

Gesture-based Interaction with Video-Projected User Interfaces on Furniture

Alexandru-Tudor Andrei

MintViz Lab

Ștefan cel Mare University of Suceava

Suceava, Romania

tudor.andrei@usm.ro

Radu-Daniel Vatavu

MintViz Lab

Ștefan cel Mare University of Suceava

Suceava, Romania

radu.vatavu@usm.ro



Figure 1: Example of a video-projected user interface enabled by our system for interacting with digital content displayed on different furniture surfaces, including the floor, a table, a shelf, and a polygonal surface in this photograph.

ABSTRACT

Video-projected user interfaces enable digital content to be displayed in physical environments, yet irregular surfaces require precise alignment among projection, surface geometry, and user input tracking. This paper presents a web-based system for projected interaction on furniture, which employs a calibration procedure to register the interaction area and align it with the graphical user interface. In this process, users' hands positions, tracked in 3D, are mapped to normalized surface coordinates through a homography technique, supporting taps, swipes, dwell, and motion-based interaction using either one or two hands. In a technical evaluation of our implementation, we report robust detection of user interaction and low-latency system performance.

AUTHOR KEYWORDS

Furniture; augmented surfaces; augmented reality; extended reality; motion capture; video projection; homography; gesture-based interaction; user interface.

DOI: 10.37789/icusi.2026.8

INTRODUCTION

Projected user interfaces can turn everyday physical surfaces into interactive displays, allowing digital content to be placed

directly within the user's environment. Prior work has showcased interactions on ad-hoc interactive surfaces and room-scale projection-mapped spaces [3, 5, 10], enabling interaction with digital content to expand beyond fixed screens and digital devices, and embedding it into physical contexts.

However, projecting an interface onto a physical surface introduces several technical challenges. Unlike conventional displays, physical surfaces may vary in size, orientation, shape, and location relative to the video projector and gesture sensing system. As a result, graphical elements must remain visually aligned with the projection surface, while input devices mapped accurately to corresponding interface controls. In this context, this paper presents a web-based system for the video-projection of graphical user interfaces onto physical surfaces with variable dimensions in everyday living room environments, such as furniture, which can act as a temporary interactive support for media navigation, smart home control, and digital content manipulation.

The specific contributions of this paper are as follows: (1) a web-based system for projected interaction on calibrated furniture surfaces; (2) a physical-to-visual calibration technique connecting hand locations tracked in 3D to projected 2D graphical user interface controls; and (3) the results of a technical evaluation of system performance.

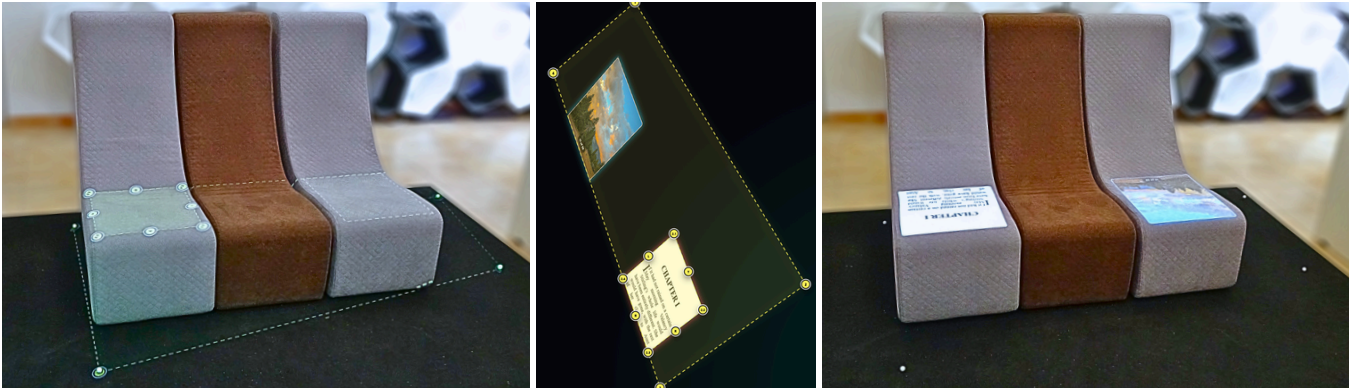


Figure 2: Calibration of video projection on a modular sofa. The physical surface is specified using corner markers (left), the projected quadrilateral is adjusted in the client application (center), and the calibrated user interface is displayed on the furniture surface (right).

