Structured Light Speckle: Joint egocentric depth estimation and low-latency contact detection via remote vibrometry

Paul Streli
Department of Computer Science
ETH Zürich, Switzerland

Jiaxi Jiang
Department of Computer Science
ETH Zürich, Switzerland

Juliete Rossie
Department of Computer Science
ETH Zürich, Switzerland

Christian Holz
Department of Computer Science
ETH Zürich, Switzerland

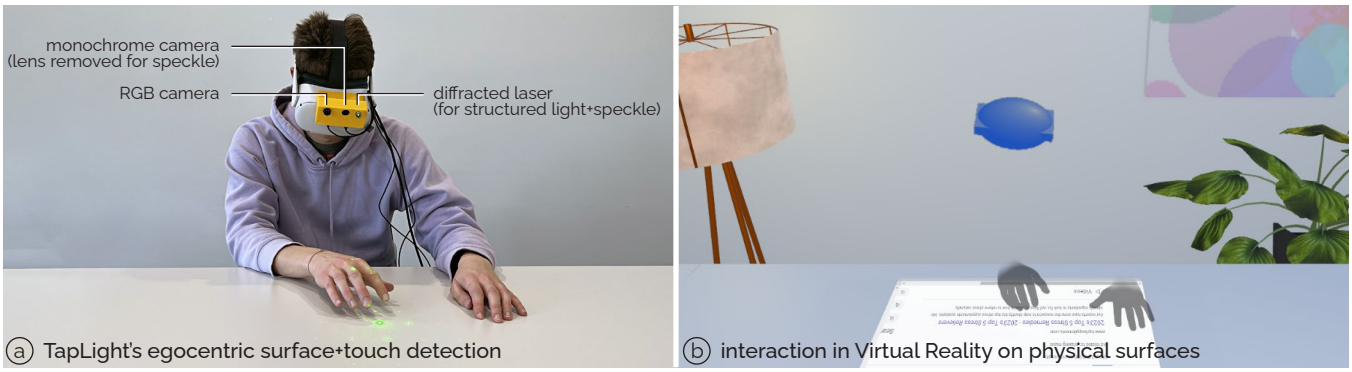

Figure 1: *TapLight* is an egocentric remote contact sensing system that *simultaneously* discovers physical surfaces and moments of touch through a novel integration of structured light and laser speckle: (a) The diffracted laser creates a sparse dot pattern on the surface, producing binocular disparity in the RGB camera's view, from which we estimate depth values and fit a plane. The monochrome camera's exposed sensor captures the interference of diffused laser reflections, and its high framerate reveals the remote vibrations propagating through both, the hand and the surface upon contact. TapLight combines detected surfaces with the tracked hand poses from the VR headset, verifies and determines touch locations, and relays them into Virtual Reality (b).

ABSTRACT

Despite advancements in egocentric hand tracking using head-mounted cameras, contact detection with real-world objects remains challenging, particularly for the quick motions often performed during interaction in Mixed Reality. In this paper, we introduce a novel method for detecting touch on discovered physical surfaces purely from an egocentric perspective using optical sensing. We leverage structured laser light to detect real-world surfaces from the disparity of reflections in real-time and, *at the same time*, extract a time series of remote vibrometry sensations from laser speckle motions. The pattern caused by structured laser light reflections enables us to simultaneously sample the mechanical vibrations that propagate through the user's hand and the surface upon touch.

We integrated *Structured Light Speckle* into TapLight, a prototype system that is a simple add-on to Mixed Reality headsets. In our evaluation with a Quest 2, TapLight—while moving—reliably detected horizontal and vertical surfaces across a range of surface materials. TapLight also reliably detected rapid touch contact and robustly discarded other hand motions to prevent triggering spurious input events. Despite the remote sensing principle of Structured Light Speckle, our method achieved a latency for event detection in realistic settings that matches body-worn inertial sensing without needing such additional instrumentation. We conclude with a series of VR demonstrations for situated interaction that leverage the quick touch interaction TapLight supports.

UIST '23, October 29–November 1, 2023, San Francisco, CA, USA

ACM ISBN 979-8-4007-0132-0/23/10…$15.00
https://doi.org/10.1145/3586183.3606749

CCS CONCEPTS

• **Computing methodologies** → *Reconstruction*; • **Human-centered computing** → **Graphics input devices**; **Mixed / augmented reality**; **Virtual reality**.

