. For Stratocumulus, we adapt this to real time by running YIN on overlapping frames from a circular input buffer, restricting the lag range to vocal/instrumental pitches, and applying confidence gating and temporal smoothing to obtain stable, low-latency f_0 estimates.

The detector operates continuously on the live stream with a hop size chosen to balance latency and low-frequency resolution. Frames with low periodicity or low YIN confidence are discarded, and the last reliable estimate is held to avoid spurious pitch jumps.

4.2 Pitch Tracker Implementations

We implemented three successive YIN-based trackers (YIN2, YIN3, and YIN4) to explore the trade-off between accuracy and computational cost in a real-time plugin context.

YIN2 used the standard YIN procedure with a full-resolution difference function, cumulative mean normalization, and parabolic interpolation evaluated at a high update rate. While this configuration provides stable estimates for monophonic vocal and instrumental input, it is computationally intensive and approaches the limits of real-time operation in the target environment.

YIN3 refines this design by restructuring the analysis pipeline and introducing more robust minimum selection, short-term history constraints, and confidence-weighted smoothing. These additions reduce spurious

System	f_0 tracking	Granular engine	Control
Stratocumulus	Yes	Yes	MIDI in
MI Clouds	No	Yes	Panel / CV
Eventide H910	No	No	Built-in keyboard
Commercial harmonizers	Yes	FX only	Preset

Table 1: Comparison of Stratocumulus with representative systems.

octave errors and frame-to-frame jitter but only modestly lower CPU usage.

YIN4 is an optimized single-block implementation designed specifically for Stratocumulus. It reduces window size, evaluates the difference function with adaptive sub-sampling over τ, decouples the analysis rate from the audio block size, and applies a lightweight median filter, adaptive smoothing, and hysteresis-based gating. This preserves the essential behavior of YIN while substantially decreasing the number of operations per frame, making continuous f_0 tracking feasible alongside the granular engine.

We compare the computational cost and pitch accuracy of these variants in Section 7 (Fig. 3, 4).

5 Granular Synthesis Engine

The initial granular engine design was adapted from a public gen~ patch by Ed Roberts (toneparticle)[1], which implements a Hann-windowed, multi-voice buffer-based grain cloud. For Stratocumulus, we modified this architecture for live performance by (i) integrating it with the real-time input buffer used by the f_0 tracker, (ii) exposing MIDI-driven pitch control and grain parameter mapping suitable for performance, and (iii) fine-tuning the buffer position to avoid artifacts when the grain is played across the circular buffer boundary.

The granular engine comprises 30 parallel grain generators implemented as a polyphonic voice bank. Each voice reads from the shared circular buffer with independently specified onset time, duration, and transposition factor. Grain duration is controlled by a user parameter (e.g., 10–200 ms) and shaped by Hann-window envelopes to ensure smooth overlap.

Figure 2 illustrates the granular engine's signal flow. It uses parameters to determine the playback speed and position from the buffer, grain size, and density. A stereo parameter assigns grains pseudo-randomly across the stereo field after the final grain. These controls are exposed in the user interface to support expressive, performance-oriented manipulation.

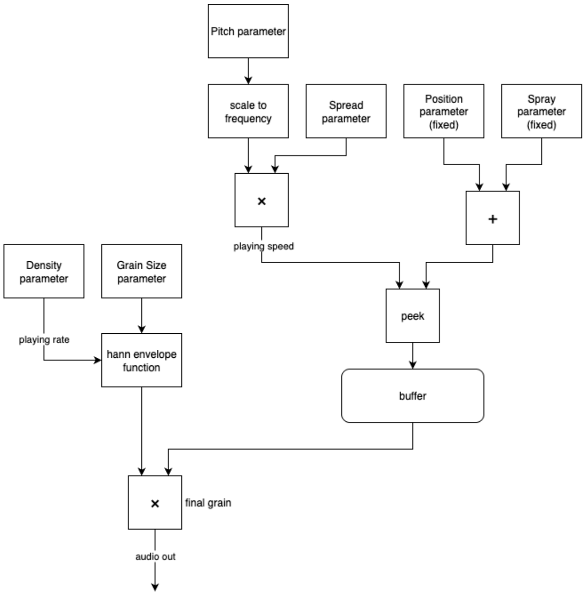


Figure 2: Signal flow of the granular engine.

6 Implementation

A prototype implementation was developed in Max for Live using `gen` for granular processing and the `yin~` external from the Max Sound Box collection by IRCAM[2] for pitch detection. This environment enabled rapid evaluation of parameter mappings and interaction design with minimal integration overhead. We also implemented a version of Max for Live device using the custom YIN tracker described in Section 4.2.