RELATED WORK

Projection-based interfaces expand interaction beyond conventional displays onto physical surfaces. Early systems such as PlayAnywhere [8] showed how video projection and computer vision can support gesture-based interaction on tabletops, while OmniTouch [3] extended this to wearable configurations for interactions on hands, walls, and everyday surfaces. Similarly, WorldKit [10] enabled users to employ interactive applications through projector-camera sensing; LightSpace [9] combined multiple depth cameras and projectors to support room-scale interaction on, above, and between surfaces; and RoomAlive [5] utilized projector-camera units to render an entire room interactive.

Researchers have also investigated context-aware video projection, where the output surface adapts to the user's activity, posture, or interaction context. For instance, Gesture Bike [2] explored projected interfaces, display surfaces, and turn-signal systems for urban cycling, while wear Ur world [6] used wearable video projectors for accessing digital information on reachable physical surfaces. In the context of smart environments, gesture input has been integrated directly into furniture and everyday objects. Focusing on seated interaction, ChairMX [7] treated chairs as interactive surfaces for novel media consumption experiences, while Andrei et al. [1] investigated user preferences for chair-based gesture input.

Deploying interfaces onto diverse surfaces requires geometric calibration to align the projected image and the input coordinate system. Because projecting onto oblique, non-planar, or irregular surfaces introduces spatial distortion, prior work has focused on geometric correction and projection mapping techniques. For example, passive-active geometric calibration addresses view-dependent projection onto arbitrary surfaces through explicit calibration and corrective warping [11]. For systems mapping user input directly to projected graphical controls, homography transformations offer a practical approach for processing coordinates across planar surfaces [4].

TECHNICAL IMPLEMENTATION

Apparatus

Hand location and movement in 3D was captured using a motion tracking system composed of six Vicon Vero 1.3¹

cameras, calibrated to work with Vicon Nexus 2.15,² with reflective markers providing 3D coordinates for physical surface registration and hand tracking. The cameras were connected to a processing workstation (Dell Optiplex 7080 with an Intel Core i7-10700 CPU and 16 GB RAM) through a dedicated Dahua S4101-16ET2GF-240-C network switch. The client application ran on a Lenovo ThinkStation P3 (Intel Core i7-14700 CPU, 64 GB RAM, and NVIDIA RTX A5000 GPU). An Epson EH-TW6250 projector featuring 4k PRO-UHD resolution, a luminous flux of 2800 lumens, and a response time below 20 ms, was used to display the interface on furniture surfaces. The application components and networked devices communicated through a local 5 GHz wireless network. This setup supported motion tracking, video projection, and gesture processing within a calibrated working area, sufficient for our demonstrative purposes.

System Architecture

The system implements a client-server architecture. The client was developed using HTML, CSS, and JavaScript, while an asynchronous backend process was implemented in Python to track motion data, recognize interactions, and communicate with external devices. Client-server communication uses a bidirectional WebSocket, while a separate WebSocket supplies Vicon tracking frames to the backend. The client reports its viewport size, calibration points, interaction regions, and graphical controls. Control and region boundaries are represented as polygons in client coordinates. For each processed Vicon frame, the backend maps the 3D tracked left- and right-hand positions to 2D coordinates. It associates hands with controls through point-in-polygon testing and stores trajectory histories for detecting specific user interactions, such as taps, directional swipes, dwell, and bimanual input.

Calibration and Interface Mapping

Calibration enables the correspondence between the furniture surface and the projected client interface. This process is separated into physical and visual calibration so that changes to the projection or interface layout do not require the Vicon setup to be recalibrated. In a first step, four reflective markers are placed at the corners of the interaction surface, ordered clockwise from the top-left corner (Figure 2, left). The calibration utility collects 60 Vicon frames, a sample size

¹<https://www.vicon.com/hardware/cameras/vero>

²<https://www.vicon.com/software/nexus>

Algorithm 1 Maps a tracked hand position to video-projected interface elements.

Require: $\mathcal{P} = (P_{00}, P_{10}, P_{11}, P_{01})$, representing a list of ordered 3D physical surface corners.

Require: $\mathcal{Q} = (Q_{00}, Q_{10}, Q_{11}, Q_{01})$, client calibration corners

Require: H is the tracked 3D hand position, C represents the client polygons, and ε the calibration tolerance.

