Projected Realities: A Dialogue between Paint and Code

Bianca Gauthier

40180637

CART 398- A-2252- Special Topics In Computation Arts

Dr Gabriel Vigliensoni

Concordia University

December 11, 2025

1. Abstract

Projected Realities: A Dialogue Between Paint and Code investigates the intersection of traditional studio art and interactive computational media through embodied interaction and machine learning. A hand-painted background provides the fixed physical environment, while a projected digital character and responsive soundscape dynamically respond to full-body and hand gestures.

The system combines PoseNet and MediaPipe Hands for real-time pose recognition, streaming data via OSC into Max/MSP for sound synthesis, with p5.js managing animation control. Machine learning models are trained on a custom "pose language" inspired by iconic anime gestures, specifically *Sailor Moon*, chosen for their clarity, recognizability, and reproducibility, while still allowing original stylistic interpretation in both the character and background. These discrete pose classifications trigger character behaviours and audiovisual responses, while continuous gesture data modulates movement and sound parameters. This report details the construction of a user-generated dataset, the iterative training process, and the integration of tracking, ML inference, and audiovisual output, highlighting how artistic decisions informed the machine learning design.

Challenges encountered include pose misclassification, ambiguity, and hardware limitations, which influenced both design and interaction dynamics. Despite these constraints, the project demonstrates a cohesive fusion of analog and digital media, emphasizing the expressive potential and interpretive nature of embodied machine learning. Future work could expand pose libraries, incorporate multi-hand tracking, and introduce richer environmental interactivity, offering broader creative possibilities for performative installations.

2. Introduction

Context

Interactive installations that respond to human gestures have become increasingly prevalent in contemporary media art and HCI research. This project situates itself at the intersection of real-time body tracking, machine learning, and animation, using computer vision techniques to interpret human movement and map it to a projected animated character. Unlike pre-programmed animations, the system allows a dynamic and embodied interaction, giving users direct agency over a character's movement in a virtual platformer environment.

The fusion of traditional artistic practices with computational techniques is increasingly explored in contemporary media art. By combining hand-painted visual elements with live, responsive projections controlled via machine learning, this project bridges the analog and digital The project draws inspiration from early animation techniques, where characters and backgrounds were created on separate layers, allowing for dynamic interactions between moving and static components. By integrating embodied interaction through gesture-based ML, the system allows participants to influence the behaviour of a projected character, thus mediating a dialogue between physical and digital forms.

Although this report focuses on the conceptual and technical development of Projected Realities within the context of CART398, the project was conceived from the beginning as a single integrated system that simultaneously fulfilled the goals of both CART398 and CART346. In parallel with the animation system, sound was incorporated as part of the interactive feedback loop. Because Max/MSP acted as the central hub for receiving OSC messages from the browser and interpreting pose data, it also enabled real-time sonic responses to gestures. These audio behaviours supported the sense of embodiment by providing an additional modality through

which users could perceive how the system interpreted their movements. Even in its early stages, the project explored how sound could reinforce or contrast the characters' on-screen actions, contributing to the overall responsiveness and atmosphere of the installation.

Literature Review

Existing literature in interactive art demonstrates the potential of user movement to inform projected visuals, creating co-authored experiences. For example, Lozano-Hemmer's *Body Movies* (2001) utilized participants' shadows as real-time triggers for visual projection, creating an interplay between human presence and algorithmic output. Similarly, Chung's *Drawing Operations Unit* (2017) explored the co-creation of imagery between humans and robotic systems, emphasizing the relational aspect of interactive drawing. These precedents informed the design of *Projected Realities*, which extends prior work by combining pose-based ML with narrative and character-driven interactions, rather than abstract visuals alone.

Recent research in embodied machine learning highlights the challenges of real-time pose estimation and gesture recognition. Small, curated datasets allow for quick iterative training but require careful curation to ensure pose distinctiveness and robustness against environmental variation. This project situates itself within these frameworks, applying ML as an interface for performance and play, rather than as a purely autonomous creative agent.

Significance

The significance of Projected Realities lies in its reactivation of historical animation practices through contemporary interactive and machine learning–driven techniques. Early animation workflows relied on the separation of static backgrounds and animated characters, a process that foregrounded layering, timing, and gesture as central expressive elements. This project revisits that methodology by pairing a hand-painted, fixed environment with a digitally

projected character whose movement is governed by embodied user input, reassembling traditional processes through computational means rather than replacing them.