For deployment as a VST/AU/standalone plugin, we used the RNBO–JUCE template by Cycling'74[8] to port the source code into a custom C++ implementation. The graphical interface is constructed using the JIVE extension for a layout with all synthesis parameters. Both implementations share the same conceptual signal flow, facilitating direct comparison of performance and sound quality.

[1] `https://youtu.be/VU2TQmxte9A`

[2] `https://forum.ircam.fr/projects/detail/max-sound-box/`

7 Evaluation

We evaluated Stratocumulus with respect to pitch accuracy, latency, and computational load on a MacBook Pro with M3 Pro chip (5+6, no background processes) running Max 9.0.0, and used the `plot` object for visualization. We compared the three pitch tracker implementations with the same vocal sample for CPU usage comparison, and compared YIN2 and YIN4 with three different vocal samples for pitch accuracy comparison.

Figure 3 reports average CPU usage for the three implementations under identical test conditions. YIN2 exhibits the highest load, with usage frequently approaching real-time limits. YIN3 provides a small reduction, whereas YIN4 reduces CPU consumption significantly while remaining stable across input types, confirming the effectiveness of the structural and sampling optimizations.

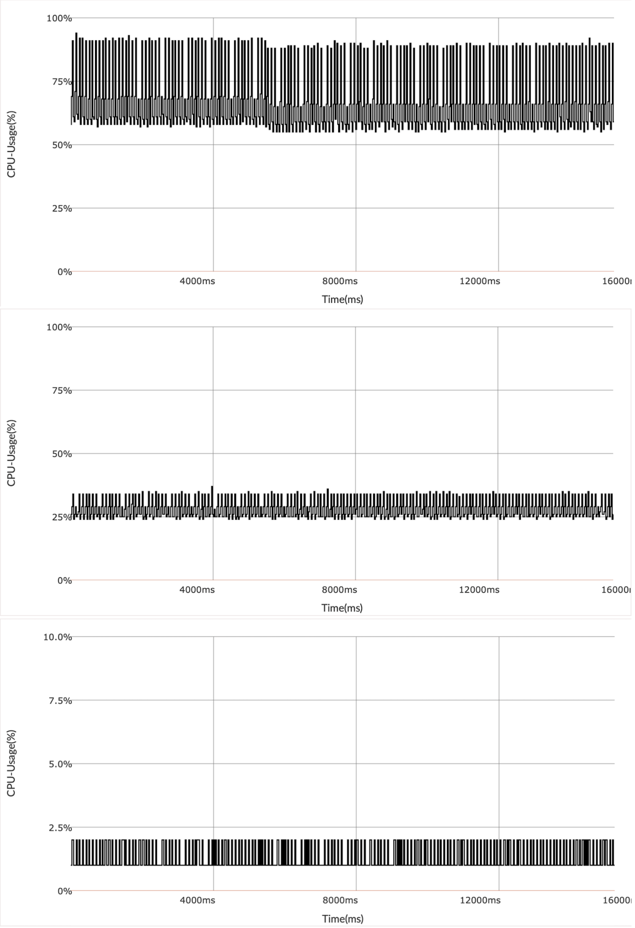


Figure 3: CPU usage comparison for YIN2, YIN3, and YIN4 under identical test conditions. The Y-axis for the third plot differs from the first two to accommodate lower values.

Figure 4 reports the frequency–tracking behavior of the YIN2 and YIN4 implementations on three vocal test samples, shown alongside iZotope RX spectrograms that we use as a visual reference for assessing tracking stability. Aside from their differing behavior to address silence, both trackers capture the overall pitch contour effectively.

