Sounding Canvas: An Embedded, Networked, Interactive Audio Artwork

Luciano Ciamarone†, Dora Motèque†, Marco Giordano*
*Dpt. of Information Engineering, Computer Science and Mathematics (DISIM),
University of L'Aquila, L'Aquila, Italy
Email: marco.giordano3@graduate.univaq.it
†Independent Researcher
Email: valmovigo@musician.org

***Abstract*—The "Sounding Canvas" is an interactive art installation that transforms a traditional canvas into a dynamic, sound-producing interface. Users interact by touching different areas of the canvas, triggering a variety of sounds that appear to emanate directly from the artwork. This experience is facilitated by an embedded system comprising of capacitive sensors connected to an Arduino, a Raspberry Pi 4 for processing and sound generation logic, and a HiFi Berry Amplifier for audio output. The system employs a machine learning decisional algorithm in Python, ensuring that touch interactions result in evolving and varied auditory responses. Furthermore, each canvas can connect to a remote server, allowing multiple installations to share "touch events" in real-time, creating the potential for networked, participatory experiences. This paper details the architecture, implementation, and interactive qualities of the Sounding Canvas, highlighting its relevance as a responsive sound installation and a platform for sonifying touch-based sensor data.**

***Index Terms*—Embedded audio systems, responsive sound installations, participatory performance, sonification, capacitive sensing, Raspberry Pi**

I. Introduction

The Sounding Canvas project, conceived by Iranian artist Dora Motèque, transforms traditional paintings into dynamic, sonic interfaces, pushing the boundaries of artistic engagement. Each canvas subtly embeds unseen sensors and computational means, generating unique sounds upon touch, thereby shifting the viewer's experience from passive observation to active, responsive engagement. The visual forms of the artworks are a direct result of Ms. Motèque's semiographic research, which combines Persian characters and musical notation (see Fig. 1).

Sensor interactions are streamed to a Raspberry Pi hidden behind each canvas, where a real-time decisional algorithm analyses the evolving touch sequence. Rather than replaying fixed samples, the model infers the visitor's intent and generates context-aware sounds, avoiding repetition and sustaining interaction. Every canvas also broadcasts its events to a central Python WebSocket server, allowing geographically separated units to listen and react to one another. This network transforms isolated gestures into a distributed, collaborative soundscape. By uniting traditional painting, embedded sensing, adaptive sound generation, and an Internet-of-Sounds backbone, Sounding Canvas pushes interactive art toward interconnected, responsive and deeply engaging experiences.

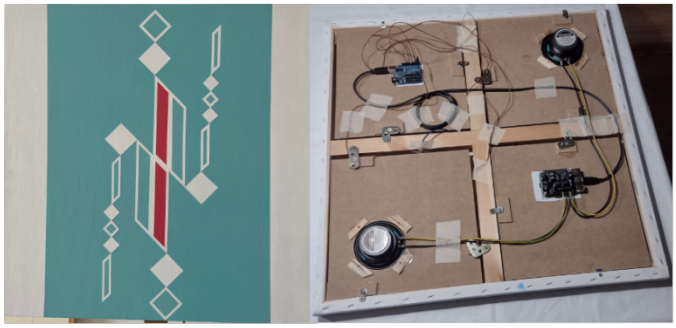

Fig. 1. Front and Rear view of *"Echo of Lines"*

II. Related Work

Peter Vogel's "sound sculptures" (e.g., Klangskulptur 3, 1981) translated a visitor's shadow on photo-resistors into logic-controlled oscillator bursts, establishing a minimalist cause-and-effect template for responsive art [1]. Scenocosme, the French artist duo composed of Grégory Lasserre and Anaïs met den Ancxt, expanded the sensory palette: Akousmaflore couples skin conductance with plant electrodes to trigger granular textures, while Kymapetra embeds piezo discs in polished stones so micro-vibrations drive spectral processing [2], [3]. Ian Costabile's Sound Canvases have built-in LDR sensors that convert user touch to audio spatialisation [4]. Anders Lind's large-scale pieces, such as The Web and Lines, use overhead motion tracking and networked loudspeakers to let audiences weave harmonic structures through bodily movement [5]. Collectively, these works span local analogue sensing, embodied touch sonification, embedded painting-instruments, and distributed gestural environments.

