# Beyond the Grasp: Extending Embodied Projected Mixed Reality with Volumetric Gestures and Novel Applications

Reggie Segovia

December 2025

# Acknowledgments

The deepest gratitude is extended to my thesis advisor, Dr. Alexandre Gomes de Siqueira. His expertise allowed for this project to not only be completed but to spark the motivation to continuously pursue meaning and improvements within my work. Our collaboration on the original platform "Graspable Memories," as well as a plethora of other projects at the University of Florida, has provided clarity as I explore my ambitions toward human-computer interaction, ultimately helping me to become a better academic and person.

I would also like to extend this gratitude to the two other members of my thesis committee, Dr. Neha Rani and Dr. Michael Bumbach. Their combination of deliberate comments and outside perspectives truly refined what this research has become, while providing the forward momentum to understand where the work can be framed within the context of the field and beyond.

My wife, Yilin Wang, has served as the epicenter of all of these formative years of my academic and research experience. Her unwavering support has pushed me to better myself in every way, regardless of the circumstances, serving as the backbone to every endeavor I take on. Without her, I would never have been able to accomplish everything I had set out to do during my time at the University of Florida. My family's support similarly demands recognition, providing support in every way possible as I pursue my educational ambitions.

This thesis serves not only as an expansion of the Embodied Projected Mixed Reality idea presented during the "Graspable Memories" project, but also as an insightful glance into the collaborative interface this platform can ultimately become. It further performs as a tribute to those who have been by my side while supporting and believing in me every step of the way.

**Abstract**

This thesis extends the Embodied Projected Mixed Reality (EPMR) paradigm introduced in "Graspable Memories," transforming it from a personal memory interaction system into a versatile platform for collaborative design and creative interaction. While the original prototype successfully demonstrated how bodily occlusion could become an intentional interface element, it was limited to discrete hand poses and a narrow range of gestures. This work addresses these limitations by developing a fully volumetric interaction model that integrates spatial depth sensing, continuous 360-degree hand rotation, hand tilt detection, and fine-grained pinch gestures for precise UI control. Additionally, I explore the expansion of projection surfaces beyond the hand through custom-trained object detection models, enabling projection onto environmental objects such as trays. These technical enhancements are evaluated through novel application domains that demonstrate the platform's potential for collaborative design workflows, including slider-based parameter adjustment and multi-surface interaction. By moving from personal reflection to general-purpose spatial interaction, this research establishes EPMR as a foundation for human-centered, spatially-aware experiences that meaningfully bridge physical and digital worlds.

**Keywords:** Human-Computer Interaction, Spatial Augmented Reality, Embodied Interaction, Volumetric Gestures, Computer Vision.

# 1 Introduction

The evolution from analog to digital media has fundamentally altered how we interact with information. While digital content offers unprecedented convenience and accessibility, it often lacks the tangible, embodied qualities that make physical interaction intuitive and meaningful. The Graspable Memories project introduced Embodied Projected Mixed Reality (EPMR) as a paradigm that addresses this tension by treating bodily occlusion not as interference to be minimized, but as an intentional interaction mechanism [8].

In the original EPMR implementation, users could select projected photographs by covering them with an open palm, move these images through space as they followed the hand, and transfer them to a wearable pendant display through a closing gesture. This approach successfully demonstrated how projection-based systems could restore physicality and ritual to interactions with digital memories while maintaining sustainability benefits over physical photo printing.

However, the initial prototype revealed significant limitations. The interaction model relied on discrete hand states—open palm, closed fist, or rotated hand—without leveraging the rich, continuous three-dimensional movement capabilities of human hands. The system tracked only coarse hand orientation, missing opportunities for nuanced control through spatial positioning, fine-grained rotation, or precise finger movements. Moreover, interaction was confined to the hand as the sole projection surface, overlooking the potential for environmental objects to serve as dynamic interface elements.

This thesis addresses these limitations by developing a comprehensive set of technical enhancements that transform EPMR from a proof-of-concept memory system into a general-purpose platform for spatially-aware interaction. My contributions include:

- **Volumetric gesture expansion**: Integration of spatial depth as a continuous control input, full 360-degree hand rotation (yaw), and hand tilt (pitch and roll) to enable nuanced manipulation of projected content
- **Fine-grained gesture recognition**: Implementation of pinch gestures for precision interaction with small UI elements, including functional slider controls
- **Multi-surface projection**: Development of custom landmark detection models to enable projection onto arbitrary objects (demonstrated with a tray), expanding the interaction space beyond the hand
- **System optimization**: Performance improvements addressing tracking latency, computational efficiency, and projector-camera parallax correction
- **Novel applications**: Demonstration of EPMR's potential for collaborative design workflows and parameter adjustment interfaces

These enhancements fundamentally shift EPMR's scope from personal, reflective engagement with memories to a versatile interaction paradigm suitable for creative collaboration, design exploration, and spatial manipulation of digital content. This work establishes EPMR as a platform that bridges embodied interaction research with practical applications in human-centered computing.

The remainder of this thesis is organized as follows: Section 2 reviews the foundational Graspable Memories work and positions the current contributions within the broader context of tangible and projected interfaces. Section 3 details the technical implementation of volumetric gestures and multi-surface projection. Section 4 presents the novel applications enabled by these enhancements. Section 5 discusses implications for interaction design and evaluates the system's capabilities. Section 6 concludes with reflections on future directions for EPMR as a general-purpose interaction platform.

# 2 Background and Related Work

## 2.1 Graspable Memories: Foundation of EPMR

The Graspable Memories project established EPMR as a paradigm that reconceptualizes projected interfaces from flat, static surfaces to volumetric interaction fields where the body becomes the primary interface element [8]. Unlike traditional Spatial Augmented Reality (SAR) systems that treat bodily occlusion as visual interference to be minimized [3], EPMR embraces occlusion as a meaningful gesture that transforms the hand into both an input mechanism and a display surface.

The original system employed four core interaction cues: hand region (palm, back, fingertips), postural configuration (open, closed, rotated), spatial depth (though limited), and intentional occlusion. A user study with twelve participants used the User Experience Questionnaire [18] to evaluate the system, demonstrating strong perceived novelty (UEQ mean = 1.75) and attractiveness (mean = 1.65), with participants reporting that gestures felt intuitive and emotionally resonant. The act of "grasping" a photograph and transferring it to

a wearable pendant was described as "special" and "satisfying," successfully restoring ritual and presence to digital memory interaction.

However, the evaluation also revealed limitations. The Efficiency dimension scored in the neutral range (mean = 0.60), reflecting both technical constraints in hand tracking and the inherently contemplative pacing of the interaction. Participants noted that while the metaphor of grasping worked conceptually, it was "not fully intuitive or truly comparable to a real-world grasp." These observations motivated the technical work presented in this thesis.

## 2.2 Projected Interfaces and Volumetric Interaction

Projection-based systems have long explored how to augment physical environments with digital content. Early work by Raskar et al. introduced Shader Lamps [25], which used projectors to animate physical objects through image-based illumination. This pioneering approach demonstrated that projectors could transform neutral objects by "lifting" their visual properties into the projected imagery. Building on this foundation, Bimber and Raskar's comprehensive treatment of Spatial Augmented Reality [3] established projection as a viable alternative to head-mounted displays, treating the environment itself as the display surface.

More recent systems have pushed toward interactive projection. The MirageTable [1] combined depth sensing with stereoscopic projection to enable freehand interaction with projected 3D content on tabletops. Users could manipulate virtual objects through physically realistic gestures without gloves or trackers, demonstrating that projection-based systems could support rich 3D interaction. Jones et al.'s RoomAlive [15] scaled this concept to entire rooms using distributed projector-camera units, creating immersive experiences that dynamically adapted to room geometry and user position.

However, these systems generally treated the user's body as an actor within the augmented space rather than as the interface itself. Systems like OmniTouch [10] and Skinput [11] began to challenge this distinction by projecting directly onto the body and sensing touch input on skin. OmniTouch enabled users to appropriate everyday surfaces—including their own hands and arms—as interactive displays, while Skinput used bio-acoustic sensing to detect finger taps on the arm. LumiWatch [35] further miniaturized this concept into a smartwatch form factor with on-arm projection capabilities.

Yet even these body-centric systems treated occlusion primarily as a technical challenge. Kim et al. [16] developed sophisticated shadow removal techniques for front-projection systems, while earlier work on occlusion compensation [4] sought to minimize the visual disruption caused by users blocking projected content. In contrast, EPMR repositions occlusion as an intentional interface event—a design philosophy that transforms what was previously considered interference into meaningful interaction.

### 2.2.1 Gesture Recognition and Hand Tracking

The technical feasibility of EPMR relies heavily on advances in hand tracking and gesture recognition. MediaPipe Hands [36] provides real-time detection of 21 hand landmarks using lightweight convolutional neural networks optimized for mobile and embedded devices. This

enables the precise tracking of individual fingers, palm orientation, and hand pose without specialized hardware beyond an RGB camera.