Ensure: X as the client position, δ , representing the surface distance, and I as the list of intersected polygons.

```

1:  $\mathbf{a} \leftarrow P_{10} - P_{00}$ 
2:  $\mathbf{b} \leftarrow P_{01} - P_{00}$ 
3:  $(u, v) \leftarrow \text{PLANECOORDINATES}(H - P_{00}, \mathbf{a}, \mathbf{b})$ 
4:  $\mathbf{S} \leftarrow P_{00} + u\mathbf{a} + v\mathbf{b}$ 
5:  $\hat{\mathbf{n}} \leftarrow \text{NORMALIZE}(\mathbf{a} \times \mathbf{b})$ 
6:  $\delta \leftarrow (H - \mathbf{S}) \cdot \hat{\mathbf{n}}$ 
7: Apply axis correction to  $(u, v)$ 
8:  $\mathbf{G} \leftarrow \text{UNITSTOQUADRILATERAL}(\mathcal{Q})$ 
9: if  $\mathbf{G}$  is valid and  $|g_{31}u + g_{32}v + 1| > \varepsilon$  then
10:    $\mathbf{X} \leftarrow \text{PROJECTIVEDIVIDE}(\mathbf{G}[u \ v \ 1]^T)$ 
11: else
12:    $\mathbf{X} \leftarrow Q_{00} + u(Q_{10} - Q_{00}) + v(Q_{01} - Q_{00})$ 
13: end if
14:  $I \leftarrow \{C \in C \mid \text{POINTINPOLYGON}(\mathbf{X}, C)\}$ 
15: return  $(\mathbf{X}, u, v, \delta, I)$ 
    
```

Notes. PLANECOORDINATES computes the normalized coordinates of the hand projection in the planar basis defined by $\mathbf{a}$ and $\mathbf{b}$. NORMALIZE converts a vector to unit length. UNITSTOQUADRILATERAL constructs the homography that maps $(0, 0)$, $(1, 0)$, $(1, 1)$, and $(0, 1)$ to the four client calibration corners in $\mathcal{Q}$. PROJECTIVEDIVIDE converts the homogeneous mapped point (x, y, w) to client coordinates $(x/w, y/w)$. The POINTINPOLYGON function tests whether the mapped client position lies inside a control or polygon, using the same geometry reported by the client. Axis corrections represent the configured swaps or inversions applied to align the tracked surface orientation with the projected interface.

determined empirically, and averages the coordinates associated with each marker to reduce tracking noise. The resulting positions are used to approximate the furniture region as a plane originating at P_{00} (Figure 2, middle). Its longitudinal and latitudinal directions are obtained from the two adjacent corners, as follows: $\mathbf{a} = P_{10} - P_{00}$, $\mathbf{b} = P_{01} - P_{00}$. This representation is appropriate for surfaces that can be approximated by a plane (the current implementation does not attempt to model arbitrary curved or non-planar geometry).

Visual calibration takes place at the client level. Four draggable points, denoted in the following by Q_{00} , Q_{10} , Q_{11} , and Q_{01} , delimit the quadrilateral of the projected interface; Figure 2 illustrates this process on a modular sofa. The points are stored as normalized viewport coordinates in a reusable profile, together with the interaction regions. During operation, the client converts this geometry to pixels and reports its viewport, calibration quadrilateral, interaction-region polygons, and active-control polygons to the server over WebSocket.

For each tracked hand position H , represented by a 3D point, the server estimates where the hand projects onto the calibrated surface and computes a location relative to the surface origin as a combination of surface directions:

$$H - P_{00} \approx u\mathbf{a} + v\mathbf{b}$$

The corresponding point on the calibrated plane is $S(u, v) = P_{00} + u\mathbf{a} + v\mathbf{b}$. The server also computes the distance δ between the hand and the plane, as follows:

$$\delta = (H - S(u, v)) \cdot \hat{\mathbf{n}}, \quad \hat{\mathbf{n}} = \frac{\mathbf{a} \times \mathbf{b}}{\|\mathbf{a} \times \mathbf{b}\|}$$

This distance is applied to distinguish contact actions on the surface from gestures performed above it. The server maps (u, v) to the client position $X = (x, y)$ using a transformation from the unit square to the calibrated client quadrilateral:

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} g_{11} & g_{12} & g_{13} \\ g_{21} & g_{22} & g_{23} \\ g_{31} & g_{32} & g_{33} \end{bmatrix}}_{\mathbf{G}} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