Sounding Canvas combines and extends those threads by (i) concealing a high-resolution capacitive array that preserves the painting's visual integrity, (ii) running an on-board probabilistic model that learns each visitor's evolving touch sequence to avoid repetitive mappings, (iii) broadcasting events via a WebSocket backbone so geographically separated canvases co-create a shared soundscape, and (iv) fusing these technologies with Dora Motèque's semiographic research, delivering a networked artwork that supports fine-grained tactile dialogue, cross-site collaboration, and reproducible research.

III. System Overview

The *SoundingCanvas* is an interactive audiovisual system that integrates tactile sensing, real-time gesture interpretation, and distributed communication to produce responsive sound behavior across multiple canvases. Each unit is self-contained, embedding hardware and software components that manage user interaction, audio playback, and inter-device messaging. Fig. 2 illustrates the high-level architecture, highlighting the primary subsystems: capacitive sensors embedded behind the canvas, a Raspberry Pi running the main logic, an Arduino for sensor data acquisition, and a HiFiBerry audio amplifier for sound output. The software stack implements a gesture-driven decision model, network synchronization, and audio rendering. The following subsections describe the hardware and software layers in detail.

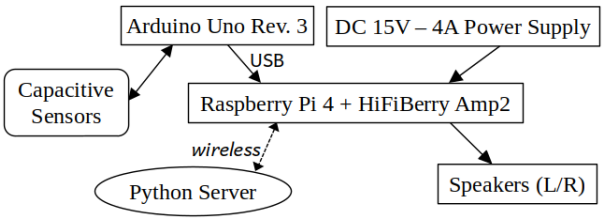

Fig. 2. Hardware architecture of the Sounding Canvas.

A. Hardware Platform

The interactive system is built on a robust hybrid hardware platform [6], combining high-fidelity audio, dedicated microprocessing, and precise capacitive sensing. A Raspberry Pi 4 Model B handles audio playback and communication protocols, routing output through a HiFiBerry Amp2 HAT connected to two 4-inch loudspeakers for clear, responsive sound.

Touch input is captured by capacitive sensors made from aluminum foil pads embedded beneath the artwork, geometrically aligned with its visual composition. Each sensor forms an RC circuit with a 1.4 MΩ resistor, in which capacitance changes are digitized in real time by an Arduino Uno Rev 3. Sensors are grouped into spatially proximal clusters, with each cluster associated with an Arduino pin to form an integer-ID channel, and its number limited by the available Arduino pins. A set of sound files is linked to each channel, and these are stereo-panned to localize the sounds, making them appear to emanate directly from that area on the canvas. The Arduino transmits continuous sensor data to the Raspberry Pi via USB. This integration aligns the artwork's visual and functional layers. Additionally, the Raspberry Pi's network capabilities support multi-canvas collaboration through WebSocket communication with a central Python server, enabling synchronized 'touch event' exchange across multiple Sounding Canvases.

B. Software Stack

The SoundingCanvas system is primarily implemented in Python [7], and its design features effective hardware-network integration. Each canvas runs a local Python application that interfaces with touch sensors (via serial communication with an Arduino), manages audio playback (using `pygame` or `pydub`), and handles inter-canvas communication over Wi-Fi. Asynchronous WebSocket communication, powered by the `websockets` library, enables bi-directional messaging with a central relay server. This design ensures responsiveness, allowing each canvas to concurrently process sensor input, network events, and audio rendering without blocking.

A decisional algorithm maps in real-time touch inputs to spatialized audio responses, ensuring perceived sound origin matches touch location via stereo panning. A high-order Markov model predicts user interactions and remote events, enabling context-aware sound selection. The system runs over a local or internet-connected network, supporting both on-site and remote installations. Its modular software stack ensures portability (e.g., on Raspberry Pi with HiFiBerry) and future extensibility toward adaptive behaviors and user feedback loops.

As an off-line preparatory step, a convolutional neural network (CNN) has analyzed visual features (shape, color, texture) of the painting to guide the composition of sound samples, reinforcing the link between visual and auditory elements.

IV. Interactive Sound Design

At the core of the *SoundingCanvas* sound design is a cross-modal mapping between visual and auditory perceptual spaces. This system comprises an *off-line* component for learning this mapping and an *on-line* component for interactive sound generation. The off-line phase analyzes the visual surface using a Convolutional Neural Network (CNN) [8], [9] to produce high-dimensional embeddings $\mathbf{x} \in \mathbb{R}^n$ that capture salient image features. The on-line component, driven by capacitive touch input, interprets sensor data via an *Event Manager* into symbolic gestures. These gestures, along with inter-canvas communication facilitated by a WebSocket server, drive the *Audio Engine* to select and play contextually appropriate sounds, yielding a dynamic sonic response tied to both local and distributed interaction. The software components are publicly available on GitHub [7].