Prior work on gesture design has established important principles for spatial interfaces. Wobbrock et al.'s study on user-defined gestures [33] revealed that users naturally prefer gestures that physically resemble their effects—for instance, grasping gestures for selection. This aligns with EPMR's occlusion-based interaction, where covering an object with the hand serves as a natural metaphor for taking possession of it. However, as gesture vocabularies expand, designers must balance expressiveness with discoverability and avoid overwhelming users with complex command sets.

Depth-sensing cameras have proven particularly valuable for projection-based interaction. Wilson demonstrated how depth cameras could detect touch on arbitrary surfaces [32], while Izadi et al.'s KinectFusion [14] showed how moving depth cameras could reconstruct 3D environments in real time. These techniques inform EPMR's approach to tracking hand position in 3D space and understanding the geometric relationship between hands, projected content, and physical surfaces.

## 2.3 Tangible and Multi-Surface Interaction

EPMR's conceptual foundations draw from decades of research in tangible and embodied interaction. Ishii and Ullmer's seminal Tangible Bits framework [13] articulated a vision of "seamless interfaces between people, bits and atoms," where digital information could be manipulated through physical form. While traditional tangible interfaces embed computation in dedicated physical objects, EPMR explores how projection can create ephemeral tangibles that appear and disappear as needed.

Dourish's philosophical treatment of embodied interaction [6] provides crucial theoretical grounding. He argues that meaning in interaction emerges from how we act in the world, not merely from symbolic representations. EPMR embodies this perspective by treating hand movements—reaching, grasping, rotating—as inherently meaningful actions rather than abstract commands. The system responds to bodily engagement with projected content, making interaction feel direct and spatially situated.

More recent work has explored hybrid approaches that blend physical and virtual elements. De Siqueira et al. introduced the concept of "hard and soft tangibles" [5], combining physical tokens with multi-touch surfaces for scientific visualization. Their work demonstrates that different modalities can complement each other—physical objects providing tactile feedback and spatial stability, while virtual elements offer flexibility and dynamic reconfiguration. This philosophy extends to EPMR's pendant, which serves as a reusable physical interactor with a reconfigurable digital display, bridging tangible and virtual interaction modes [27].

Hornecker and Buur's framework for tangible interaction [12] emphasizes the importance of spatial relationships and embodied constraints. They identify four key themes: tangible manipulation, spatial interaction, embodied facilitation, and expressive representation. EPMR directly engages with spatial interaction by extending the interface from flat surfaces into volumetric space, treating the hand's position and orientation in 3D as continuous input parameters rather than discrete states.

The recent critical examination of embodiment in TEI by Spiel [26] raises important

questions about whose bodies are assumed as normative in interface design. This critique reminds us that hand morphology, gesture capabilities, and cultural associations with bodily interaction vary across individuals. EPMR must be designed with this diversity in mind, supporting adaptation to different hand sizes, gesture styles, and interaction preferences.

## 2.4 Digital Memory and Personal Informatics

EPMR's original application to personal memory curation connects to broader research on how digital technologies support remembering. Petrelli and Whittaker's comparison of physical and digital mementos [23] revealed that physical objects remain preferred for their tangibility and visibility in everyday spaces. People enjoy managing physical collections—sorting, arranging, and revisiting them—in ways that generic digital interfaces often fail to support. Van den Hoven's work on materializing memories [29, 30] emphasizes that external memory cues are most effective when embedded in contexts that naturally prompt remembering.

However, the shift to cloud storage has made digital possessions feel increasingly intangible. Odom et al. [21] documented how people struggle to maintain emotional connections to files stored in distant, invisible cloud systems. Similarly, Kirk and Sellen [17] found that home archiving practices involve complex negotiations around meaning, obligation, and forgetting—not just remembering. These insights motivated Graspable Memories' design, which sought to restore physicality to digital content through projection and embodied gesture.

Elsden et al.'s concept of the "quantified past" [7] highlights how wearables and personal informatics can shape memory practices through data capture and reflection. Yet as Petrelli et al. [22] demonstrated, people prefer carefully curated memory cues over exhaustive lifelogging. This selective approach informs EPMR's design: rather than automatically capturing everything, the system supports intentional selection and ritualized transfer of meaningful moments to the wearable pendant.

## 2.5 Wearable Displays and On-Body Interaction

The pendant component of EPMR builds on research into wearable displays and on-body interaction. Mann's early work on wearable computing [20] envisioned personal imaging systems that could augment perception and memory. Modern implementations have explored various form factors, from smartwatches to electronic skin. Weigel et al.'s iSkin [31] demonstrated flexible, stretchable touch sensors that could be temporarily applied to the body, while Lopes et al. explored proprioceptive interaction [19], where users feel their own body pose as output through electrical muscle stimulation.

These systems highlight trade-offs between functionality and social acceptability. Head-mounted displays provide immersive experiences but isolate users from their surroundings [9]. In contrast, projection-based interfaces like EPMR remain socially visible—others can see what the user is manipulating—supporting co-present collaboration while avoiding the awkwardness of wearing conspicuous technology [24].

## 2.6 Positioning This Work

This thesis advances EPMR beyond its initial memory-curation focus toward a general-purpose interaction platform. By integrating continuous gesture recognition, volumetric interaction, and multi-surface projection, we demonstrate that occlusion-based projected interfaces can support diverse applications requiring precise spatial manipulation. This work contributes to ongoing efforts [2, 28] to understand how emerging spatial computing technologies can create more natural, embodied interactions with digital content while remaining grounded in physical space and visible to others.

# 3 Technical Implementation

## 3.1 System Architecture

The enhanced EPMR platform utilizes a compact, unified hardware configuration designed to ensure spatial consistency between sensing and display. Unlike distributed systems or ceiling-mounted installations, our setup consists of a custom **3D-printed platform** housing both a pico projector (1920×1080 resolution) and a calibrated RGB camera. This rig is mounted above the interaction volume, ensuring that the projection frustum and the camera's field of view originate from the same relative vantage point. This co-location minimizes occlusion shadows cast by the hardware itself and simplifies the coordinate transformation between the vision system and the display canvas. The software stack is implemented in Unity with C# and leverages MediaPipe Hands.

The core architecture consists of several key components:

- `HandLandmarkerRunner`: Manages MediaPipe hand tracking integration and provides hand position/visibility queries
- `HandGestureRecognizer`: Implements gesture classification logic with support for volumetric inputs
- `MediaPipeBookController`: Orchestrates interaction between gesture recognition and projected content
- `PageImageManager` and `SelectablePageImage`: Handle projection and selection of media content
- `OSCImageSender`: Manages communication with the wearable pendant via Open Sound Control (OSC) [34]
- `HandPinchSlider`: Implements pinch-based UI control for parameter adjustment

MediaPipe Hands provides 21 3D hand landmarks following a standard anatomical model. The landmark indexing used throughout the system is:

- Landmark 0: Wrist

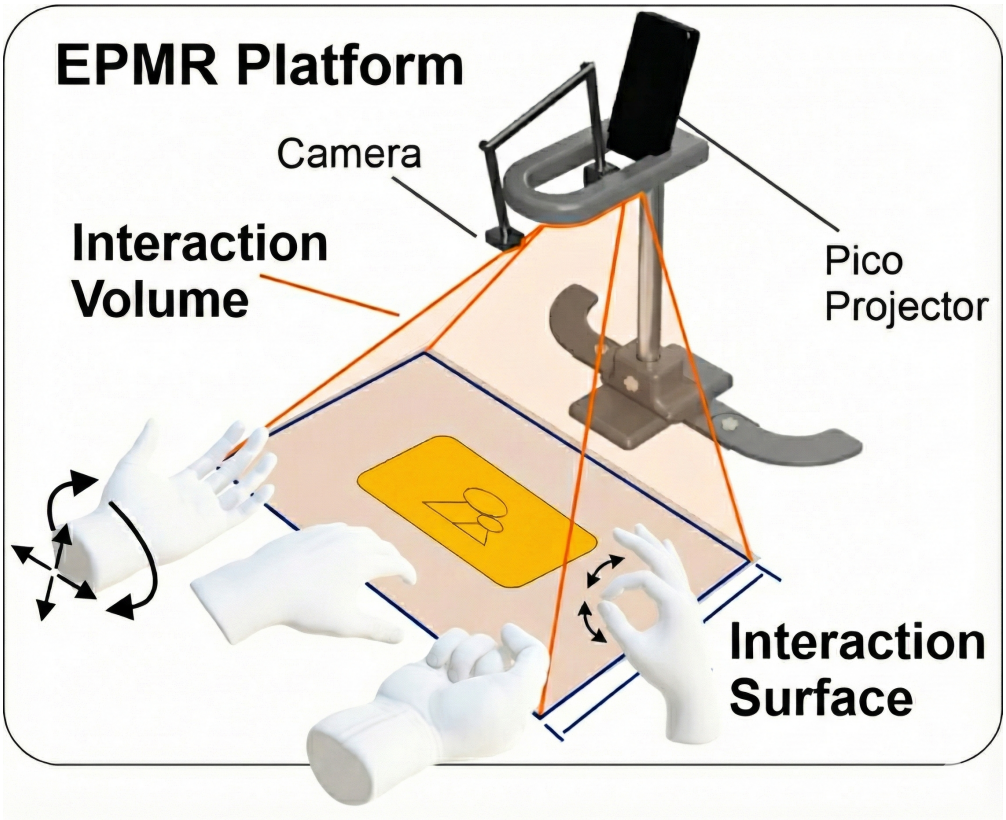


Figure 1: The EPMR hardware setup. A custom 3D-printed mount aligns the RGB camera and pico-projector, ensuring both originate from the same relative area to minimize parallax and simplify calibration.