KEYWORDS

Virtual Reality, Mixed Reality, Augmented Reality, Touch Sensing, Laser Vibrometry, Surface Reconstruction, Structured Light.

ACM Reference Format:
Paul Streli, Jiaxi Jiang, Juliete Rossie, and Christian Holz. 2023. Structured Light Speckle: Joint egocentric depth estimation and low-latency contact detection via remote vibrometry. In *The 36th Annual ACM Symposium on User Interface Software and Technology (UIST '23), October 29–November 1, 2023, San Francisco, CA, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3586183.3606749

1 INTRODUCTION

Directly interacting with elements in Virtual Reality (VR) has numerous advantages over indirect techniques (e.g., ray cast [27] or cursor techniques [4]). Resembling how we manipulate objects in the real world, direct interaction with virtual mid-air objects in VR contributes to a stronger sense of presence [3]. Previous studies have also shown that direct interaction leads to efficient task completion in VR [11], allowing users to leverage their existing spatial [48] and motor skills [32]. It also leads to higher user engagement [55] and satisfaction [54], making the user experience seamless and enjoyable within the virtual world.

Direct interaction in VR additionally can be supplemented with tactile sensations, such as by appropriating passive physical objects and surfaces for input [2, 9, 53]. Recent work has shown that integrating even larger surfaces into VR, such as tables [35, 39, 47, 50], may additionally reduce fatigue during interaction [11, 52], as users can lean and rest on them. Augmenting virtual experiences with these physical affordances could thus support more efficient [68, 69] and longer work sessions [2] by reducing the strain on the body.

To detect touches on physical surfaces, previous approaches have augmented real-world surfaces with input tracking (e.g., capacitive sensing [14, 36] or multiple depth cameras [26, 58]). Alternatively, previous work directly equipped users with wearable sensors (e.g., rings [18, 23] and wristbands [35, 50]), which typically detect physical contact through inertial sensing.

While such sensor augmentations can detect touch interaction in VR, they increase system complexity, require low-latency communication, and need more elaborate manufacturing. In contrast, recent consumer devices integrate all system functionality into the headset as a single device, including inside-out tracking through multiple cameras that additionally track the user's hand poses for input in VR (e.g., Meta Quest 2 [34], VIVE Focus 3 [12]). Although it stands to reason that such camera setups may be capable of detecting touch input, precise detection of quick events on uninstrumented real-world surfaces is a substantial challenge in practice, particularly in mobile settings and egocentric systems [61].

In this paper, we introduce *structured light speckle*, a sensing method that reliably detects physical surfaces as well as touch events on them—even while in motion. The key element of our method is the integration of structured light from a laser source, as found in depth cameras (e.g., Kinect 1 or Intel RealSense [65], Face ID [56]), with remote vibrometry by sensing laser speckle from the interference of diffusely reflected laser rays. Our purely optical approach affords sensing from a single vantage point; this allows it to be integrated into a single wearable headset, which we demonstrate through our prototype system TapLight.

Structured light speckle → surfaces & touches

Figure 1a shows a user wearing a VR headset equipped with TapLight, interacting inside an immersive environment (b) where a browser is aligned with the physical table. The headset exposes TapLight's two additional embedded cameras and its diffracted laser emitter to implement structured light speckle sensing. Our system complements the headset's inside-out tracking inside world space and its hand pose tracking, supplying the location of discovered real-world surfaces as well as touch-input events on them. TapLight seamlessly triggers input events as they occur in the interactive environment, while the headset's tracking supplies the input location.