A. off-line components

To numerically represent each image of a painting's surface, a pre-trained ResNet-50 Convolutional Neural Network is used as a static feature extractor. This is achieved by removing the network's final classification layer and freezing its parameters, which prevents fine-tuning and produces a 2048-dimensional feature vector capturing high-level visual patterns. Images are preprocessed and normalized before feature extraction. These visual feature vectors are saved, forming the basis for connecting visual and auditory domains.

For sound representation, a set of 10 low-level audio features (e.g., spectral centroid, rolloff, flux, bandwidth, flatness, zero-crossing rate, RMS energy, tempo, and simplified attack/decay times) are extracted using the Librosa library. Each sound file is analyzed, and the resulting feature values are averaged into

a compact audio feature vector. These vectors are then normalized to the $[0, 1]$ range using min-max scaling and saved. This ensures consistent and interpretable audio descriptors.
These two distinct feature spaces are then linked by a linear transformation $T \in \mathbb{R}^{m \times n}$, which projects visual embeddings into the sound descriptor space:

$$\mathbf{y} = T\mathbf{x}$$

A linear transformation, represented by the matrix T, was specifically chosen over a more complex Multi-Layer Perceptron (MLP) due to the constraints of the available database dimension and to mitigate the significant risk of overfitting inherent with a larger model on a potentially limited dataset. The matrix T is learned by training a simple linear regression model, implemented as a `nn.Linear` layer in PyTorch. This model is trained on a unique dataset assembled by the composer, comprising 50 distinct pairs of images and their corresponding sound clips. These were carefully selected from art documentaries and audiovisual materials based on their strong aural-visual correlations. For each entry, a high-resolution screenshot of the painting's surface was chosen to serve as the visual input, and a 20-second segment of its associated audio was extracted. The training data consists of 2048-dimensional image feature vectors derived from these screenshots and their corresponding 10-dimensional normalized sound feature vectors from the extracted audio segments. Training minimizes the Mean Squared Error (MSE) between predicted and actual sound features, with an Adam optimizer iteratively refining T. For a new, unseen image, its 2048-dimensional visual feature vector is first extracted using the identical CNN pipeline. This vector is then multiplied by the loaded T matrix to yield a 10-dimensional sound feature vector in the normalized $[0, 1]$ range. These predicted features are subsequently denormalized using pre-saved parameters and clamped within plausible ranges, resulting in a robust descriptor vector. This final audio descriptor vector, serving as a compositional reference, guided the composer in a subjective and aural evaluation of the sound images. Based on the predicted qualities of each descriptor, the composer, constructed the audio files primarily using a sustained electric guitar.

B. on-line components

To capture temporal dependencies in user interaction, the Event Manager employs an adaptive high-order Markov model, where the probability of selecting a specific sound is conditioned on a variable-length history of prior gestures (both local and remote). This statistical model is updated online, allowing the system to gradually learn typical interaction sequences during the operation of an installation.
Let $\mathcal{C} = \{1, 2, 3, 4\}$ be the set of local channel IDs (associated to the cluster of sensors on the canvas), and $\mathcal{S} = \{0, 1, \ldots, 7\}$ the set of possible sound indices per channel. Events are defined as:

- Local: (local, c, d), where $c \in \mathcal{C}$ and d is the duration in seconds.
- Remote: $(\text{remote}, \text{canvas\_id}, c)$, where $c \in \mathcal{C}$.

At time t, the interaction context is represented by the K most recent events:

$$h_t = [e_{t-K}, \ldots, e_{t-1}]$$

The system maintains smoothed transition counts $C(h_t, s)$ for each context–sound pair. The probability of selecting sound $s \in \mathcal{S}$ given context h_t is estimated using Laplace smoothing as:

$$\hat{P}(s \mid h_t) = \frac{C(h_t, s) + \alpha}{\sum_{s' \in \mathcal{S}} C(h_t, s') + \alpha \cdot |\mathcal{S}|}$$

where $\alpha > 0$ is a smoothing parameter to ensure nonzero probabilities for unseen transitions.
If the context h_t is unseen, the system recursively backs off to lower-order histories (e.g., last $K - 1$ events, etc.) or defaults to a uniform distribution over $\mathcal{S}$. On triggering a local channel c, the model:

1) Selects a sound $s \sim \hat{P}(s \mid h_t)$,
2) Plays the corresponding audio file from the folder associated with c, and
3) Updates the transition count for the observed outcome.