By integrating pose-based machine learning into this structure, the project positions ML not as an autonomous creative force, but as a mediating system between human gesture and audiovisual expression. Audience movement becomes a form of authorship, transforming bodily action into animation, sound, and narrative progression. Rather than striving for perfect recognition or seamless automation, the system embraces moments of ambiguity, misclassification, and instability as expressive outcomes, revealing the interpretive nature of machine learning systems and their dependence on human context.

The work is also significant in how it merges multiple artistic media, painting, animation, sound, performance, and code into a cohesive experiential installation. Traditional studio art practices coexist with real-time computation, inviting viewers to inhabit a liminal space between analog and digital creation. Through embodied interaction, participants are not simply controlling a character, but actively negotiating with the system's limitations and affordances, reinforcing the idea of interaction as a shared process of meaning-making.

In doing so, Projected Realities contributes to ongoing discussions within interactive and computational art regarding authorship, embodiment, and the role of technology in creative practice. The project demonstrates how machine learning can be meaningfully situated within artistic workflows, not as an end in itself, but as a tool that enables dialogue between past and present media, human intuition and algorithmic interpretation, and static imagery and living motion.

The sonic dimension of the project further emphasized the interpretive qualities of the machine-learning system. Because audio was generated from the same pose predictions that

drove the animation, the system produced a unified audiovisual expression of each gesture. Subtle classification errors, timing shifts, or exaggerated poses became audible as well as visible, allowing sound to function as both an expressive layer and an extension of the embodied interaction. This integration contributed to the installation's sense of presence and heightened the connection between user movement and computational output.

Objectives

The primary objectives of the project were to:

- Design a custom pose language suitable for embodied interaction
- Train a real-time pose classifier using small, user-generated datasets
- Integrate pose classification into an audiovisual installation
- Explore ambiguity, instability, and interpretation in embodied ML systems
- Maintain conceptual continuity between traditional painting and digital projection

3. Project Description

Vision for Embodied Interaction

The conceptual foundation of *Projected Realities* is the creation of an interactive platform where users engage directly with a projected character via gestures and poses. This embodied interaction encourages participants to think critically about how their physical movements influence digital behaviour, effectively becoming co-authors of the projected narrative. By assigning distinct poses to specific character actions, such as walking, crouching, climbing, or jumping, the system translates subtle human motion into digital animation, reinforcing the interplay between intention and outcome.

The installation combines narrative, visual, and auditory elements. The character's design draws inspiration from the magical girl aesthetic of *Sailor Moon*, with simplified, recognizable poses adapted to the constraints of real-time pose tracking. The pastel colour palette and cartoon-inspired visuals ensure visibility and cohesiveness when projected onto a painted canvas. Each gesture triggers not only movement but also audio feedback, emphasizing the multimodal interaction between participant and installation.

Technologies Used

Software:

- **p5.js:** Handles animation and integrates with pose data from the ML model. Provides frame-based control for smooth character motion and easing functions for natural movement.
- **Max/MSP:** Manages audio synthesis and processes OSC messages to respond to character actions. It allows the system to provide audio feedback, reinforcing participant engagement.
- **Node.js with WebSocket & OSC libraries:** Bridges browser-based ML pose detection with Max/MSP for bidirectional communication. Ensures real-time synchronization between gestures, animation, and sound.
- **PoseNet + MediaPipe Hands:** Provides real-time pose estimation and hand tracking. These frameworks offer high accuracy while being lightweight enough to run in browser environments, supporting rapid prototyping.

Max/MSP played a central role not only in routing OSC data between the browser and the animation system but also in generating audio responses to gesture data. The patch received normalized PoseNet and MediaPipe landmarks and used them to drive sound synthesis

processes, including FluCoMa-based classification that mapped feature vectors to sonic states. The same OSC messages that animated the character in p5.js also informed sound behaviours, ensuring that visual and sonic responses remained synchronized during interaction.

Hardware:

- **Webcam:** Captures participant gestures and serves as input for the ML model. Optimal placement and lighting were critical for accurate pose recognition.
- **Projector:** Projects the animated character onto a hand-painted background, overlaying the digital and analog elements seamlessly.
- **Painted Canvas:** Functions as the physical environment and visual reference for the participant. The design includes distinct zones and visual cues to encourage interaction.

Machine Learning Implementation

The core ML component involves classification: predicting the participant's pose based on keypoints extracted from PoseNet and MediaPipe Hands. A custom "pose language" maps each gesture to a character action, inspired by iconic *Sailor Moon* postures. For example, one pose with hands raised above the head triggers a jump, while a side-step motion initiates lateral movement. Each pose was selected for clarity, recognizability, and ease of translation to animation.