In this equation, $\mathbf{G}$ represents the closed-form unit-square-to-quadrilateral homography determined by the four correspondences $(0,0)$ maps to Q_{00} , $(1,0)$ to Q_{10} , $(1,1)$ to Q_{11} , and $(0,1)$ to Q_{01} , respectively. The implementation fixes g_{33} to 1, and computes the remaining eight coefficients from these corner correspondences. The client position (x, y) is obtained as follows:

$$x = \frac{g_{11}u + g_{12}v + g_{13}}{g_{31}u + g_{32}v + 1}, \quad y = \frac{g_{21}u + g_{22}v + g_{23}}{g_{31}u + g_{32}v + 1}$$

This mapping accounts for perspective skew in the projected area. If the four points produce a singular transformation, or if the homogeneous denominator approaches zero, the server falls back to the affine mapping:

$$\mathbf{X} = Q_{00} + u(Q_{10} - Q_{00}) + v(Q_{01} - Q_{00})$$

If the client has not supplied calibration points, the normalized coordinates are scaled to the viewport. Finally, the server associates the mapped hand's coordinates with the current graphical interface. For each polygon reported by the client, a ray-casting point-in-polygon test counts horizontal edge crossings from X , with an odd number of crossings representing the criterion to determine the hand inside that polygon. The server returns the (x, y) position, normalized surface coordinates, distance δ , containment state, and intersected controls, which are then used by the client for gesture input detection. Algorithm 1 presents the procedure for aligning the video-projected interface with a given physical surface, represented as a quadrilateral, and detecting user input.

TECHNICAL EVALUATION

The evaluation was conducted with one participant, who performed 100 gesture trials. Since the focus of the study was the system's technical performance (e.g., response latency), not inter-user variability, only valid input data were required at this stage; thus, a single participant was sufficient for our purposes. Five gesture types, informed by prior work [1, 3, 6, 7, 9], were performed 20 times each: unimanual and bimanual hover, swipe, tap, and transfer of digital content across different physical surfaces. We found that all gestures were detected correctly (100%), without missed or false detections, misclassifications, or incorrect target selections.

System response latency was measured from server-side gesture detection until the client application confirmed that the resulting graphical interface update had passed through two animation frames. The average system latency measured across the 100 trials was 26.4 ms (SD=5.1, CI_{95%}=[25.5, 27.3]). Per-gesture analysis indicated similar results, as follows: taps averaged at 26.7 ms (SD=5.1, CI_{95%}=[24.5, 29.0]), swipe at 25.8 ms (SD=5.3, CI_{95%}=[23.5, 28.1]), unimanual hover gestures at 26.3 ms (SD=6.0, CI_{95%}=[23.7, 29.0]), content movement at 28.4 ms (SD=4.3, CI_{95%}=[26.5, 30.3]), and bimanual hover gestures at 25.4 ms (SD=4.1, CI_{95%}=[23.6, 27.2]).

Task-completion time was measured from displaying the start trial cue until confirmation that the interface state had been rendered. It therefore includes gesture execution, recognition, communication, and client response. The average task time was 2.1 s (SD=0.7, CI_{95%}=[2.0, 2.2]). Tapping was the fastest interaction, averaging 1.4 s (SD=0.1, CI_{95%}=[1.3, 1.4]), followed by swipe at 1.9 s (SD=0.6, CI_{95%}=[1.6, 2.1]), and unimanual hover at 1.9 s (SD=0.1, CI_{95%}=[1.8, 1.9]). Movement of digital content across physical surfaces averaged 2.6 s (SD=0.2, CI_{95%}=[2.5, 2.6]) because it combined selection with movement into another region. Bimanual hover gestures were the slowest, at 3.0 s (SD=0.2, CI_{95%}=[3.0, 3.2]), because both hands had to remain airborne in separate regions for the configured dwell interval.

Limitations

The evaluation characterized technical behavior of our system implementation under controlled laboratory conditions. It was not intended as a user study and, consequently, it does not address aspects of perceived usability, comfort, cognitive load, or acceptance of the proposed system and interactions. The trials also used a limited set of gestures and surfaces, so the results cannot be directly generalized to other furniture geometries or deployment contexts. Furthermore, the current implementation depends on Vicon motion tracking and approximates the interaction surface as a plane defined by four calibration points. This provides accurate tracking for approximately planar surfaces, but it limits deployment for irregular surfaces. Future work should consider alternative motion sensing technology for user input, system performance on additional surface geometries, and end-to-end latency including the video projector and user response.