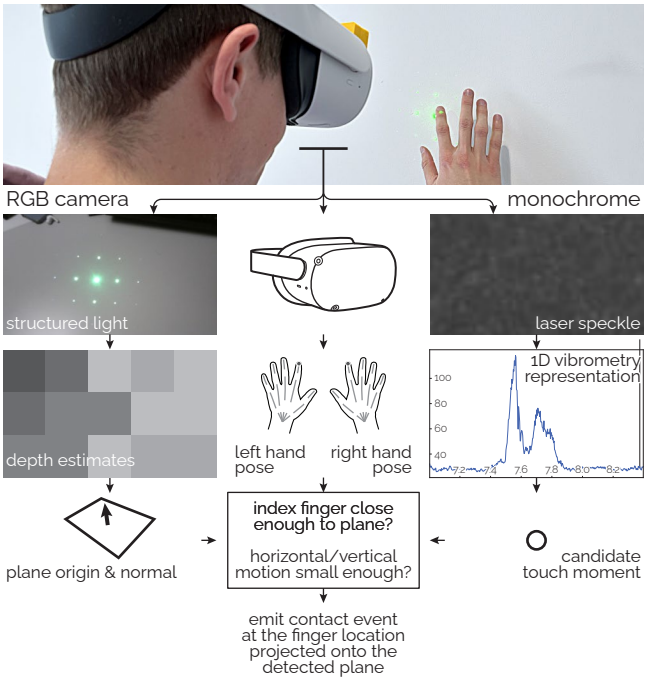

Figure 2: TapLight fuses cues from its own plane detection, moment of touch detection, and the headset-provided hand poses to trigger input events in VR.

Figure 2 illustrates TapLight's implementation of structured light speckle sensing. The diffraction grating added to the laser produces a grid of 12×7 rays with a constant angle in between individual rays. An RGB camera resolves diffuse laser reflections from real-world geometry, which our pipeline extracts and estimates depth from, thereby sparsely implementing structured light depth sensing. Our system then fits a plane to these observations and places it in world coordinates based on the headset's global pose.

At the same time, TapLight's monochrome global shutter camera with an exposed sensor captures the laser speckle from the diffused reflections, which we transform into a 280 Hz excitation signal to resolve remote vibrations that propagate through the user's hand upon touch as well as through the surface. Our system then extracts moments of touch, verifies touch events based on the headset's reported hand positions, velocities, and the previously detected physical surface, and passes validated events to the VR frontend.

We evaluated TapLight's detection of depth and touch in a controlled experiment. For depth, TapLight discovered planar surfaces with an error of 22 mm (horizontal) and 45 mm (vertical) within a range of 1 m, estimating surface normals with a mean accuracy of 6° (wall) and 2.8° (table). For touch detection, we evaluated TapLight's performance with 8 participants on a table, a wall, and a shelf to examine the impact of surface rigidity. TapLight accurately detected touch events ($F_1 = 0.95$), hardly reporting any false-positive input events as participants moved their hands in front of the headset.

Contributions

We make the following contributions in this paper:

(1) a novel non-contact depth+touch sensing method that supports mobile use, operating from a single vantage point, and, thus, adequate for the use in head-mounted devices. The key component of our approach is the simultaneous detection of surfaces from the binocular disparity between the reflections of diffracted laser rays and the recognition of touch events on them through motions in speckle patterns (i.e., remote vibrometry).
(2) an integrated system that fuses information about headset poses, hand poses, and detected surfaces into a joint 3D model of physical surroundings to provide low-latency touch detection and false-positive rejection of input events with a moving sensor.
(3) a technical evaluation of surface detection, showing promising accuracy for depth and touch detection for practical purposes.
(4) a set of demo applications that showcases the use of our real-time system in everyday settings to support the interaction in VR through passive haptic feedback.

Collectively, our contributions show the promise of touch interaction with physical affordances for interacting in VR, detected directly by the headset with no other wearable sensors or instrumentation of the environment. Our method outlines a path to supporting touch as the input modality people use with their devices on a daily basis, though in immersive scenarios and mobile settings.

2 RELATED WORK

The work in this paper is related to touch interaction in AR/VR, depth-based touch detection, and speckle-based sensing.

2.1 Direct (touch) interaction in AR/VR

Not least due to the availability of hand pose tracking on emerging Mixed Reality headsets (e.g., Quest [34], HoloLens [13], or VIVE Focus [12]) is direct interaction in immersive environments becoming increasingly popular [3, 53]. Researchers have proposed a multitude of direct interaction techniques and shown their benefits over indirect interaction, though mostly for mid-air use.

In parallel, previous work has investigated direct interaction that leverages real-world affordances for better input control [28]. The added benefit of such interaction is the complementary passive haptic feedback from physical objects [9, 47, 53]. To track touches in such scenarios, previous work has used external tracking systems [9, 58], which are adequate for stationary VR setups. More portable solutions have opted for wearable approaches that directly capture the sensations of contact with surfaces, such as acoustic sensors [44] or inertial measurement units (IMUs) placed on the user's fingers [23, 45, 51], wrists [42, 63], or on the surface using a wrist-worn interface [19]. Common to these approaches is that they detect the mechanical vibration waves that originate as users make physical contact with a surface.