Training the model required iterative testing and careful data collection. The initial dataset consisted of hundreds of frames per pose, captured with a consistent camera angle and lighting. However, misclassification was frequent due to subtle variations in participant posture. To mitigate this, MediaPipe Hand keypoints were incorporated to improve accuracy in distinguishing poses with similar torso positioning but differing hand orientations. Incremental

training allowed continuous improvement: first stabilizing basic walking and idle poses, then incorporating complex movements such as crouch, climb, and jump.

The FluCoMa classifier inside Max/MSP was developed concurrently with the browser-based PoseNet/MediaPipe classification. Both systems worked together as part of the same ML pipeline, with Max receiving streams of pose and hand landmarks and using them for additional audio-responsive classification. The inputs to FluCoMa were the same normalized feature vectors extracted from PoseNet and MediaPipe, and the output labels were used within the same patch to drive both sound behaviour and outgoing messages to p5.js. This ensured that ML inference, audio response, and character movement all originated from a single unified dataset and training workflow.

Training Data

- **Collection Method:** Participants performed gestures in front of a webcam, producing labelled data for training. Variations in height, arm angle, and stance were intentionally included to improve generalization.
- **Dataset Details:** Each pose had 150–300 frames, with 2-3 participants contributing. Data augmentation techniques, including slight rotations and scaling, were applied to increase model robustness. This small, curated dataset allowed fast training and real-time testing, aligning with the interactive ML approach.

System Integration

The system integrates ML, animation, and sound in real-time. The workflow follows a clearly defined pipeline:

- Participant performs a gesture in front of the webcam.
- PoseNet and MediaPipe extract body and hand keypoints.
- Pose data is sent to the browser via the OSC WebSocket bridge.
- p5.js interprets the predicted pose label and updates the character animation.
- OSC messages are dispatched to Max/MSP to trigger corresponding audio events.
- The animated character and audio are projected onto the hand-painted environment.

This pipeline ensures low latency and consistent behaviour. Fail-safes handle missing keypoints and network interruptions, allowing graceful degradation of system performance.

Because both the animation system and the sound engine were driven by the same classification data, the installation maintained a high degree of audiovisual consistency. Every predicted pose influenced character behaviour in p5.js and simultaneously shaped the sonic output generated in Max/MSP. This shared control structure created a coherent interaction model in which sound reinforced on-screen actions and helped communicate the machine-learning system's internal state to participants.

Code Highlights

OSC Data Reception:

```
socket.onmessage = async (event) => {
    try {
        let data = event.data;
        if (data instanceof Blob) data = await data.text();

        const msg = JSON.parse(data);

        // LISTEN FOR /predictpoint
        if (msg.address === "/predictpoint" && msg.args.length > 0) {
            const label = msg.args[0];
            console.log("🎯 Received pose prediction:", label);
```

This snippet illustrates real-time reception of pose predictions and dispatch to p5.js for animation updates, enabling fluid interaction. It is waiting to receive an OSC message from Max. If the message comes with the address /predictpoint, then it will read the float that it sends as a label for the pose.

Character Movement:

```
//------------------------------------------------------------
// Move character from pose
//------------------------------------------------------------
moveFromPose(label) {
    if (this.isJumping || this.jumpDownTarget !== null || this.isClimbing) return;

    switch (label) {
        case "walk_left": this.x -= this.speed; break;
        case "walk_right": this.x += this.speed; break;
        case "walk_front": this.y += this.speed; break;
        case "walk_back": this.y -= this.speed; break;
    }
}
```

This function demonstrates how each pose directly influences character movement, ensuring that gestures translate into immediate and smooth animation. After the OSC message is received, it will then say that, based on the label of which pose is being read live, the character will act accordingly. This is how I was able to manipulate/train this output.

Pose Handling:

```
//------------------------------------------------------------
// Handle incoming OSC pose
//------------------------------------------------------------
changePose(label, barriers = null) {

    if (!Array.isArray(barriers)) barriers = [];

    // Jump-down mechanic
    if (label === "jump" && barriers && this.triggerJumpDown(barriers)) return;

    // Normal jump
    if (label === "jump" && !this.isJumping) {
        this.isJumping = true;
        this.jumpVelocity = this.jumpStrength;
        this.currentAnimation = "jump_up";
        return;
    }
```

Special actions like jump and climb are handled with collision awareness, ensuring the character behaves realistically within the virtual environment. To add complexity to the character actions, I made it that if the label for the pose ‘jump’ is read, the character will jump up, but if, in addition to the label reading ‘jump’, the character is also in contact with the top of this barrier, then the character will jump down to the other top of the building. This is how the way i found to limit areas that the character cannot interact with without having the incoming OSC message affect/crash my character.

