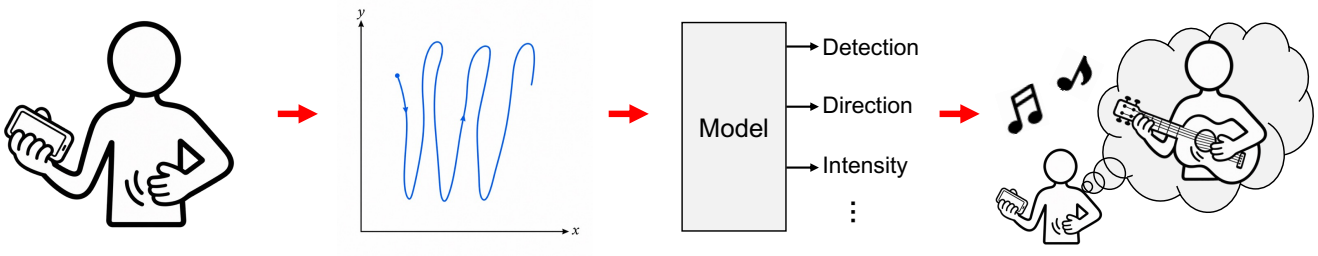
STRUM: Strum Tracking via Reflected Ultrasound on Mobiles

JoongWon Shin
leodal@snu.ac.kr
Seoul National University
Seoul, South Korea



(a) Mid-air strumming (b) Acoustic 2D hand tracking (c) ML-based strum recognition (d) Guitar sound output

Figure 1: Overall workflow of STRUM.

Abstract

STRUM is a smartphone-only virtual guitar application, as a project for the Fall 2026 "Mobile Computing and Its Applications" course at Seoul National University. STRUM uses the phone's built-in speaker and microphones for ultrasonic 2D hand tracking, then applies a machine learning model to detect strums and estimate their direction and intensity. Users select chords on the touchscreen with one hand and strum in mid-air with the other to produce guitar sounds. We plan to apply quantization and pruning to reduce inference latency and support responsive interaction.

Keywords

Acoustic Sensing, Gesture Recognition, Around-Device Interaction, Low-Latency Interaction

1 Motivation

Playing the guitar is fun, but it requires a physical instrument. However, guitars are bulky and heavy, making them inconvenient to carry around for casual or spontaneous playing.

Mobile apps such as GarageBand [1] offer a portable alternative, but both chord selection and strumming are confined to a small touchscreen. This makes it difficult to reproduce the physical hand movements of guitar playing, creating a gap between touchscreen interaction and playing a real instrument. In addition, limited screen space restricts the number of chords that can be displayed for immediate selection during performance.

I propose STRUM for users who want to casually enjoy music and play spontaneous accompaniment without carrying a guitar. Using only a smartphone, users can select chords on the touchscreen with one hand and control strumming direction and intensity through mid-air gestures with the other. STRUM aims to provide a portable, two-handed virtual guitar experience using the phone users already carry, without additional equipment.

2 Literature Review

In this project, I reviewed related work in two areas: acoustic-based gesture sensing and mid-air guitar interfaces. LLAP [6] uses a smartphone's built-in speaker and microphones to analyze phase changes in reflected acoustic signals and track hand trajectories. FingerIO [4] transmits OFDM-based acoustic signals and precisely estimates the arrival times of reflected signals to track 2D finger positions. These studies present methods for tracking hand movements using smartphone audio hardware without additional wearable devices and serve as the technical foundation for implementing STRUM.

Prior mid-air guitar interfaces include MagiGuitar, Air Guitar, and Modified Virtual Air Guitar [2, 3, 5]. agiGuitar [3] enables mid-air guitar playing by detecting the movement of an external permanent magnet using a smartphone's magnetometer. Air Guitar [2] is an Android-based air-guitar prototype whose subsequent sensing system, uses external infrared sensors and a microcontroller to detect upward and downward hand movements and transmit the results to the smartphone. Modified Virtual Air Guitar [5] uses a webcam and colored gloves to track both hands and generates sound. STRUM differs from these systems by implementing a guitar interface using only a single smartphone, without external devices or equipment.

3 Proposed Design

3.1 System Architecture

STRUM's system architecture consists of four stages: data collection, hand tracking, ML-based inference, and guitar sound output. First, during data collection, the smartphone's speaker transmits ultrasonic signals at multiple frequencies, and two microphones capture reflections from the hand. On the touchscreen, users select the chord to play by specifying which strings to press and at which positions. Second, the hand tracking stage applies LLAP-based signal processing to estimate highly accurate 2D hand trajectories. Third, the ML-based inference stage uses a sliding window over

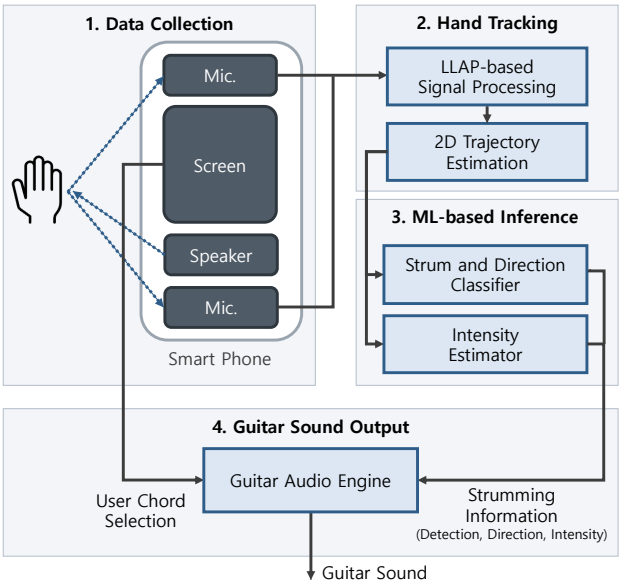


Figure 2: System architecture of STRUM.