An optional exponential decay factor $\lambda \in (0, 1)$ may be applied to gradually reduce the influence of older transitions:

$$C(h_t, s) \leftarrow \lambda \cdot C(h_t, s) + 1$$

This adaptive model captures both *spatial dependencies* (via the mapping between channel IDs and spatialized audio) and *temporal dependencies* (via the high-order history of gestures). It also integrates *remote events* from connected canvases, enabling emergent coordination and distributed learning across multiple installations.

V. Networked Multi-Canvas Performance

To enable a distributed audio-visual performance involving multiple Sounding Canvases, we implemented a networked communication layer based on WebSockets. Each canvas acts as a client, capable of sending and receiving touch interaction events in real time. A central relay server, deployed on a lightweight cloud service, handles the routing of events between peer canvases, allowing geographically distributed artworks to respond collectively as a unified system.
The server is developed in Python using the `websockets` library, which provides native support for asynchronous, bi-directional communication over the WebSocket protocol. Asynchronous I/O is essential in this context to support multiple simultaneous client connections without blocking the main execution thread. This allows the server to scale efficiently, ensuring immediate feedback, which is crucial for fluid, real-time audiovisual interactions.
When a touch event is triggered on one canvas, a message containing metadata such as the sensor ID and timestamp is serialized in JSON format and sent to the server. The server then broadcasts this message to all other connected canvases. Upon receiving a remote touch event, each peer canvas decodes the message and integrates it into the evolving high-order Markov model, influencing the algorithm's subsequent

decisions. This decentralized feedback mechanism forms the basis for a collective soundscape that spans across physical and geographical boundaries.

The architecture enables novel interaction scenarios, including asynchronous and remote participation. For example, if one canvas is activated in Italy, peer canvases in other countries (e.g., the United States or Germany) immediately respond, allowing users in different locations to engage in a shared auditory experience. This infrastructure not only enriches the expressive potential of the individual artworks, but also serves as a social medium that enables the emergence of spontaneous and complex patterns of communication among multiple remote participants.

VI. Evaluation

The performance of the Sounding Canvas system was evaluated through comprehensive latency analysis and informal user experience assessment across its various exhibitions [10]. Latency sources encompass hardware components, software processing, and network communication. From a hardware perspective, the total input-to-output latency includes the capacitive sensing system, USB communication between the Arduino and Raspberry Pi, and audio playback. Capacitive sensor reads, facilitated by the `CapacitiveSensor` library [11] with a 1.4 MΩ resistor, contribute under 20 ms. USB communication adds negligible latency, typically below 1 ms. On the Raspberry Pi, audio playback via `pygame.mixer` (utilizing the ALSA backend) introduces approximately 50 ms due to internal buffering. In combination, the average end-to-end latency from touch to local sound output is around 100 ms, which falls within acceptable thresholds for real-time interactive systems.

Software-induced latency mainly comes from event detection and handling implemented in Python. To mitigate noise and spurious triggers, an exponentially weighted moving average (EWMA) filter is applied to incoming sensor data. This filter, defined as:

$$\text{filter\_output}[i] = \text{sensor\_value}[i] \cdot p + (1-p) \cdot \text{filter\_output}[i-1]$$

introduces a tunable programmatic delay, currently contributing an additional 30–40 ms for effective smoothing.

For networked interactions, logs from two peer Sounding Canvases, one in Rome and the other in Barcelona, indicate an average latency of approximately 50–70 ms between event transmission and reception via the server. Critically, this level of latency is manageable for the ambient, non-rhythmic nature of the performance, unlike musical styles that require strict temporal synchronization as latency tolerance thresholds vary with the type of musical interaction [12].

User satisfaction was informally evaluated during multiple public exhibitions. Visitors consistently reported a sense of natural interaction and surprise at the system's responsiveness. Anecdotal evidence and spontaneous verbal feedback, documented in the project's video archive [13], suggest that the interactive latency is imperceptible to most users. This qualitative feedback, while promising, highlights the need for a more rigorous and extensive evaluation.