2.2 Depth & depth-based touch sensing

Depth cameras have frequently been used for sensing touch input at scale. With the arrival of the Kinect [31], depth cameras became commodity sensors to detect the distance of objects in view and segment them [58]. Kinect implements structured light sensing, detecting the binocular disparity of an emitted light pattern of a large number of points [65], reporting depth at a sufficient resolution to detect touch input in stationary setups by applying a threshold above the background [57] or explicitly detecting fingers on surfaces [25]. Depth sensors have spurred much research on touch sensing and accuracy, including for location and event detection, and remain actively investigated in the community [7, 16, 24, 40, 43].

Achieving high accuracy in touch detection requires input sensing with high stability, positional accuracy, and reliable touch segmentation for detection. However, methods based on depth cameras face the challenge to provide this level of accuracy due to limited depth resolution and noise characteristics [60]. Xiao et al. discussed the main challenges in their evaluation of depth-based touch sensing, reporting the rate of missed touches and spurious extra touches to cause the largest issues [61]. To increase the resolution of depth-based touch detection and decrease its noise level, several approaches have been proposed integrating additional sensors. For example, DIRECT processes both depth and infrared data to detect fingertip positions with just a 5 mm error [60]. MRTouch extends this technique to the time-of-flight depth camera inside a Microsoft HoloLens [61]. Dante's multi-modal sensing approach combines depth and thermal imaging to increase stability [40]. Although optimized to detect touch positions, both DIRECT and Dante exhibit latency issues, which recent learning-based approaches have shown to decrease to interactive rates (e.g., 70 ms [16]).

In principle, structured light speckle can operate based on the technical components inside the first Kinect. While Kinect's design focused on depth estimation, it already incorporated an infrared laser that passed through a diffractive optic element to produce a dot pattern—yet Kinect's implementation was oblivious to the powerful signal caused by the laser speckle sensations that invisibly emerged during hand-object interaction events such as surface touches.

2.3 Speckle-based sensing

Laser speckle results from the diffused material reflections of laser beams, which creates interference patterns on the image sensor. The phenomenon has been explored for interactive purposes such as touch or motion detection on surfaces without instrumenting the user or the surface [38, 66, 70]. Laser vibrometry can reliably operate over a large distance to detect vibrations on physical objects and surfaces, even at city scale [67]. Since reflected speckle patterns depend on the material's surface structure, the sensing approach is also useful for texture classification [15, 41, 62].

Speckle sensing takes advantage of the collimated and coherent properties of lasers, which provide high sensitivity and signal-to-noise ratio [46]. Especially when high-speed cameras are used to resolve speckle motion, previous methods have been able to preserve and analyze the spatial correlations between speckle patterns between frames when objects are in motion [70]. For example, ForceSight detected laser speckle shifts due to object deformation under force for non-contact force sensing in a stationary setup [38], which yields continuous levels of pressure.

Our method builds on prior approaches to speckle-based sensing; we not only fuse it with structured light sensing for simultaneous depth estimation but, importantly, adapt speckle sensing for the use in a mobile and moving platform for remote vibrometry.