4. Process

The development of *Projected Realities: A Dialogue Between Paint and Code* unfolded as an iterative negotiation between concept, visual design, and machine learning implementation. Rather than following a strictly linear workflow, the project evolved through cycles of testing, adjustment, and reflection, where technical limitations often reshaped aesthetic and conceptual decisions. This section outlines the project’s development stages, focusing on conceptual and visual refinement, machine learning implementation, challenges encountered, and key learning outcomes.

Conceptual Refinement and Visual Development

From its initial proposal, the project was grounded in an interest in early animation techniques, particularly the separation of background and character into distinct layers that are later recombined. A hand-painted background provided a static, physical canvas, while the digital character was dynamic and responsive.

While the early conceptual inspiration was broad, encompassing multiple forms of animated media, the development process revealed the need for greater specificity, particularly once the idea of creating a custom pose language emerged. The project’s interactive framework

required gestures that were not only expressive but also distinct, repeatable, and easily interpretable by both humans and machine learning models. Early cartoons, while influential aesthetically, often rely on physical gags and exaggerated motion rather than discrete poses, making them less suitable for pose-based classification. Anime, by contrast, offered a visual language built around iconic, symbolic gestures. Among several references considered, *Sailor Moon* emerged as the most appropriate primary inspiration. Its transformation and attack poses are highly recognizable, visually simple, and heavily focused on hand and arm positioning. Importantly, these poses function almost as visual symbols, qualities that translate well into the idea of a pose-based "vocabulary" for machine learning interaction.

Both the character and background were designed specifically for this project. While they intentionally reference the visual style and colour palette associated with *Sailor Moon*, the aim was not replication but recognizability. Allowing aspects of my own artistic style to surface was an important part of the process, reinforcing the project's position as an interpretation rather than a remake. The background was designed using a platformer-style layout, which provided a clear sense of ground, obstacles, and movement pathways. This spatial organization supported not only animation logic and audience readability, but also machine learning testing, as pose-triggered movement could be evaluated within a predictable environment.

Machine Learning and Pose Language Development

Machine learning development progressed through repeated cycles of pose design, data collection, training, testing, and refinement. From the outset, the intention was not to apply machine learning as a background technical layer, but to foreground it as an embodied, performative interface, one in which the participant's body becomes a readable system interpreted in real time.

Early experimentation focused on full-body pose recognition using PoseNet. Initial pose selections were inspired directly by *Sailor Moon*, including larger, full-body stances. However, early testing revealed instability when participants stood farther from the camera. Body landmarks became unreliable, especially under variable lighting conditions, and classification frequently degraded during transitions between poses. In response, the interaction design shifted toward upper-body framing, allowing for increased consistency and more reliable tracking.

This shift aligned well with the chosen anime reference, as many *Sailor Moon* poses emphasize hand and arm configurations rather than complex lower-body movement. A custom pose language was developed in which specific gestures were mapped to character actions. Care was taken to create intuitive relationships between bodily configuration and resulting behaviour, enabling participants to learn the system through experimentation rather than instruction.

Initial machine learning models relied solely on PoseNet data, using a small feature set of approximately six to eight values. While sufficient for static poses, this approach proved fragile during live movement. The classifier frequently jumped between states mid-transition, producing jittery animation and inconsistent control. To address this, MediaPipe Hands was integrated via OSC, introducing detailed hand landmark data. Because many poses were hand-centric, this significantly increased expressive resolution.

As the pose language evolved, the sound layer played a useful role in evaluating the clarity and distinctiveness of poses. Because audio responded immediately to classification outputs, inconsistencies or ambiguities in training data often became audible before they were visually apparent. This real-time feedback helped refine pose examples and feature selection, ultimately contributing to the stability of both the animation and audio systems.

To manage system complexity and performance, tracking was limited to one hand, resulting in a feature set of 52 continuous values. Training was conducted using small, user-generated datasets collected in real time. As more features were introduced, training time increased, requiring careful normalization and consistent capture conditions. Poses were added incrementally, with problematic gestures, such as climbing, temporarily removed or redesigned when instability persisted.

Challenges and Solutions

Several challenges emerged throughout development, highlighting the fragility of embodied machine learning systems in installation contexts.