- Landmarks 1–4: Thumb (CMC, MCP, IP, Tip)
- Landmarks 5–8: Index finger (MCP, PIP, DIP, Tip)
- Landmarks 9–12: Middle finger (MCP, PIP, DIP, Tip)
- Landmarks 13–16: Ring finger (MCP, PIP, DIP, Tip)
- Landmarks 17–20: Pinky (MCP, PIP, DIP, Tip)

The system uses the middle finger MCP joint (landmark 9) as the reference point for palm position, providing stable tracking even during finger articulation.

## 3.2 Volumetric Gesture Expansion

### 3.2.1 Enhanced Gesture Recognition

Building on the original discrete gesture states (open palm up, open palm down, closed fist), I developed an enhanced gesture recognition system that supports continuous volumetric input. The `HandGestureRecognizer` class implements a state machine that classifies hand poses based on anatomical landmarks:

```
public enum HandGesture
{
    None,
    OpenHandUp,    // Palm facing up
    OpenHandDown,  // Palm facing down
    ClosedHand,    // Fist
    Pinch,         // Thumb and index finger touching
}
```

The recognition pipeline processes MediaPipe's 21 hand landmarks through several analysis stages:

1. **Coordinate extraction**: Landmarks are converted from MediaPipe's format to Unity Vector3 coordinates using reflection-based property access to handle API variations
2. **Handedness determination**: The system uses MediaPipe's built-in handedness classification rather than heuristic-based detection, with mirroring correction for camera perspective
3. **Gesture classification**: A hierarchical decision tree checks for pinch, then closed hand, then palm orientation

### 3.2.2 Palm Orientation Detection

Palm orientation is calculated using cross-product analysis of hand geometry. The system computes vectors from the wrist (landmark 0) to the index MCP (landmark 5) and pinky MCP (landmark 17), then calculates their cross product to determine the palm normal:

```
Vector3 vec1 = indexMcp - wrist;
Vector3 vec2 = pinkyMcp - wrist;
Vector3 normal = Vector3.Cross(vec1, vec2).normalized;
float palmZ = normal.z;
```

The Z-component of this normal vector indicates palm orientation, with handedness correction applied for left hands. This approach proved more robust than previous methods, with the palm orientation threshold reduced from 0.3 to 0.2 to improve sensitivity while maintaining accuracy.

### 3.2.3 Continuous 360-Degree Rotation

While the original system detected only binary palm-up/palm-down states, the enhanced implementation tracks continuous hand rotation. The palm normal calculation provides full 3D orientation data, enabling applications to respond proportionally to hand rotation rather than treating it as discrete state changes. This was particularly important for interactions requiring fine-grained control, such as rotating projected objects or using hand orientation as a dial input.

Figure 2: Continuous Rotation Tracking. The system tracks full 360-degree hand orientation, allowing digital content to remain physically anchored and correctly oriented even when the hand is rotated upside down.

### 3.2.4 Spatial Depth Integration

While MediaPipe Hands provides a Z-component representing relative depth, empirical testing revealed that raw depth values could be noisy or inconsistent depending on hand rotation. To implement robust depth-based scaling, the system instead calculates a "Hand Size" metric derived from the geometric span of the hand landmarks. This metric serves as a stable proxy for distance from the camera.

The `HandLandmarkerRunner` computes this metric by averaging the hand's vertical length (wrist to middle finger tip) and horizontal span (index to pinky tip) in normalized coordinates:

```
Vector2 wrist = GetLandmarkPosition(landmarks[0]);
Vector2 middleTip = GetLandmarkPosition(landmarks[12]);
Vector2 indexTip = GetLandmarkPosition(landmarks[8]);
Vector2 pinkyTip = GetLandmarkPosition(landmarks[20]);

float handLength = Vector2.Distance(wrist, middleTip);
float handSpan = Vector2.Distance(indexTip, pinkyTip);
float handSize = (handSpan + handLength) * 0.5f;
```

The system maps this hand size to an interaction scale using a direct linear interpolation. As the hand moves closer to the camera, its perceived size increases. The system maps this larger hand size to a larger content scale, creating a "pull-to-zoom" metaphor. To prevent visual overwhelming upon initial pickup, the system starts at a reduced scale of 0.6x when the hand is far, expanding to 2.2x as the hand approaches the camera:

```
// Define empirical size thresholds
float closeHandSize = 0.55f;  // Hand close (appears large)
float farHandSize = 0.25f;    // Hand far (appears small)
```

```
// Calculate interpolation factor (0.0 to 1.0)
float t = (handSize - farHandSize) /
          (closeHandSize - farHandSize);

// Map to zoom levels: Far = 0.6x, Close = 2.2x
float targetScale = Mathf.Lerp(0.6f, 2.2f, t);

// Clamp for stability to ensure visibility
return Mathf.Clamp(targetScale, 0.5f, 3.0f);
```

This approach provides intuitive push-pull control over content size, creating a dynamic range that supports both detailed inspection and overview management.

#### 3.2.5 Projector Parallax Compensation

A significant challenge in the platform-mounted projection setup is the parallax error introduced by the projector's angle. As the user lifts their hand towards the projector (increasing the Z-height), the projected image naturally shifts position relative to the hand's surface due to the oblique projection angle. Without correction, content appears to slide towards the fingertips as the hand rises.

To address this, the system implements a vertical depth compensation algorithm. The `SelectablePageImage` component calculates a dynamic Y-axis offset based on the object's current scale (which serves as a proxy for height). As the object scales up (indicating higher elevation), a proportional downward offset is applied to the projected position:

```
// Calculate downward shift based on how close the hand is
// 0.6f is the baseline scale (table height)
float projectorParallaxOffset =
    (currentDistance - 0.6f) * verticalDepthCompensation * 1.2f;

Vector3 targetPosition = new Vector3(
    targetHandTrackingPosition.x,
    targetHandTrackingPosition.y + handHeightOffset -
        projectorParallaxOffset,
    0);
```

This correction ensures the virtual content remains visually anchored to the palm center regardless of hand height, preserving the illusion of attachment during volumetric movement.

### 3.3 Fine-Grained Gesture Recognition

#### 3.3.1 Pinch Gesture Detection with Hysteresis

To enable precise interaction with small UI elements, I implemented robust pinch gesture recognition that avoids false positives from hand tremor or rapid toggling. The system calculates the 2D distance between the thumb tip (landmark 4) and the index fingertip (landmark 8) in normalized image space:

Figure 3: Volumetric Tilt and Parallax Adaptation. **Left:** Pitching the hand (bottom of palm up) triggers vertical depth compensation to prevent image sliding. **Center & Right:** Rolling the hand (raising the left or right side of the palm) updates the projection's local rotation, compressing the image to adhere to the hand's changing surface plane.

```
float rawDist = Vector2.Distance(
    new Vector2(thumbTip.x, thumbTip.y),
    new Vector2(indexTip.x, indexTip.y)
);
```

To prevent jitter, the system applies temporal smoothing and implements hysteresis with two thresholds:

- `PINCH_DISTANCE_ON = 0.045f`: Stricter threshold to *initiate* a pinch
- `PINCH_DISTANCE_OFF = 0.075f`: Looser threshold to *maintain* a pinch once started

This dual-threshold approach prevents rapid oscillation between pinched and unpinched states. Additionally, the system maintains per-hand state tracking using dictionaries to support multi-hand scenarios.