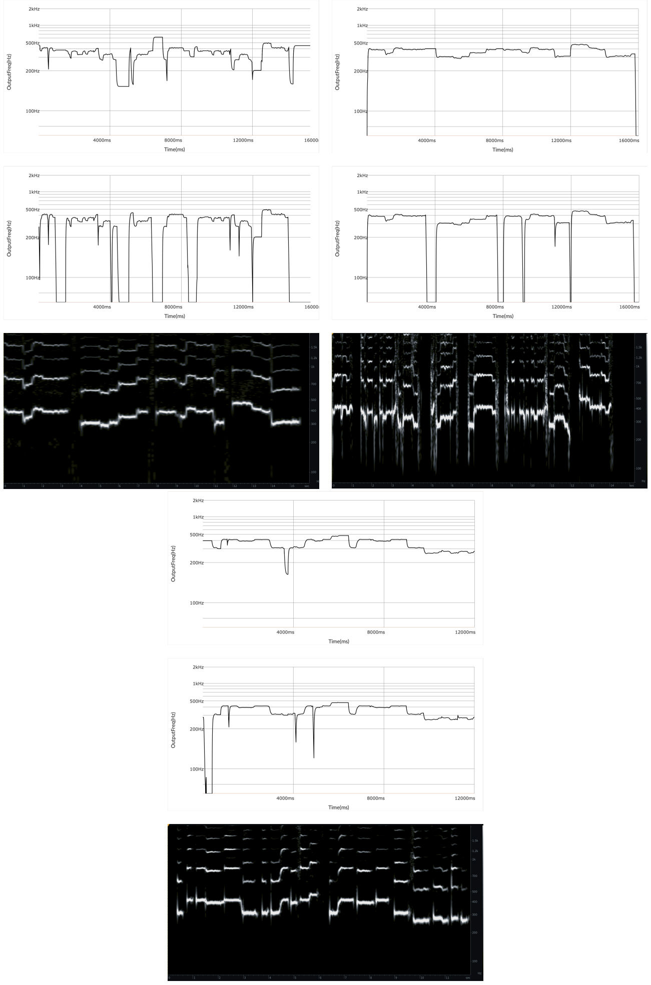


Figure 4: Comparison of frequency detection accuracy between YIN2, YIN4, and iZotope RX spectrograms for three test samples.

8 Musical Applications

Stratocumulus enables several musical applications beyond traditional harmonization and granular processing. Because the system utilizes real-time pitch tracking, it can adapt to both instrumental performance and compositional workflows.

8.1 Live Performance

The device functions as a performance-ready harmonizer for voice and monophonic instruments. By tracking the performer's fundamental frequency in real time, Stratocumulus generates polyphonic layers that follow the performer's phrasing, articulation, and timbral fluctuations.

8.2 Compositional Workflow

In a production environment, Stratocumulus can be used as a compositional tool to add texture and layers to existing vocal or instrumental tracks, transforming them into rich, harmonically complex soundscapes.

9 Conclusion and Future Work

Stratocumulus introduces a real-time framework that couples continuous fundamental frequency tracking with MIDI-driven granular synthesis, enabling monophonic audio—particularly voice and melodic instruments—to function as a controllable polyphonic instrument. Through three successive YIN-inspired tracker implementations, we examined the trade-off between accuracy and computational efficiency in a performance-oriented context, showing that the optimized YIN4 variant achieves stable, low-latency estimates at substantially reduced computational cost. The granular engine, adapted for live input and designed with performance control in mind, supports a wide range of musical applications extending from harmonization to textural sound design.

Stratocumulus is released as an open-source project, with source code and documentation available online.[3] By making the implementation publicly accessible, we aim to support further experimentation by artists, researchers, and developers, and to contribute a reproducible reference for real-time granular–pitch-tracking systems.

Future work will focus on expanding both the expressive potential and configurability of the system. On the synthesis side, we plan to introduce alternative grain window shapes and explore more advanced granular architectures to improve timbral variation and support higher-density cloud generation. Parameters such as circular buffer size, analysis rate, and pitch-detection thresholds will be exposed to allow greater adaptability across instruments and performance conditions. Finally, although the current interface is sufficient for internal testing, a refined plugin UI is planned to improve workflow clarity, provide more immediate visual feedback, and better support live-performance use cases.

Acknowledgments

This project was completed as part of EP-491 Advanced Project in Electronic Production and Design at Berklee College of Music. I would like to thank Dr. Akito van Troyer for his guidance, feedback, and support throughout the development of Stratocumulus, as well as Dr. Ryan Page and the Electronic Production and Design faculty for their instruction and insight across my studies. I also thank my peers for helpful discussions and testing, and the open-source community whose tools and resources supported this work.

References

[1] A. de Cheveigné and H. Kawahara, "Yin, a fundamental frequency estimator for speech and music," *The Journal of the Acoustical Society of America*, vol. 111, no. 4, pp. 1917–1930, 2002.

[2] É. Pichenettes and M. Instruments, *Manual – Mutable Instruments Documentation: Clouds*, 2025. Accessed: 2025-11-22.

[3] Eventide Inc., *H910 Harmonizer Hardware Manual*, 1975. Accessed: 2025-11-22.

[4] C. Roads, "Introduction to granular synthesis," *Computer Music Journal*, vol. 12, no. 2, pp. 11–13, 1988. Accessed: 2025-11-22.

[5] R. Bencina, "Implementing real-time granular synthesis," 2001. Online article.

[6] M. Sanganeria and K. J. Werner, "Grain proc: a real-time granular synthesis interface for live performance," in *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, (Daejeon, Korea), May 2013.

[7] A. Fernandez, "Yin pitch tracking." GitHub repository. Accessed: 2025-11-09.

[8] "Rnbo juce examples." GitHub repository. Accessed: 2025-11-09.

[3] https://github.com/nonocutt/Stratocumulus