CONCLUSION

We presented a system, developed using web technologies, for enabling interaction with digital content on everyday furniture surfaces via video projection. The system aligns projected graphical controls, such as buttons and navigation lists, with tracked hand input through surface calibration and homography mapping, enabling taps, swipes, dwell, and motion-based gestures. The preliminary technical evaluation showed consistent performance under controlled conditions with an average system response time of 26.4 ms. Future work will evaluate the system in a user study involving perception, preference,

performance, and cognitive-based measures, and explore additional surface geometries and gesture-sensing technologies.

ACKNOWLEDGMENTS

This work was supported by a grant of the Ministry of Education and Research, CCCDI-UEFISCDI, project no. PN-IV-PCB-RO-MD-2024-0130, within PNCIDI IV.

References

- [1] Alexandru-Tudor Andrei, Laura-Bianca Bilius, and Radu-Daniel Vatavu. 2024. Take a Seat, Make a Gesture: Charting User Preferences for On-Chair and From-Chair Gesture Input. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems (CHI '24)*. ACM, New York, NY, USA, Article 555, 17 pages. doi:10.1145/3613904.3642028
- [2] Alexandru Dancu, Velko Vechev, Advije Ayça Ünlüer, Simon Nilsson, Oscar Nygren, Simon Eliasson, Jean-Elie Barjonet, Joe Marshall, and Morten Fjeld. 2015. Gesture Bike: Examining Projection Surfaces and Turn Signal Systems for Urban Cycling. In *Proceedings of the 2015 International Conference on Interactive Tabletops & Surfaces (ITS '15)*. ACM, New York, NY, USA, 151–159. doi:10.1145/2817721.2817748
- [3] Chris Harrison, Hrvoje Benko, and Andrew D. Wilson. 2011. OmniTouch: Wearable Multitouch Interaction Everywhere. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST '11)*. ACM, New York, NY, USA, 441–450. doi:10.1145/2047196.2047255
- [4] Richard Hartley and Andrew Zisserman. 2011. *Multiple View Geometry in Computer Vision* (2nd ed.). Cambridge University Press, UK. doi:10.1017/CBO9780511811685
- [5] Brett Jones, Rajinder Sodhi, Michael Murdock, Ravish Mehra, Hrvoje Benko, Andrew Wilson, Eyal Ofek, Blair MacIntyre, Nikunj Raghuvanshi, and Lior Shapira. 2014. RoomAlive: Magical Experiences Enabled by Scalable, Adaptive Projector-Camera Units (UIST '14). ACM, New York, NY, USA, 637–644. doi:10.1145/2642918.2647383
- [6] Pranav Mistry, Pattie Maes, and Liyan Chang. 2009. WUW - wear Ur world: A Wearable Gestural Interface. In *Proceedings of the CHI '09 Extended Abstracts on Human Factors in Computing Systems (CHI EA '09)*. ACM, New York, NY, USA, 4111–4116. doi:10.1145/1520340.1520626
- [7] Radu-Daniel Vatavu, Laura-Bianca Bilius, Alexandru-Tudor Andrei, Mihail Terenti, Adrian-Vasile Catană, and Alexandru-Ionuundefined undefinediean. 2024. ChairMX: On-Chair Input for Interactive Media Consumption Experiences for Everyone, Everywhere. In *Proceedings of the 2024 ACM International Conference on Interactive Media Experiences (IMX '24)*. ACM, New York, NY, USA, 447–451. doi:10.1145/3639701.3661090
- [8] Andrew D. Wilson. 2005. PlayAnywhere: A Compact Interactive Tabletop Projection-Vision System. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology (UIST '05)*. ACM, New York, NY, USA, 83–92. doi:10.1145/1095034.1095047
- [9] Andrew D. Wilson and Hrvoje Benko. 2010. Combining Multiple Depth Cameras and Projectors for Interactions On, Above and Between Surfaces. In *Proceedings of the 23rd Annual ACM Symposium on User Interface Software and Technology (UIST '10)*. ACM, New York, NY, USA, 273–282. doi:10.1145/1866029.1866073
- [10] Robert Xiao, Chris Harrison, and Scott E. Hudson. 2013. WorldKit: Rapid and Easy Creation of Ad-hoc Interactive Applications on Everyday Surfaces. In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 879–888. doi:10.1145/2470654.2466113
- [11] Stefanie Zollmann, Tobias Langlotz, and Oliver Bimber. 2007. Passive-Active Geometric Calibration for View-Dependent Projections onto Arbitrary Surfaces. *Journal of Virtual Reality and Broadcasting* 4 (2007), 10 pages. doi:10.20385/1860-2037/4.2007.6