### 3.3.2 Interactive Slider Implementation

The `HandPinchSlider` component demonstrates practical application of pinch gestures for precise UI control. The slider implementation required several specific design considerations to function reliably in a projection context:

1. **Precise Interaction Point**: Early iterations using the Hand Center (approximate palm centroid) proved imprecise for fine motor tasks like slider manipulation, as the visual pinch occurred far from the computed tracking point. The updated implementation calculates a specific "Pinch Midpoint"—the exact geometric center between the thumb tip (landmark 4) and index fingertip (landmark 8).

```
Vector2 thumbTip = GetLandmarkPosition(landmarks[4]);
Vector2 indexTip = GetLandmarkPosition(landmarks[8]);
Vector2 pinchCenter = (thumbTip + indexTip) * 0.5f;
handPinchPositions[i] = NormalizedToScreenPosition(pinchCenter);
```

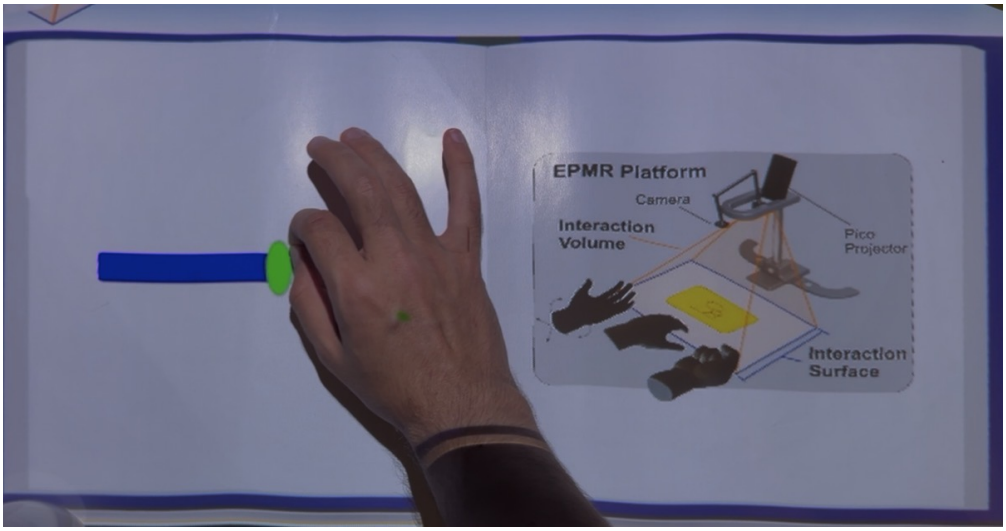


Figure 4: The Pinch Slider interface. The user adjusts a parameter by performing a fine-grained pinch gesture. The UI tracks the geometric center between the thumb and index finger for precise control.

This midpoint is tracked specifically for interaction events, aligning the UI cursor exactly with the user's physical pinch action.

2. **Coordinate transformation**: Hand position is converted from screen space to the slider's local UI coordinate space using Unity's `RectTransformUtility`:

```
RectTransformUtility.ScreenPointToLocalPointInRectangle(
    sliderRect, screenPos, uiCamera, out Vector2 localPoint)
```

3. **Position smoothing**: To compensate for tracking noise, the slider uses exponential smoothing with a configurable factor (default 0.25).

4. **Temporal filtering**: A minimum hold time (0.03s) prevents accidental activation, while a release grace period (0.25s) allows users to briefly relax their grip without losing control of the slider handle.

This implementation demonstrates how EPMR can support precise, continuous control despite the challenges of mid-air, camera-based tracking. The slider proved effective for adjusting parameters like volume, brightness, or timeline position in subsequent application prototypes.

### 3.3.3 Skeuomorphic Page Turning

To support natural reading metaphors for multi-page content, the system repurposes the pinch gesture for skeuomorphic page turning. The 'Book' controller monitors the smoothed pinch centroid relative to the projected content's bounds.

The page-turning implementation incorporates several calibrated parameters to balance responsiveness with stability. A spatial threshold of 0.4 (40% of the document width from each edge) defines the interaction zone where pinch gestures trigger page curls. To prevent accidental activation, the system requires a 120ms sustained pinch before initiating the curl animation. Once activated, a 250ms grace period allows brief gesture dropouts—common with hand tracking—without terminating the interaction. Position smoothing ($\alpha = 0.35$) filters high-frequency jitter in hand tracking data, ensuring smooth page deformation as the hand moves across the surface. These parameters were empirically tuned through iterative testing to achieve natural, stable page-turning behavior.

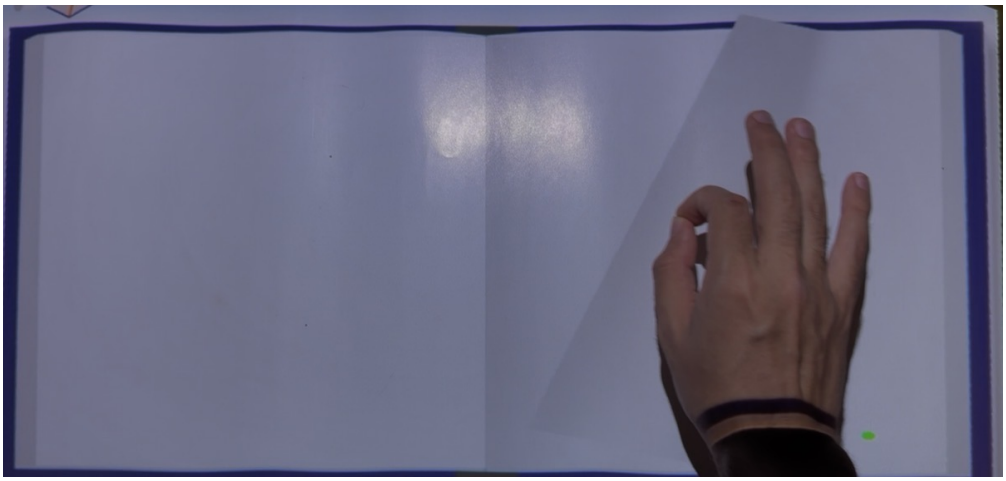

Figure 5: Skeuomorphic interactions. The user pinches the corner of a projected digital book to peel the page, blending physical gesture with virtual deformation.

## 3.4 Multi-Surface Projection

### 3.4.1 Custom Object Detection

To expand the interaction space beyond the hand, I explored projection onto environmental objects using custom-trained landmark detection models. The proof-of-concept focused on a tray as a secondary projection surface, chosen for its flat geometry and practical utility as a "palette" or "shelf" for organizing content.

The detection pipeline involved:

1. **Dataset collection**: Capturing images of the target tray from multiple angles and lighting conditions
2. **Landmark annotation**: Manually labeling key corners and reference points on the tray
3. **Model training**: Using MediaPipe Model Maker (based on MobileNet architectures) to train a custom landmark detector

4. **Integration**: Incorporating the trained model alongside hand tracking, running both detection pipelines in parallel

The trained model outputs landmark coordinates in the same normalized image space as hand landmarks, enabling consistent projection mapping across both surfaces.

### 3.4.2 Projection Mapping and Surface Integration

Once tray landmarks are detected, the system calculates a homography transformation to map projected content onto the tray's physical plane. This involves:

- Computing the tray's 3D pose from detected 2D landmarks and known physical dimensions
- Adjusting projector output to compensate for perspective distortion
- Handling occlusion when the hand passes over the tray (determining which surface has rendering priority)

To accommodate different content types, the tray projection system implements two distinct adaptive locking modes. In *Centroid-Lock Mode*, content snaps to the geometric centroid of the detected landmarks, maintaining its original aspect ratio—ideal for reviewing individual photographs. In *Surface-Fill Mode*, the system applies a homography transformation to stretch the content to the four detected corners, filling the entire physical bounds of the tray. This effectively textures the object, transforming the tray into a dedicated display surface for maps, documents, or background elements.

The integration enabled new interaction patterns:

- **Transfer gestures**: Moving content from hand to tray by bringing the hand near the tray surface while releasing a grasp
- **Spatial organization**: Using the tray as persistent storage area for "parked" content while the hand manipulates other items
- **Multi-zone interfaces**: Treating hand and tray as distinct interaction contexts with different affordances

## 3.5 Performance Optimization

Throughout development, I implemented several optimizations to maintain real-time responsiveness:

- **Gesture recognition caching**: Per-hand state is cached and only updated when landmarks change, avoiding redundant calculations
- **Selective landmark processing**: Only landmarks relevant to the current interaction context are processed each frame

Figure 6: Adaptive Object Projection. Top Row: *Centroid-Lock Mode* anchors the photo to the center of the tray, maintaining aspect ratio even as the tray is moved and rotated (Right). Bottom Row: *Surface-Fill Mode* texture-maps the content to the tray's physical corners, warping the image to fit the surface geometry dynamically.

- **Smoothing trade-offs**: Balancing temporal smoothing (which adds latency) against responsiveness through configurable parameters
- **Coordinate space optimizations**: Pre-computing transformations between screen, world, and UI coordinate spaces rather than recalculating each frame

These optimizations improved frame rates from approximately 15-20 fps in early prototypes to consistent 30+ fps, with gesture recognition latency reduced from 150-200ms to 50-80ms. While still not matching the immediacy of direct touch input, this performance proved sufficient for fluid interaction.