recent trajectory data to determine whether a strum has occurred and identify its direction, while an intensity estimator estimates the strumming intensity. Finally, when a strum is detected, the guitar sound output stage plays the corresponding guitar sound according to the user-selected chord and the strumming direction and intensity. This provides a two-handed playing interface in which users select chords with one hand and perform mid-air strumming gestures with the other.

3.2 Challenges and Solutions

Implementing STRUM presents three potential challenges: tracking errors caused by smartphone motion, interference from non-target reflections, and the low response latency required for real-time performance.

First, since users hold the smartphone while playing, device motion may affect the estimation of the hand's relative position. To mitigate this effect, I will investigate a compensation method that uses the smartphone's IMU data as auxiliary input.

Second, acoustic reflections from the torso, arms, and surrounding objects may overlap with reflections from the hand, reducing tracking accuracy. To mitigate this interference, I will use LLAP [6]'s background reflection removal and multi-frequency fusion techniques. Background reflection components that remain relatively stable over time will be estimated and removed, and distance-change estimates obtained at multiple frequencies will be combined to reduce multipath effects.

Third, the delay between hand movement and sound output must be minimized to provide a natural playing experience. I will use a lightweight model and apply model optimization techniques, such as quantization and pruning, to reduce inference latency. I will also adjust signal processing windows and audio buffers to reduce overall response latency.

Table 1: Implementation plan.

Item	Plan
Device	Galaxy S24+ (Android).
Development	Android Studio; Kotlin for the application and C++ (NDK) for signal processing.
Dataset	Self-collected acoustic signals and hand trajectories, labeled by strum occurrence, direction, and relative intensity.
Hand tracking	Adapt LLAP's public code [6] to Android and implement 2D trajectory estimation.
Model training	TensorFlow/Keras; train a lightweight strum and direction classifier and an intensity estimator using sliding-window trajectories.
On-device inference	Deploy models using LiteRT and investigate quantization and pruning.
Development order	Acoustic recording and playback → hand tracking → dataset collection and model training → application integration.

Table 2: Evaluation plan.

Item	Evaluation Method
Detection accuracy	Measure strum/non-strum classification accuracy on held-out, manually collected test data containing strumming and various non-strumming motions.
Inference latency	Measure the mean model inference time per sliding window on the Galaxy S24+.
Optimization comparison	Compare detection accuracy and inference latency before and after applying different optimization methods, including quantization and pruning, using the same test data and device.
Selection goal	Select the optimization method that reduces inference latency while preserving detection accuracy as much as possible.

4 Prototype Implementation Expected Outcomes

The expected outcome of this project is a smartphone-only guitar-playing application that integrates touch-based chord selection with acoustic-based mid-air strum detection. Users will be able to select chords with one hand and strum with the other without additional equipment, with guitar sounds reflecting the detected strumming direction and intensity.

I will also compare recognition performance and processing latency before and after applying model optimization and pipelining between hand tracking and inference to analyze the trade-off between the two. Through this project, I aim to assess the feasibility of a smartphone-only mid-air guitar interface and identify the design requirements for real-time playing.

References

[1] Apple. n.d.. *Play the Guitar in GarageBand for iPhone.* Retrieved November 7, 2026 from https://support.apple.com/guide/garageband-iphone/play-the-guitar-chs39282a88/ios

[2] Alberto Boem. 2012. *Air Guitar.* Project website. Retrieved November 7, 2026 from https://www.albertoboem.com/?project=air-guitar

[3] Hamed Ketabdar, Hengwei Chang, Peyman Moghadam, Mehran Roshandel, and Babak Naderi. 2012. MagiGuitar: A Guitar That Is Played in Air!. In *Proceedings of the 14th International Conference on Human-Computer Interaction with Mobile Devices and Services Companion (MobileHCI '12)*. Association for Computing Machinery, 181–184. doi:10.1145/2371664.2371704

[4] Rajalakshmi Nandakumar, Vikram Iyer, Desney Tan, and Shyamnath Gollakota. 2016. FingerIO: Using Active Sonar for Fine-Grained Finger Tracking. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. Association for Computing Machinery, 1515–1525. doi:10.1145/2858036.2858580

[5] John Santiago, Stanly Samuel, and Rahul Sawn. 2015. Modified Virtual Air Guitar: A Concept Realized using Image Processing Techniques. In *Proceedings of the 3rd International Conference on Context-Aware Systems and Applications (ICCASA)*. ACM. doi:10.4108/icst.iccasa.2014.257296

[6] Wei Wang, Alex X. Liu, and Ke Sun. 2016. Device-Free Gesture Tracking Using Acoustic Signals. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking (MobiCom '16)*. Association for Computing Machinery, 82–94. doi:10.1145/2973750.2973764