VII. Discussion and Future Work

While the current implementation of the *SoundingCanvas* system successfully captures temporal dependencies in user interactions through an adaptive high-order Markov model, several limitations remain. A primary challenge for the interaction framework is its reliance on learning and assigning probabilities to specific, observed action sequences. While adaptive, this can restrict expressiveness and scalability, especially since the high-order context windows introduce sparsity in transition counts during early usage, potentially impacting both robustness and responsiveness when faced with novel or highly varied user gestures.

In terms of future directions, replacing the Markov-based model with a Recurrent Neural Network (RNN), such as Long Short-Term Memory (LSTM) or Gated Recurrent Unit (GRU) architectures, could enable richer temporal modeling and generalization to unseen gesture patterns. This would allow the system to learn continuous temporal dynamics from raw sensor data, bypassing handcrafted symbolic representations and potentially improving adaptability across users and canvases.

Future evaluations will incorporate structured user testing with questionnaires to more rigorously quantify both usability and expressivity. Beyond technical validation, we will specifically assess the system's impact on audiences and performers from a Human-Computer Interaction (HCI) perspective, as this represents the most interesting evaluation of our system's ability to foster emotional dialogues.

VIII. Conclusion

This project successfully developed and implemented the Sounding Canvas, demonstrating a novel approach to interactive art that is inherently designed for networked environments. Our system significantly contributes to the themes of IoS 2025 by showcasing how networked canvases can form a tangible, intuitive interface for dynamic sound generation across distributed spaces. The robust hybrid hardware-software architecture, which seamlessly integrates real-time sensor processing with advanced networked communication protocols, highlights the profound potential for new forms of human-computer interaction within collaborative artistic installations. By enabling touch events on one canvas to influence the sonic landscape of others, this work underscores the significance of interdisciplinary approaches in creating truly immersive and engaging distributed experiences that transcend traditional sensory boundaries through pervasive networked sensing.

Acknowledgments

We extend our sincere gratitude to the Centro di Ricerche Musicali in Rome, whose rich tradition of interactive artworks provided invaluable inspiration and a supportive environment for this endeavor.

REFERENCES

[1] P. Vogel, *Peter Vogel: Interactive Electronic Art*, P. Weibel, Z. C. for Art, and M. Karlsruhe, Eds. Hatje Cantz, 2011.

[2] S. G. Lasserre and A. met den Ancxt), "Akousmaflore," https://www.scenocosme.com/akousmaflore_e.htm, 2007, accessed: 2025-05-28.

[3] ——, "Kymapetra," https://www.scenocosme.com/kymapetra_e.htm, 2016, accessed: 2025-05-28.

[4] I. Costabile, "Sound canvases: A cross-modal installation for touch interaction," in *Proceedings of the International Computer Music Conference (ICMC)*, Daegu, South Korea, 2018. [Online]. Available: https://www.researchgate.net/publication/327070353_Sound_Canvases_A_Cross-Modal_Installation_for_Touch_Interaction

[5] A. Lind, "The web: Collaborative music-making in a large-scale interactive installation," in *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, 2016, pp. 385–386. [Online]. Available: https://www.nime.org/proceedings/2016/nime2016_paper0073.pdf

[6] L. Ciamarone, "Sounding Canvas Project Report," https://luciamarock.github.io/Projects/technicalities/tech_report.html, 2024, accessed: 28 May 2025.

[7] ——, "SoundingCanvas GitHub Repository," https://github.com/luciamarock/SoundingCanvas, 2024, accessed: 28 May 2025; GitHub repository.

[8] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[9] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[10] L. Ciamarone, "Sounding Canvas Exhibitions," https://luciamarock.github.io/news/soundingcanvas.html, 2024, accessed: 28 May 2025.

[11] P. Stoffregen, "CapacitiveSensor Library," https://github.com/PaulStoffregen/CapacitiveSensor?tab=readme-ov-file, 2016, accessed: 28 May 2025; GitHub repository.

[12] C. Rottondi, C. Chafe, C. Allocchio, and A. Sarti, "An overview on networked music performance technologies," *IEEE Access*, vol. 4, pp. 8823–8843, 2016.

[13] D. Motèque, "Sounding Canvas YouTube Playlist," https://www.youtube.com/playlist?list=PLWCEF9zFjxsRQh7xD3kHHybqgGCL4OYLn, 2024, accessed: 28 May 2025; YouTube playlist.