## 3.6 Communication Architecture

The system maintains the original OSC-based communication with the wearable pendant display. The `OSCImageSender` component transmits image indices over the network when transfer gestures are detected:

```
OSCMessage message = new OSCMessage(imageOscAddress);
message.AddValue(OSCValue.Int(imageIndex));
transmitter.Send(message);
```

This decoupled architecture allows the pendant to be implemented as a standalone device (ESP32-based AMOLED display) that simply listens for image update commands, without requiring tight integration with the Unity application. The approach could be extended to support multiple pendants or other networked displays.

# 4 Novel Applications and Use Cases

## 4.1 From Memory to Design Tool

While Graspable Memories focused on personal, reflective engagement with photographs, the enhanced EPMR platform supports active, collaborative design workflows. The combination of volumetric gestures, pinch interaction, and multi-surface projection enables diverse applications that leverage spatial manipulation and embodied interaction.

### 4.1.1 Parametric Design Exploration

The pinch-slider interface enables precise, real-time adjustment of design parameters while maintaining spatial context. In 3D modeling scenarios, designers could project a CAD model onto their palm, then use sliders projected nearby to adjust parameters:

- **Geometric transformations**: Sliders control scale, rotation angle, or extrusion depth while the model updates in real-time on the hand. Depth positioning of the hand could control overall model size, creating a natural zoom metaphor—bringing the hand closer enlarges the model, moving away shrinks it.
- **Material properties**: A color picker slider lets designers scrub through material options (matte, glossy, metallic) while the projected model reflects changes instantly. Multiple sliders could control hue, saturation, and brightness independently.
- **Document manipulation**: Documents project onto the table or tray surface, where users can grasp them with an open palm—similar to Graspable Memories' selection mechanism. Once grasped, the document follows the hand and can be repositioned or transferred to other surfaces. Sliders projected near the document enable zoom level adjustment, brightness/contrast control, or rotation. A "page turner" slider could scrub through multi-page documents, with thumbnails of adjacent pages appearing as a filmstrip alongside the main content.
- **Image review and adjustment**: Photographs project onto flat surfaces where they're easily visible. Users grasp images to examine them more closely, then use pinch sliders to adjust properties like brightness, saturation, or crop boundaries. The combination of surface-based viewing (for legibility) and hand-based manipulation (for mobility) provides flexibility that purely hand-projected or purely fixed-surface systems lack.
- **Temporal scrubbing**: For time-series data or animations, a slider serves as a timeline, allowing designers to step through different states. Hand rotation could control playback speed—palm up for forward, palm down for reverse, rotation angle determining playback rate.
- **Multi-parameter control**: By projecting multiple sliders simultaneously (e.g., arranged around the content on the table), designers could adjust related parameters in quick succession without switching modes. Pinching different sliders activates them,

enabling fluid parameter exploration while the content remains visible on a stable surface.

This approach contrasts with traditional GUI sliders by maintaining spatial presence—content remains visible on legible surfaces (tables, trays) while users employ hand gestures for selection and manipulation, and pinch gestures for fine-tuned adjustments. The combination of surface-based display (for readability) with hand-based interaction (for mobility and gesture) provides multi-scale control unavailable in purely touch-based or purely spatial interfaces.

### 4.1.2 Collaborative Layout and Composition

EPMR's projection-based nature makes it inherently social—others can see manipulated content without specialized eyewear. By expanding the projection area to cover a full tabletop, the system could support co-located collaborative work:

- **Spatial mood boards**: Design teams project reference images, sketches, and text snippets onto a shared table. Each participant uses hand gestures to position and scale elements, arranging them spatially to explore relationships. Content "sticks" to the table surface when released, creating persistent layouts. Participants can reassign ownership of elements by grasping them and extending their hand toward a teammate—the content follows the hand across the table, then transfers when the recipient covers it with their own palm.

- **Architectural walkthroughs**: Architects project building floor plans onto the table, using hand gestures to add furniture, adjust room dimensions, or switch between floors. Multiple stakeholders can simultaneously manipulate different areas, with conflict resolution handled through proximity—the closest hand "claims" ambiguous regions. Pinch sliders adjust properties like wall thickness or ceiling height, with changes reflected in a 3D perspective view projected on a nearby vertical surface.

- **Data visualization composition**: Analysts arrange projected charts and graphs spatially to explore correlations. Grasping a chart and moving it near another triggers automatic linking—a projected line connects them, indicating a discovered relationship. Hand rotation filters data—rotating the palm while holding a chart scrubs through time periods or categories, with linked visualizations updating synchronously.

- **Storyboard sequencing**: Filmmakers or designers arrange narrative sequences by projecting scene sketches on the table. Dragging images left-to-right establishes temporal order; stacking images vertically groups alternative takes. Users can "deal" content to collaborators like playing cards—a quick flicking gesture propels a projected image toward another participant, where it appears on their palm for review.

- **Role-based interaction zones**: The system divides the table into functional regions—a "source" area where new content appears, "work areas" for individual manipulation, and a "shared review" zone at the table's center. Gestures adapt based on

zone: in source areas, open palms create new blank canvases; in work areas, pinch gestures enable precise editing; in the review zone, only viewing and basic rearrangement are permitted, preventing accidental modification of agreed-upon work.

These scenarios leverage EPMR's strengths—spatial manipulation feels direct and legible to all participants, while the lack of physical artifacts means layouts can be instantly saved, versioned, or reset without cleanup. The system could capture interaction histories, enabling playback of design evolution or reversion to earlier states.

#### 4.1.3 Multi-Surface Information Display

By training detection models for multiple object types, EPMR can create heterogeneous interaction ecologies where different objects afford different interactions:

- **Document trays as persistent displays**: Documents project onto rectangular trays, remaining readable at natural viewing angles. Users grasp documents from the tray surface using open-palm gestures—covering the document causes it to "lift" and follow the hand, enabling repositioning to other surfaces or transfer to collaborators. Multiple documents can stack on the tray with slight offsets, creating a 3D layered effect visible without needing to grasp them. Removing the top document by grasping and moving it away reveals documents beneath.

- **Reading surfaces and annotation**: Large documents (reports, diagrams, maps) project onto flat work surfaces where they remain legible. Users don't need to hold these documents in their palms—instead, the hand serves as an interaction tool. Pinch gestures create annotation marks, hand rotation adjusts viewing angle, and depth positioning controls zoom level, all while the document remains stably projected on the surface for comfortable reading. Additionally, users can navigate multi-page documents through skeuomorphic page turning: by pinching the physical corner of the projected document, they can "grab" and peel the page across the surface, with the virtual page deforming to follow the hand's trajectory.

- **Color palettes and tool selection**: A painter's palette object displays color swatches projected onto its physical wells. Users "dip" their fingertips into projected colors via pinch gestures, then draw on other surfaces (table, canvas board) by dragging the pinched fingers, leaving colored trails. The palette remains stable on the table while the hand moves freely, providing a fixed reference unlike purely hand-based interaction, where color selection would compete with drawing actions.

- **Layered information cards**: Physical cards with markers are stacked on a surface. The top card displays a summary; grasping and removing it reveals the next card underneath, which now projects detailed information. This creates a physical depth metaphor for information hierarchy—literally digging deeper by removing layers. Users can "flip" projected cards by rotating them 180° in their hand, revealing content on the virtual "back," then place them back down to share with others.

- **Multi-user information distribution**: In collaborative scenarios, detected clipboards, tablets, or trays assigned to each participant receive content routed to them. Documents initially project onto a central sharing surface. A presenter grasps a projected document (which follows their hand) and extends it toward a colleague's clipboard or tray; when the hand hovers over that surface, the document highlights, and releasing the grasp transfers it, where it becomes projected and readable on the recipient's surface. This creates a spatial delivery mechanism—physically moving content through 3D space toward recipients—that leverages human spatial reasoning.

- **Workspace organization and persistence**: Multiple trays, boxes, or flat surfaces around the interaction area serve as "folders" for organizing content by category. Documents and images project onto these surfaces where they remain visible and readable. Users grasp content from one surface and move their hands toward another to transfer it—financial reports to the left tray, design mockups to the right. The system remembers which content is assigned to which surface, maintaining state across sessions. When starting a new session, the system projects previously saved layouts instantly onto their respective surfaces, reducing setup time.

- **Presentation and review modes**: A vertical surface (wall, board) serves as a presentation screen for shared viewing, while horizontal surfaces (tables, trays) serve as individual workspaces. Presenters grasp content from their workspace and "throw" it toward the vertical surface with a flicking gesture, where it enlarges for group viewing. Audience members can grasp content from the presentation surface to examine it individually on their personal workspace surfaces, then return it to the shared display when finished.

This multi-surface approach addresses both readability and fatigue—documents remain projected on stable surfaces at comfortable viewing distances and angles, while hands serve as interaction tools for grasping, moving, and manipulating content. Users can rest their arms on surfaces while gesturing, and the distinction between "display surfaces" (for content viewing) and "hand as interaction tool" (for manipulation) creates a natural division of labor that mirrors how we interact with physical documents.