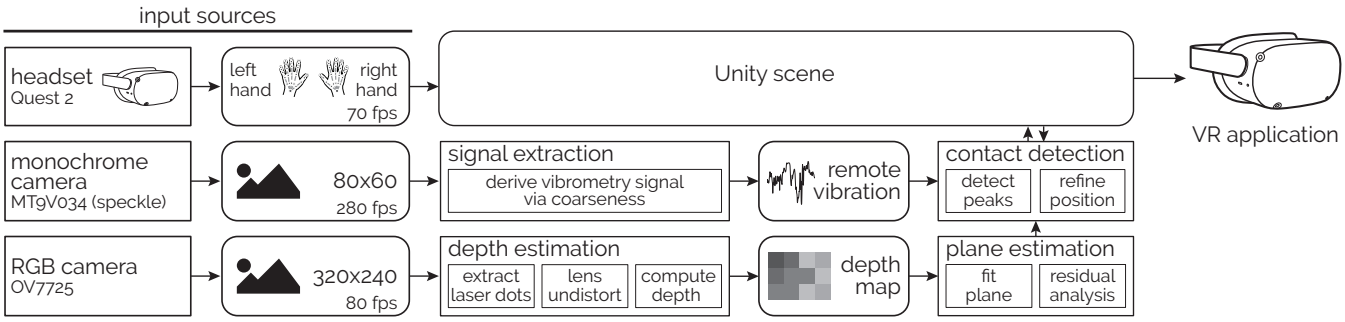

Figure 3: TapLight's implementation of structured light speckle sensing, including components and exchanged data objects.

3 METHOD & TAPLIGHT'S IMPLEMENTATION

Structured light speckle fuses two parts to simultaneously detect real-world objects as well as the events of contact with them: structured light-based depth estimation from individual laser reflections and reflection-based speckle motions that reveal remote vibrations.

As shown in Figure 3, TapLight implements our method using a diffracted laser to produce a grid of laser reflections for sparse structured light-based depth estimation and speckle-based motion detection. TapLight runs interactively and detects surfaces around the user by fitting a plane to the sparse set of depth estimations, locating touches on them in real time upon detecting a contact event in the speckle-based vibrometry signal.

Figure 1a shows our prototype, which comprises two embedded platforms with a camera on each board for sparse depth estimates and contact detection, respectively. The signal stimulus from a 532 nm laser is diffracted into the grid of rays by a grated plastic layer that we extracted from inexpensive diffraction glasses. While TapLight would equally work with an infrared laser (e.g., those in Kinect or Intel RealSense), we selected a green laser for ease of system development and debugging.

3.1 Depth estimation from structured laser light

Our method first estimates the depth distance for each projected point in the diffracted laser grid. It takes an image of the emitted laser grid as input. We leverage the binocular disparity between the emitter and the camera, which causes reflections in the image to shift to an extent that is inversely proportional to the depth distances as shown in Figure 4.

Camera interface. For depth sensing, TapLight features an embedded RGB camera (OmniVision OV7725) that we configured to output frames at QVGA resolution (320 px × 240 px) at 80 fps. The camera's field of view measures 63.5°, which we fully leverage for depth sensing. The distance (i.e., baseline) between the laser and the camera is 65 mm, optimizing parallax and thus depth resolution while fitting TapLight's case between the headset's own cameras.

Extract laser ray reflections. Due to the contrast between the laser reflections and the typically uniformly colored surfaces, TapLight adjusts the camera's exposure time so as to darken most of the background and highlight the diffused laser pattern. We adaptively threshold the image to detect connected pixels as a mask. Their center of mass then yields the coordinates in the original image.

When TapLight is located too close to a surface, the laser reflection may exhibit noise surrounding the center. The noise distribution is typically symmetric, such that the center of mass remains accurate. However, as the user moves away from a surface, reflected intensities become weaker. We defined a minimum of five reflections that TapLight needs to extract before advancing in its processing pipeline to detect physical surfaces.

Undistort extracted laser reflections. During runtime, TapLight applies the intrinsic parameters to all extracted reflection coordinates to correct for lens distortion during our depth estimation. We obtain the intrinsics using a one-time camera calibration procedure as follows: Using an 8 × 7 checkerboard pattern with a known square length, we captured multiple images from various angles and distances to cover the camera's full field of view. Through corner detection during post-processing, we extracted the pattern's corners across images and derived the intrinsic parameters of the camera, such as the focal length, the principal point, and the lens distortion coefficients using OpenCV's calibration routine [5].

Derive depth estimates from disparity. TapLight extracts the diffused reflections' 3D coordinates in the coordinate system of the camera. Starting with their 2D pixel coordinates in the captured images, we leverage the binocular disparity (i.e., the length of the baseline l) between the laser emitter and the RGB camera as shown in Figure 4.

We compute the orthogonal projection distance d from a diffused reflection point P onto the baseline (see Figure 5),

$$\begin{aligned} l &= l_1 + l_2 \\ &= d\tan(\alpha) + d\tan(\beta). \end{aligned} \tag{1}$$

Here, l is the length of the baseline and α is the laser beam's angle of diffraction. l_1 and l_2 are the distances from the laser emitter and the RGB camera to the orthogonal projection of P onto the baseline, respectively. The distance depends on the horizontal angle β of the laser grid's center pixel location to the center point C in the captured RGB image.