- **Pose Ambiguity** was a recurring issue. Gestures that occupied similar spatial regions, particularly during transitions, were frequently misclassified. This was addressed through pose redesign, increased emphasis on hand orientation, and staged training approaches that reduced the number of active poses at any given time.
- **Environmental sensitivity** significantly affected tracking reliability. Changes in camera height, angle, distance, and lighting between training and presentation environments introduced instability, especially for hand detection. These issues were mitigated by adjusting camera placement, refining pose definitions, and prioritizing upper-body framing to maintain consistency.
- One challenge that emerged during testing involved balancing the sensitivity of the classifier with the responsiveness of the sound system. Rapid shifts between similar poses could cause audio to retrigger too quickly, producing unintended bursts or unstable textures. To address this, additional smoothing, debouncing, and scaling techniques were applied so that sonic events responded clearly without overwhelming the listener. This

process revealed how closely intertwined sound and visual classification were and informed adjustments to the training dataset and pose definitions.

- **System stability** posed additional challenges. Dense OSC streams and rapid classification changes occasionally caused Max/MSP to freeze. This required limiting feature sets, reducing classification frequency, and restructuring patch logic to maintain responsiveness.
- **Projection alignment** introduced further complexity, as variations in projector resolution affected spatial consistency between the painted background and digital character. This was addressed through calibration and iterative testing during installation setup.

Rather than treating these obstacles purely as errors to eliminate, they became central to understanding how machine learning behaves within physical, performative contexts.

Successes and Learning Experiences

Unexpected system behaviours became some of the project's most compelling elements. When tracking was lost or gestures became ambiguous, the system often continued outputting the last recognized pose, causing the character to move autonomously. Participants were required to "override" the machine's interpretation through deliberate bodily action, creating a playful tension reminiscent of platformer video game mechanics.

Similarly, unplanned sonic and visual artifacts, such as slowed movement or eerie motion, aligned with the ghost-like presence of the projected character. What initially appeared as a technical failure gradually became an aesthetic texture, reinforcing the project's themes of instability, mediation, and shared agency.

Through this process, I gained a deeper understanding of machine learning not as a neutral tool, but as an interpretive system shaped by approximation, bias, and environmental context. The project highlighted how embodied interaction, visual design, and technical constraint are inseparable, and how meaningful engagement often emerges through friction rather than seamless control.

Ultimately, *Projected Realities* became a study in negotiation, between body and machine, intention and outcome, control and uncertainty, revealing machine learning as an expressive collaborator rather than a purely deterministic system.

5. Future Work

Potential enhancements include:

- Expanding the pose library and incorporating multi-hand tracking for more nuanced gestures.
- Implementing interactive environmental objects that respond dynamically to character or participant actions.
- Exploring projection mapping techniques to overlay digital content on complex, textured surfaces.
- To maybe work with the concept that the character continues to move without stopping, therefore if I develop it more as a game, it could have many areas where the character dies or is reset, therefore the audience needs to get in control to make the character evade such obstacles (not the main plan, just one of the options)
- Create more backgrounds. I would like to expand, with other backgrounds and more interactions that the character can have with the elements on the painted background, to strengthen the merging element.

- -Collision-based sonic states, such as introducing glittery or sparkly glitch effects when the character collides with or impacts the environment
- -Expanded sampler behaviours for special moves
- Additional future developments could include spoken-word fragments and spatial audio integration, as well as expanding the work into a room-scale installation format.

6. References

(Grammar, sentence structure, and repetition checked by Grammarly)

Cabannes, Vivien, et al. 2019. *Dialogue on a Canvas with a Machine.*

Chung, Sougwen. 2017–ongoing. *Drawing Operations Unit.*

Ikeda, Ryoji. 2019. *Data-verse.*

Lozano-Hemmer, Rafael. 2001. *Body Movies.*

Ridler, Anna. 2019. *The Fall of the House of Usher II.*

FluCoMa Project. 2020. *Fluid Corpus Manipulation Toolkit.*
ml5.js. 2023. *Friendly Machine Learning for the Web.*
Google. 2023. *MediaPipe Hands.*

Audio where I downloaded the audio tracks:

SailorMusic. n.d. “Transformation Music.”
SailorMusic.https://sailormusic.net/transformation-music/

7. Appendix

Pose Language Diagram

Here is the documentation, research document for finding and choosing the poses from Sailor Moon:

https://docs.google.com/document/d/1qW45K0zsDRwQmHtGUPkxl5obun2PQx_fM_1K1ooM-G8/edit?tab=t.0