## 4.2 Evaluation and Observations

Informal testing and development iteration revealed several insights about EPMR's enhanced capabilities and remaining challenges:

### 4.2.1 Projection-Free Development Benefits

Interestingly, the system worked most reliably during development when the projector was disabled, using only screen-based visualization of projected content. This counterintuitive finding highlights a core challenge: the projector's illumination interferes with hand tracking, creating feedback where bright projected elements confuse MediaPipe's landmark detection. This necessitated implementing more lenient tracking thresholds and hysteresis parameters to maintain interaction continuity despite increased noise.

This observation suggests that production EPMR systems would benefit from:

- Infrared-based tracking immune to visible light projection
- Temporal filtering strategies that predict hand position during brief tracking dropouts caused by bright projected content
- Adaptive thresholding that adjusts landmark confidence requirements based on detected projection intensity in the hand region

### 4.2.2 Visual Realism and Registration

Ensuring that projected images appeared naturally integrated with the hand proved challenging. Users reported discomfort when projected content "slipped" relative to hand motion or when perspective distortion made flat images appear skewed. Addressing this required:

- Careful projector-camera calibration using checkerboard patterns and homography estimation
- Real-time perspective correction based on hand pose—when the hand tilts, projected content is pre-warped to appear undistorted from the user's viewpoint
- Visual anchoring through subtle drop shadows projected slightly offset from hand landmarks, creating depth cues that ground projected elements to the physical surface

The modular architecture proved valuable here—separating projection mapping from gesture recognition allowed rapid iteration on visual quality without re-implementing interaction logic.

### 4.2.3 Gesture Vocabulary Management

Throughout testing, we consistently introduced new users by demonstrating the complete gesture set before allowing interaction. Without this training, users defaulted to intuitive but unrecognized gestures (pointing at images, swiping horizontally). Once trained, users generally remembered core gestures (grasp, release) but often forgot advanced features (pinch-to-adjust) if not used frequently.

This mirrors findings from Graspable Memories and reinforces the need for persistent visual reminders or just-in-time guidance. Users suggested:

- Small gesture icons projected near their hands during initial sessions, fading after successful use
- Audio feedback when attempting unrecognized gestures: "Try pinching to adjust" rather than silent failure
- A "help" gesture (e.g., spreading all fingers) that projects a quick reference card

#### 4.2.4 Hysteresis Tuning and Temporal Stability

The dual-threshold hysteresis for pinch detection (0.045 to engage, 0.075 to maintain) successfully reduced jitter but introduced subtle latency—about 100ms between initiating a pinch and the system responding. Users adapted quickly but noted the system felt "slightly sluggish" compared to direct touch. Further improvements would require:

- Reducing camera-to-projection latency through hardware acceleration or lower-level MediaPipe integration
- Predictive algorithms that anticipate pinch onset from finger velocity, triggering a response before fingers fully touch
- Adaptive hysteresis bands that tighten once a stable pinch is detected, balancing initial stability with subsequent responsiveness

Despite optimizations, fundamental latency constraints of vision-based tracking remain. Users tolerated 50-80ms delays for coarse spatial manipulation but found them problematic for precision tasks like slider adjustment. This suggests EPMR is best suited for workflows where spatial expressiveness matters more than millisecond-level precision—conceptual design over technical drafting, creative exploration over data entry.

Overall, observations confirm that EPMR's technical foundations are sound but highlight where refinement is needed to achieve the seamless, tangible-feeling interaction that justifies the paradigm's promise. The system successfully moved from discrete memory selection to continuous parametric control, validating the core thesis while exposing new challenges in tracking robustness, user guidance, and ergonomic sustainability.

# 5 Discussion

## 5.1 Expanding the EPMR Design Space

This work demonstrates that EPMR's core principle—treating bodily occlusion as intentional interaction—extends far beyond personal memory systems. By adding volumetric sensing and multi-surface projection, the paradigm becomes applicable to:

- **Creative collaboration**: Spatial arrangement of design elements, shared workspaces
- **Data manipulation**: 3D visualization where hand position and orientation control the viewpoint or filter parameters
- **Educational contexts**: Interactive demonstrations where physical gestures illustrate abstract concepts

The transition from discrete to continuous gestures is particularly significant. While the original binary states (open/closed hand) were sufficient for selecting and transferring memories, nuanced creative work requires proportional control. Depth, rotation, and pinch gestures provide this, aligning EPMR with broader trends in spatial computing toward expressive, high-dimensional input.

## 5.2 Challenges and Limitations

Despite these advances, several challenges remain that impact both the technical performance and user experience of EPMR systems.

### 5.2.1 Tracking Robustness and Interaction Fidelity

MediaPipe Hands performs well under controlled lighting, but fast movements or partial occlusion still cause occasional failures that fundamentally detract from the feeling of tangible interaction. The core promise of EPMR—that bodily occlusion can restore physicality to digital content—relies on the immediacy and reliability that characterize physical manipulation. When tracking failures interrupt the continuous flow of interaction, the illusion of holding and touching projected content breaks down, reducing the experience to mere command-and-control rather than embodied engagement.

Several technical factors contribute to these robustness issues. First, the RGB camera used in the current implementation provides limited depth perception and is susceptible to interference from the projector's illumination, creating feedback loops where projected content affects hand detection. Second, the MediaPipe Hands Unity plugin, while convenient, introduces additional latency compared to native implementations. Third, the system currently operates on CPU, leaving GPU resources underutilized despite their potential for parallel processing of landmark detection and projection mapping.

Multiple avenues exist for addressing these limitations:

- **Depth-sensing cameras**: Replacing the RGB camera with an Intel RealSense or Azure Kinect depth camera would provide robust 3D hand tracking immune to projected light interference. Depth data enables more accurate occlusion detection and eliminates ambiguity in determining whether the hand is touching the projection surface.

- **Multi-camera configurations**: Deploying multiple cameras at different angles could improve tracking reliability through sensor fusion, reducing failures from self-occlusion or momentary tracking loss. This approach has proven effective in motion capture systems and could be adapted for hand tracking.

- **Alternative MediaPipe implementations**: Running MediaPipe through Python with direct TensorFlow Lite integration may reduce latency compared to the Unity C# plugin. Alternatively, compiling MediaPipe solutions natively in C++ could achieve further performance gains.

- **GPU optimization**: Offloading landmark detection and coordinate transformations to GPU compute shaders would free CPU resources and potentially halve processing latency. Unity's Compute Shader API or external libraries like CUDA could be leveraged for this purpose.

- **Predictive tracking**: Implementing Kalman filtering or other predictive algorithms could smooth hand trajectories and maintain interaction continuity during brief tracking losses, similar to techniques used in VR controller tracking.

Ultimately, achieving the sub-10ms latency characteristic of direct touch interfaces may require custom hardware acceleration or hybrid approaches combining vision-based and sensor-based tracking (e.g., IMUs on a lightweight glove).

### 5.2.2 Gesture Discoverability and User Guidance

As the gesture vocabulary expands beyond the original three states (open palm up, open palm down, closed fist) to include pinch, continuous rotation, and depth manipulation, users may struggle to discover and remember available interactions. Unlike graphical user interfaces where affordances are visually evident, mid-air gestures lack persistent visual cues indicating what actions are possible.

This challenge is particularly acute during initial use. In our observations, users required explicit demonstration of all gestures before feeling comfortable interacting with the system. Without guidance, participants defaulted to familiar gestures (pointing, waving) that the system did not recognize, leading to frustration.

Several strategies could improve discoverability:

- **Projected tutorial sequences**: Before each session, the system could project an interactive tutorial directly onto the interaction surface, showing animated hand silhouettes performing each gesture alongside text descriptions. Users could practice each gesture until the system confirms successful recognition.

- **Persistent gesture palette**: A small, semi-transparent legend could remain visible at the edge of the projection field, displaying icons representing available gestures and their current states (e.g., highlighting the pinch icon when a pinch is detected). This provides continuous reinforcement without cluttering the main interaction area.

- **Contextual hints**: The system could detect when users attempt unrecognized gestures (e.g., repeated failed selections) and offer just-in-time guidance, projecting a brief animation showing the correct gesture for the intended action.

- **Progressive disclosure**: Rather than exposing all gestures simultaneously, the system could introduce gestures gradually, starting with core interactions (grasp/release) and revealing advanced features (pinch, rotation) only after users demonstrate proficiency with basics.

- **Multimodal feedback**: Combining visual feedback (e.g., highlighting projected content when a hand approaches) with auditory cues (subtle tones when gestures are recognized) could provide immediate confirmation that the system has understood user intent, building confidence in the interaction model.

These approaches align with research on gesture-based interfaces, showing that explicit training combined with persistent reminders significantly improves learnability and retention of novel gesture vocabularies.