We obtain β by relating the RGB camera's field of view ϕ to the number of pixels w_c from the image center C to the image edge and the number of pixels w_p from the image edge to the pixel of P in the captured image,

$$\tan(\beta) = \frac{w_c - w_p}{w_c} \tan\left(\frac{\phi}{2}\right). \tag{2}$$

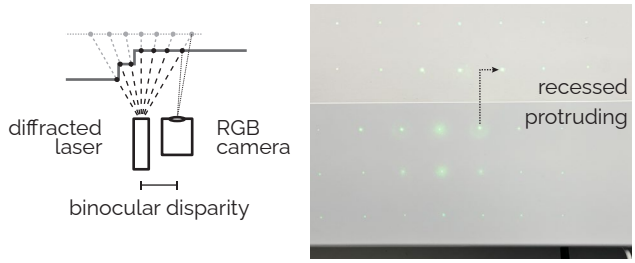

Figure 4: TapLight implements structured light-based depth sensing, using the grid of diffused reflections from the diffracted laser emitter to estimate a sparse depth map.

Using Equation 1 and Equation 2, we compute d as

$$d = \frac{l}{\tan(\alpha) + \tan(\beta)} = \frac{l}{\tan(\alpha) + \frac{w_c - w_p}{w_c} \tan\left(\frac{\phi}{2}\right)}. \tag{3}$$

To improve the accuracy of depth estimates, our one-time calibration obtains the defraction angle α using a known distance d_α:

$$\alpha = \arctan\left(\frac{l}{d_\alpha} - \frac{w_c - w_{p,\alpha}}{w_c} \tan\left(\frac{\phi}{2}\right)\right). \tag{4}$$

To estimate the depth of any point in the detected diffraction pattern, we associate it with the diffraction angle α of its corresponding laser beam. We start with the central reflection that originates from the one non-diffracted ray that occurs at the center of all reflected points (and often with higher intensity). We then link the detected reflections to their corresponding α-values of the diffracted laser rays based on their offset to the central reflection. TapLight's diffraction grating produces an angle α of 4.9° between the laser beams, which we verified during calibration.

In a final step, we transform each reflection point to the 3D world coordinate system. For this, we add the corresponding 3D offset to the 6D pose of the camera, which is rigidly mounted to the VR headset and thus tracked within the environment.

Fit a plane to the sparse depth observations. To finish the discovery of potential physical touch surfaces in the environment, we make use of the fact that a plane is defined by three non-collinear points. We thus fit a 3D plane for each frame using the plane equation: $ax + by + c = z$. Given n points with 3D coordinates $[x_i, y_i, z_i]$:

$$\begin{bmatrix} x_0 & y_0 & 1 \\ x_1 & y_1 & 1 \\ & \cdots & \\ x_n & y_n & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} z_0 \\ z_1 \\ \cdots \\ z_n \end{bmatrix} \tag{5}$$

We can rewrite this as $\mathbf{Ax} = \mathbf{B}$ where $\mathbf{x}$ represents the coefficients of the plane. Since TapLight detects a minimum of 5 points in each valid frame, this system is over-determined. We thus use the left Moore–Penrose inverse to obtain the plane coefficients as

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{B}. \tag{6}$$

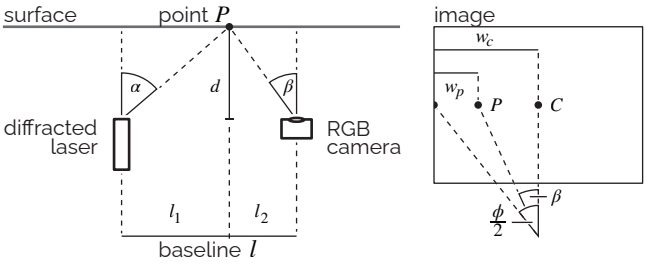

Figure 5: Triangulation with a camera and a laser emitter to estimate depth. Left: We calculate the depth d of a point in the scene using the baseline length l, the laser beam's diffraction angle α, and the angle β. Right: The horizontal angle β results from the horizontal position of the point in the camera image and the camera's field of view ϕ.

