Pixel Soundscape

A Visual Instrument for Environmental Listening

By Sohyun Jun

https://vimeo.com/1153339333?fl=pl&fe=sh

Introduction

This project explores how environmental video data can be translated into sound to reveal the often-overlooked sonic relationships between human, non-human, and urban environments through interactive audiovisual synthesis.

Concept and Background Research

In contemporary urban life, direct engagement with nature has increasingly become a luxury. Cities are composed of layered environments in which humans, non-human organisms, technologies, and hybrid entities coexist. This project begins by questioning whether human and non-human relationships within urban environments can be harmonised through sound. Rather than treating environmental sound as background noise, the work considers sound as an active medium through which relationships between beings and spaces can be perceived and reimagined.

Figure 1. Songs of the Horizon

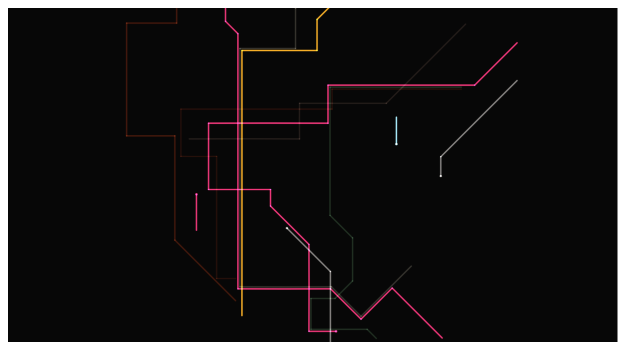

Figure 2. Mta.me

The primary visual inspiration for this project comes from media artworks (Figure 1) that combine video imagery with generative sound, particularly works in which lines move across video frames and generate sound based on image data (de Valk, 2025). Additionally, Mta.me (Figure 2) shows how data could translate into sound (Chen, 2025). These works demonstrate how visual information can be translated into auditory experiences. Building on this approach, I aimed to recreate and modify this visual–sonic relationship in my own way, focusing on environmental interpretation rather than musical composition alone.

This project explores the interaction between humans, computers, and non-human entities in urban environments through sound. Environmental dynamics are interpreted as sonic information, with brightness and movement in video footage translated into audible oscillations. Pijanowski (2024) argues that sound should not be understood merely as noise, but as a carrier of ecological and behavioural meaning for organisms within an environment. He describes sound as an ecological layer that both human and non-human beings actively perceive and respond to. This perspective strongly informs the conceptual grounding of the project.

By mapping visual and spatial patterns to sound, the installation highlights often-overlooked acoustic relationships between city life, natural elements, and human perception. In doing so, it offers a playful and interactive experience in which audiences can engage by controlling sensors and interacting with the pixel data of the videos.

Technical Implementation

The development of this project was informed through discussion and peer exchange. During a lab session, I discussed my idea with a colleague, Jiayun Song, who introduced me to her own work involving video input and a fixed vertical line that generates sound from pixel data in p5.js (Figure 3). Her system divided the canvas into seven sections, assigning discrete pitches to each region. I adapted part of her approach, specifically the method of reading pixel brightness values and converting them into oscillator control data.

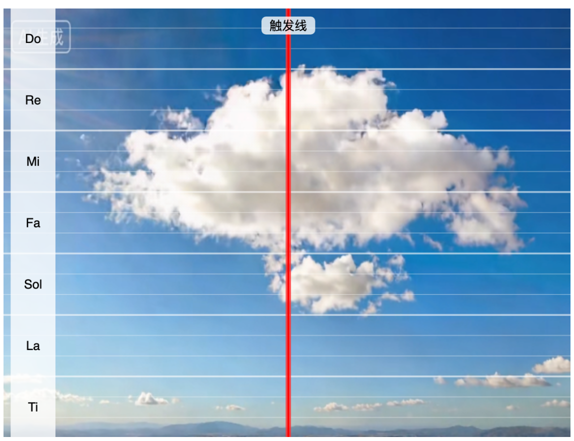

Figure 3. Screenshot of Jiayun's work

The key difference between her work and mine lies in the sound-mapping strategy. Rather than dividing the screen into fixed sections, my system defines minimum and maximum pitch values across the entire screen, allowing for smoother and more continuous pitch variation. This approach enables more dynamic and expressive sound output. Additionally, the vertical trigger line moves horizontally across the video frame, introducing a form of controlled randomness that produces evolving sonic textures.

Throughout the development process, I encountered several technical challenges, particularly in managing sound synthesis, interaction logic, and performance. These issues were addressed through iterative testing, peer discussion, and assistance from AI-based tools. A full AI declaration is provided in the end of this section.