### 5.2.3 Physical Fatigue and Ergonomic Considerations

Extended mid-air interaction inevitably causes arm fatigue, a well-documented challenge in gesture-based systems. Unlike touch interfaces, where hands rest on a surface, or tangible interfaces, where objects provide physical support, EPMR requires users to hold their hands aloft within the projection volume. Muscle fatigue accumulates quickly—typically within 5-10 minutes of continuous use—degrading interaction quality as users compensate with coarser movements.

Our observations revealed that fatigue manifests both physically (users lowering their arms between interactions) and behaviorally (increased reliance on faster, less precise gestures to complete tasks quickly). This fundamentally limits session duration and makes EPMR unsuitable for tasks requiring sustained interaction without ergonomic considerations.

Several design strategies can mitigate fatigue:

- **Rest positions and "parking" gestures**: The system could recognize when users lower their hands below the active interaction zone and enter a paused state, preserving context while allowing rest. A simple raise-hand gesture would resume interaction. The multi-surface tray projection exemplifies this—users can "park" content on the tray and lower their hands, reducing continuous arm elevation.

- **Seated interaction design**: Configuring the projection to work on horizontal or near-horizontal surfaces (e.g., tabletops at 30-45° angles) allows users to rest forearms on the table edge while maintaining hand visibility. This mirrors successful interactive tabletop systems that support extended collaboration sessions.

- **Interaction density reduction**: Designing workflows that minimize required gestures—e.g., using sustained poses rather than repeated actions, or combining multiple operations into compound gestures—can reduce total movement and delay fatigue onset.

- **Multi-user rotation**: In collaborative scenarios, the system could suggest periodic role rotation, where different users take turns manipulating content while others observe, naturally enforcing rest periods.

- **Hybrid modalities**: Combining EPMR with voice commands or foot pedals for mode switching could reduce the number of hand gestures required, reserving physical interaction for spatially meaningful tasks like positioning and orientation.

Commercial systems have addressed similar challenges through varied approaches: Microsoft's HoloLens encourages brief gestural interactions supplemented by gaze and voice; Leap Motion research demonstrated that downward-facing cameras enable interaction with hands resting on surfaces. EPMR could benefit from integrating these lessons, perhaps projecting onto vertical surfaces for brief, high-impact interactions while reserving horizontal surfaces for sustained work.

### 5.2.4 Multi-Object Scalability and Generalization

Training custom detection models for each object is labor-intensive, requiring dataset collection, manual annotation, model training, and validation—a process that consumed several hours for the tray prototype. This workflow does not scale to environments with diverse objects or scenarios where users wish to appropriate arbitrary surfaces as projection targets.

Several technical challenges compound this limitation:

- **Object differentiation**: When multiple trained objects are visible simultaneously, the system must correctly identify and track each instance. Landmark-based approaches struggle when objects have similar geometric features, potentially causing misclassification or tracking jumps between objects.
- **Occlusion handling**: When hands occlude trained objects, landmark detection often fails or produces spurious detections. Unlike hands, which are expected to move and occlude content, environmental objects should remain stable—distinguishing intentional object manipulation from incidental occlusion requires additional heuristics.
- **Projection surface variance**: Objects have varying geometries, materials, and reflectance properties that affect projection quality. A homography transformation calibrated for a flat white tray produces distorted results when applied to curved or textured surfaces. Robust projection mapping requires per-object geometric models and photometric calibration.
- **Real-time performance**: Running multiple detection models in parallel (hands + multiple objects) increases computational load, potentially pushing frame rates below interaction thresholds. The system must prioritize detection tasks and gracefully degrade when resources are constrained.

Several approaches could improve scalability:

- **Marker-based hybrid tracking**: Attaching fiducial markers (e.g., ArUco tags) to objects enables fast, robust detection without custom training. While less "magical" than markerless tracking, this approach is practical for controlled environments and supports unlimited object variety. The system could project registration patterns during setup, guiding users to place markers correctly.
- **General object segmentation**: Recent advances in zero-shot object segmentation (e.g., Meta's Segment Anything Model) could enable the detection of arbitrary objects without prior training. The system could ask users to indicate objects of interest through pointing gestures, then track those regions across frames. However, segmentation alone provides insufficient geometric information for projection mapping—additional depth sensing or user-guided calibration would be needed.
- **Object categories and templates**: Rather than training models for specific object instances, training category-level detectors (e.g., "rectangular trays," "cylindrical containers," "flat books") could generalize better while providing sufficient geometric constraints for projection. Users could select from a menu of trained categories when introducing new objects.

- **Interactive calibration**: When users place a new object in the scene, the system could project a calibration pattern onto it and ask users to trace its outline with hand gestures. This quick calibration establishes a 2D-to-3D mapping without requiring model training, though accuracy would depend on user precision.

- **Depth-based surface detection**: With depth cameras, the system could automatically detect planar surfaces in the environment and offer them as projection targets, bypassing object recognition entirely. This approach works well for architectural elements (walls, tables) but struggles with small or movable objects.

The choice of approach depends on the deployment context. Controlled environments (e.g., design studios) could benefit from marker-based tracking's reliability, while public installations might favor markerless approaches despite reduced robustness. Ultimately, achieving truly general-purpose multi-surface projection may require hybrid strategies that combine multiple detection methods, selecting the most appropriate technique based on object characteristics and interaction requirements.

### 5.2.5 Depth Estimation Characteristics

MediaPipe Hands provides relative depth values normalized to the hand's scale, with the wrist serving as the reference origin. While this enables proportional depth-based control, several characteristics affect interaction design:

- **Relative scaling**: Depth values are normalized to hand size, meaning absolute distances cannot be measured. However, changes in depth remain consistent and suitable for continuous control within a bounded range.

- **RGB-based limitations**: Without true depth sensing, MediaPipe infers depth from monocular cues. Ambiguities can occur with unusual hand poses or when hands approach the camera edges.

- **Orientation coupling**: Hand rotation can affect perceived depth estimates, though MediaPipe's model attempts to compensate for this through learned priors.

- **User variability**: Hand size differences between users result in different absolute depth ranges, though relative depth changes remain proportionally consistent.

Future implementations using depth cameras (Intel RealSense, Azure Kinect) would provide millimeter-accurate 3D tracking, eliminating these ambiguities. However, for the interaction patterns explored in this work—zoom control, layer selection, and mode switching—MediaPipe's relative depth proved sufficient when properly bounded and normalized.

## 5.3 Design Implications

The work presented here suggests several principles for designing volumetric projected interfaces:

1. **Layer discrete and continuous**: Combine binary gestures (pinch/release, open/closed) for mode switching with continuous parameters (depth, rotation) for fine control

2. **Respect embodied metaphors**: Users expect certain mappings (e.g., lifting hand = lifting object). Violating these creates confusion

3. **Provide rich feedback**: In the absence of tactile sensation, visual and potentially auditory feedback must clearly communicate system state

4. **Support transitions between surfaces**: Moving content from hand to tray (or other objects) should feel fluid, leveraging proximity or explicit gestures

5. **Bound continuous parameters**: Define explicit working ranges for depth, rotation, and other continuous inputs to prevent extreme values and provide natural limits that users can discover through interaction

### 5.4 Toward General-Purpose EPMR

This thesis repositions EPMR from a domain-specific memory system to a general-purpose interaction platform. Future work should explore:

- **Multi-user interaction**: Enabling multiple participants to simultaneously manipulate projected content, with conflict resolution and shared awareness

- **Adaptive gesture recognition**: Machine learning approaches that learn user-specific gesture styles or adapt to different hand morphologies

- **Integration with other modalities**: Combining EPMR with voice input, gaze tracking, or handheld controllers for hybrid interaction

- **Application domains**: Evaluating EPMR in specific contexts like architecture, data analysis, or education to identify domain-specific requirements

## 6 Conclusion

This thesis extends the Embodied Projected Mixed Reality paradigm from a personal memory interaction system to a versatile platform for spatially-aware, collaborative design. By integrating volumetric gestures—spatial depth, continuous rotation, hand tilt, and pinch interactions—and expanding projection beyond the hand to arbitrary objects, I have demonstrated that EPMR can support nuanced manipulation of digital content in ways that feel intuitive and embodied.

The original Graspable Memories work showed that occlusion could be reframed as an opportunity rather than an obstacle. This work shows that the same principle scales: as we add dimensions to the interaction space—depth, rotation, multiple surfaces—the richness of possible interactions grows proportionally. The hand becomes not just a selector or container,

but a multi-degree-of-freedom input device that maintains the directness and legibility of physical manipulation.

EPMR now stands as a foundation for exploring how projected interfaces can serve domains beyond personal reflection: collaborative design, parametric exploration, educational demonstration, and more. By treating the body as an interface, we create experiences that are visible to others, grounded in shared physical space, and free from the isolation of head-mounted displays or the flatness of touchscreens.

Looking ahead, the challenge is not merely technical refinement but conceptual: understanding what kinds of interactions are best suited to projected, embodied interfaces, and how EPMR can complement rather than replace other interaction paradigms. This thesis offers one answer—that when digital content needs to be manipulated spatially, collaboratively, and expressively, projection onto the body and environment provides a compelling alternative to established approaches.

# References


[1] Hrvoje Benko, Ricardo Jota, and Andrew Wilson. MirageTable: freehand interaction on a projected augmented reality tabletop. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, page 199–208, New York, NY, USA, 2012. Association for Computing Machinery.

[2] Mark Billinghurst, Adrian Clark, and Gun Lee. A survey of augmented reality. *Foundations and Trends in Human–Computer Interaction*, 8(2-3):73–272, 2015.

[3] O. Bimber and R Raskar. *Spatial Augmented Reality: Merging Real and Virtual Worlds*. A K Peters/CRC Press, Wellesley, Massachusetts, 1st edition, 2005.

[4] Oliver Bimber and Bernd Frohlich. Occlusion shadows: Using projected light to generate realistic occlusion effects for view-dependent optical see-through displays. In *Proceedings. International Symposium on Mixed and Augmented Reality*, pages 186–319. IEEE, 2002.

[5] Alexandre G. de Siqueira, Brygg Ullmer, Mark Delarosa, Chris Branton, and Miriam K. Konkel. Hard and soft tangibles: Mixing multi-touch and tangible interaction in scientific poster scenarios. In *Proceedings of the Twelfth International Conference on Tangible, Embedded, and Embodied Interaction*, TEI '18, page 476–486, New York, NY, USA, 2018. Association for Computing Machinery.

[6] Paul Dourish. *Where the Action Is: The Foundations of Embodied Interaction*. MIT Press, Cambridge, MA, USA, 2001.

[7] Chris Elsden, David S. Kirk, and Abigail C. Durrant. A quantified past: Toward design for remembering with personal informatics. *Human–Computer Interaction*, 31(6):518–557, 2016.

[8] Alexandre Gomes de Siqueira, Reggie Segovia, Eduardo Gabriel Queiroz Palmeira, and Brandon Grill. Graspable Memories: A sustainable approach to holding personal memories through occlusion-aware projected interaction. In *Proceedings of IEEE AIxVR*, 2026. Under review.

[9] Jan Gugenheimer, Evgeny Stemasov, Harpreet Sareen, and Enrico Rukzio. FaceDisplay: Towards asymmetric multi-user interaction for nomadic virtual reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, page 1–13, New York, NY, USA, 2018. Association for Computing Machinery.

[10] Chris Harrison, Hrvoje Benko, and Andrew D. Wilson. OmniTouch: wearable multitouch interaction everywhere. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, page 441–450, New York, NY, USA, 2011. Association for Computing Machinery.

[11] Chris Harrison, Desney Tan, and Dan Morris. Skinput: appropriating the body as an input surface. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, page 453–462, New York, NY, USA, 2010. Association for Computing Machinery.

[12] Eva Hornecker and Jacob Buur. Getting a grip on tangible interaction: a framework on physical space and social interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '06, pages 437–446, New York, NY, USA, 2006. Association for Computing Machinery.

[13] Hiroshi Ishii and Brygg Ullmer. Tangible bits: towards seamless interfaces between people, bits and atoms. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, CHI '97, page 234–241, New York, NY, USA, 1997. Association for Computing Machinery.

[14] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, and Andrew Fitzgibbon. KinectFusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, pages 559–568, New York, NY, USA, 2011. Association for Computing Machinery.

[15] Brett Jones, Rajinder Sodhi, Michael Murdock, Ravish Mehra, Hrvoje Benko, Andrew Wilson, Eyal Ofek, Blair MacIntyre, Nikunj Raghuvanshi, and Lior Shapira. RoomAlive: magical experiences enabled by scalable, adaptive projector-camera units. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST '14, page 637–644, New York, NY, USA, 2014. Association for Computing Machinery.

[16] Jaedong Kim, Hyunggoog Seo, Seunghoon Cha, and Junyong Noh. Real-time human shadow removal in a front projection system. *Computer Graphics Forum*, 38(1):443–454, 2019.

[17] David S. Kirk and Abigail Sellen. On human remains: Values and practice in the home archiving of cherished objects. *ACM Trans. Comput.-Hum. Interact.*, 17(3), July 2010.

[18] Bettina Laugwitz, Theo Held, and Martin Schrepp. Construction and evaluation of a user experience questionnaire. In Andreas Holzinger, editor, *HCI and Usability for Education and Work*, pages 63–76, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[19] Pedro Lopes, Alexandra Ion, Willi Mueller, Daniel Hoffmann, Patrik Jonell, and Patrick Baudisch. Proprioceptive interaction. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, page 939–948, New York, NY, USA, 2015. Association for Computing Machinery.

[20] Steve Mann. Wearable computing: a first step toward personal imaging. *Computer*, 30(2):25–32, 1997.

[21] William Odom, Abi Sellen, Richard Harper, and Eno Thereska. Lost in translation: understanding the possession of digital things in the cloud. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, page 781–790, New York, NY, USA, 2012. Association for Computing Machinery.

[22] Daniela Petrelli, Elise van den Hoven, and Steve Whittaker. Making history: intentional capture of future memories. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, page 1723–1732, New York, NY, USA, 2009. Association for Computing Machinery.

[23] Daniela Petrelli and Steve Whittaker. Family memories in the home: contrasting physical and digital mementos. *Personal and Ubiquitous Computing*, 14(2):153–169, February 2010.

[24] Aaron J. Quigley and Florin Bodea. Chapter 1 - face-to-face collaborative interfaces. In Hamid Aghajan, Ramón López-Cózar Delgado, and Juan Carlos Augusto, editors, *Human-Centric Interfaces for Ambient Intelligence*, pages 3–32. Academic Press, Oxford, 2010.

[25] Ramesh Raskar, Greg Welch, Kok-Lim Low, and Deepak Bandyopadhyay. Shader lamps: Animating real objects with image-based illumination. In Steven J. Gortler and Karol Myszkowski, editors, *Rendering Techniques 2001*, pages 89–102, Vienna, 2001. Springer Vienna.

[26] Katta Spiel. The bodies of TEI – investigating norms and assumptions in the design of embodied interaction. In *Proceedings of the Fifteenth International Conference on Tangible, Embedded, and Embodied Interaction*, TEI '21, New York, NY, USA, 2021. Association for Computing Machinery.

[27] Brygg Ullmer, Sida Dai, Alexandre Gomes de Siqueira, Millon McLendon IV, Breanna Filipiak, Laila Shafiee, Winifred Elysse Newman, and Miriam K Konkel. Variations on a hexagon: Iterative design of interactive cyberphysical tokens and constraints. In *Proceedings of the Eighteenth International Conference on Tangible, Embedded, and*

*Embodied Interaction*, TEI '24, New York, NY, USA, 2024. Association for Computing Machinery.

[28] Brygg Ullmer, Orit Shaer, Ali Mazalek, and Caroline Hummels. *Weaving Fire into Form: Aspirations for Tangible and Embodied Interaction*, volume 44. Association for Computing Machinery, New York, NY, USA, 1 edition, 2022.

[29] Elise van den Hoven. A future-proof past: Designing for remembering experiences. *Memory Studies*, 7(3):370–384, 2014.

[30] Elise van den Hoven and Berry Eggen. The cue is key: Design for real-life remembering. *Zeitschrift für Psychologie*, 222(2):110–117, 2014.

[31] Martin Weigel, Tong Lu, Gilles Bailly, Antti Oulasvirta, Carmel Majidi, and Jürgen Steimle. iSkin: Flexible, stretchable and visually customizable on-body touch sensors for mobile computing. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, page 2991–3000, New York, NY, USA, 2015. Association for Computing Machinery.

[32] Andrew D. Wilson. Using a depth camera as a touch sensor. In *ACM International Conference on Interactive Tabletops and Surfaces*, ITS '10, page 69–72, New York, NY, USA, 2010. Association for Computing Machinery.

[33] Jacob O. Wobbrock, Meredith Ringel Morris, and Andrew D. Wilson. User-defined gestures for surface computing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 1083–1092, New York, NY, USA, 2009. Association for Computing Machinery.

[34] Matthew Wright. Open sound control: an enabling technology for musical networking. *Organised Sound*, 10(3):193–200, 2005.

[35] Robert Xiao, Teng Cao, Ning Guo, Jun Zhuo, Yang Zhang, and Chris Harrison. LumiWatch: On-arm projected graphics and touch input. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, page 1–11, New York, NY, USA, 2018. Association for Computing Machinery.

[36] Fan Zhang, Valentin Bazarevsky, Andrey Vakunov, Andrei Tkachenka, George Sung, Chuo-Ling Chang, and Matthias Grundmann. MediaPipe Hands: On-device real-time hand tracking, 2020.