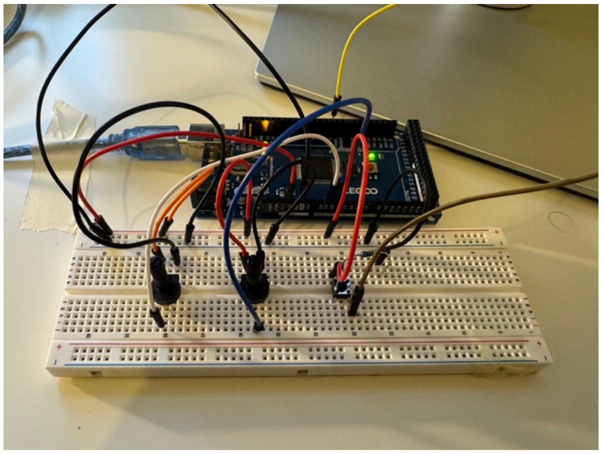

Figure 4. Arduino Connection

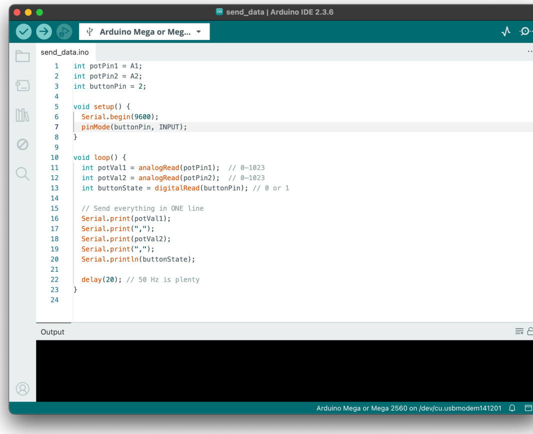

Figure 5. IDE script to send data

In summary, the system captures video input and samples pixel data along a vertical trigger line, calculating brightness and positional values. These values are mapped to p5.js oscillators using FM synthesis, controlling parameters such as carrier frequency, modulation depth, and filtering. Interactive controls are implemented through Dat.GUI and Arduino input (Figure 4, 5), allowing real-time adjustment of pitch range, FM amount, and waveform type. Visual particles indicate points of interaction between the line and the video, reinforcing the connection between visual and sonic feedback. Responsive canvas scaling and fullscreen handling ensure an immersive experience across different screen sizes and devices.

AI declaration

I acknowledge the use of [1] ChatGPT (https://chat.openai.com/) to [2] make system for particle generation with scale and opacity change for natural animation. I entered the following prompts on 15 December 2025:

- [3] How do I make array of particles made of circle function naturally disappear after hitting the pixels

[4] The output from the generative artificial intelligence was used in a part of function updateAndDrawParticles and modified as the project developed.

I acknowledge the use of [1] ChatGPT (https://chat.openai.com/) to [2] create circles where the line hits y position of the pixels without being heavy on performance. I entered the following prompts on 29 Decemeber 2025:

- [3] I think I create too many particles every frame. It is lagging. How to limit particle creation with a variable for max particle per frame?

[4] The output from the generative artificial intelligence was used in part of function detectVideoFM, especially the use of hitCount, and MAX_PARTICLES_PER_FRAME, as well as the appropriate way to add and remove elements in hitY array.

I acknowledge the use of [1] ChatGPT (https://chat.openai.com/) to [2] crop videos when it is set to fullscreen without making video stretched. I entered the following prompts on 12 January 2026:

- [3] I want to do fullscreen if I press f, but I have a small conflict. because of vieo width and height, the contents might need to be cropped according to screen (browser) size. How to acheive this without streching the screen, avoiding setting too high resolution for the video for performance?

[4] The output from the generative artificial intelligence was used in creating a function loadVideoByState and applied with function keyPressed.

Reflection and Future Development

This project provided valuable insight into interactive sound design and iterative coding practices. One notable strength was the effective use of AI-assisted problem-solving. Having learned programming prior to the widespread use of AI tools, I was familiar with relying on documentation and manual debugging; however, AI support in this project accelerated technical troubleshooting and enabled greater focus on conceptual exploration and aesthetic refinement.

The project also revealed certain limitations. Due to a limited number of potentiometer sensors, not all sound parameters could be physically controlled during installation. A future version would incorporate additional sensors to allow more comprehensive real-time control. Additionally, the particle system could be further developed with more dynamic visual behaviours, such as colour and shape changes driven by Arduino input, to strengthen audiovisual coherence.

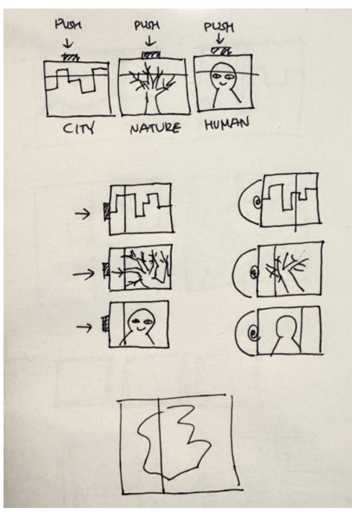

Figure 6. Sketch for Pin-up exhibition

Future development could expand the work into a larger-scale installation. For example, multiple screens showing different environments connected through physical elements such as elastic bands could allow users to control the speed of the lines by flicking the strings, transforming the system into a more embodied, instrument-like experience (Figure 6).

References

Pijanowski, B.C. (2024). *Principles of Soundscape Ecology*. University of Chicago Press.

Werner de Valk. (2025). *Songs of the Horizon - Werner de Valk*. [online] Available at: https://wernerdevalk.nl/songs-of-the-horizon/ [Accessed 29 November 2025].

Alexander Chen. (2025). *MTA.ME*. [online] Available at: http://mta.me/ [Accessed 20 November 2025].

p5-serial (2022). *GitHub - p5-serial/p5.serialserver: Server for use with p5.serialport.* [online] GitHub. Available at: https://github.com/p5-serial/p5.serialserver [Accessed 2 Jan. 2026].