To assess the goodness of fit for the estimated plane, we calculate the residual error e as

$$e = |\mathbf{B} - \mathbf{Ax}| = |\mathbf{B} - \mathbf{A}\left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{B}|. \tag{7}$$

Finally, we conclude that the laser reflection pattern lies on a surface if the residual between the points and the plane is smaller than a threshold value τ. We empirically established τ to be 30 mm.

3.2 Contact detection from speckle vibrometry

To detect touch contact, our method analyzes the temporal changes in the laser speckle pattern that emerges when laser beams intersect with the user's hand or a physical surface. In such cases, their diffused reflections interfere and are subsequently captured as textured frames by a camera's exposed sensor. Moments of contact induce temporal effects within this speckle pattern, which manifest as distinctive peaks in coarseness and, thus, provide an effective means to remotely resolve object vibration.

Camera interface. For contact detection, TapLight integrates an embedded global-shutter grayscale camera (Onsemi MT9V034) with an exposed sensor. Removing the lens but leaving the case intact limits incident light to a 62°, which conveniently renders it oblivious to motions outside the user's field of view. The camera is configured to output frames at a resolution of 80 px × 60 px at 280 Hz, which allows TapLight to resolve minute input sensations from vibrations of up to 140 Hz—the frequency range that covers the meaningful band of tap and touch events [23, 35].

Extract a 1D vibrometry signal from speckle. TapLight senses remote vibrations from the variations in the captured speckle across time. We reduce the 2D speckle images over time into a 1D vibrometry time series via the dissimilarity DSS(δ) [59]. This computes the mean absolute intensity difference for various inter-sample spacing distance vectors $\delta = (\Delta x, \Delta y)$,

$$DSS(\delta) = \mathbf{E}\{|I(x, y) - I(x + \Delta x, y + \Delta y)|\},$$

where $I(x, y)$ is the intensity value at pixel (x, y) and $\mathbf{E}$ is the expectation operator over all pixels of the image. We then construct a statistical feature matrix $\mathbf{M} \in \mathbb{R}^{(L+1)\times(2L+1)}$ where the

element (i, j) computes the dissimilatrity for the inter-sample distance $\delta = (j - L, i)$. The coarseness F_{crs} for each frame is then

$$F_{\text{crs}} = 1/\sum_{(i,j)\in N_r} DSS(i, j)/n,$$

where $N_r = \{(i, j) : |i|, |j| \leq L\}$ and n is the number of elements in the set N_r. We set $L = 4$ to analyze changes in the texture.

Identify candidate moments of physical contact. Making physical contact causes rapid changes in the coarseness signal through which TapLight detects such events. We apply Pan-Tompkins [37] for peak detection on the band-pass filtered coarseness signal (lower cutoff: 0.05 Hz, upper cutoff: 105 Hz) and square the forward derivative of the filtered signal. Peaks above a threshold are extracted using a moving window integration with a window length of 35 ms. To prevent the repeated detection of the same contact event, we set a minimum inter-peak distance of 0.5 s.

Detect actual touch events. Finally, our touch detection verifies the presence of an actual touch event and, if confirmed, forwards it to the VR frontend. As input, our touch detection takes the hand poses reported by the headset in world coordinates, the physical surfaces TapLight has previously discovered (Section 3.1), as well as candidate moments of physical contact, each of which triggers this process. For each candidate moment, we verify that one of the two index finger positions reported by the headset in world coordinates is within a maximum distance to a plane that our method has discovered before. We reject touch event candidates if the lateral velocity of the user's hand is too large, since upon touch the magnitude of the hand's motion towards a surface is considerably larger than its lateral velocity.

If both conditions are satisfied for a finger, TapLight confirms the touch event. For increased touch accuracy, we refine the input location reported by the headset by projecting the headset-supplied 3D coordinate of the tracked finger onto the previously discovered plane following the user's current gaze vector. TapLight then emits the input event in the VR app.

3.3 System integration

TapLight is powered by an 8-core Intel Core i7-9700K CPU at 3.6 GHz to run its tracking pipeline, signal processing, and message exchange with the VR frontend. All signal processing, for depth and touch alike, runs on the CPU in real-time thanks to the low resolution of our input images. The PC's NVIDIA GeForce RTX 3090 GPU exclusively rendered the VR environment, but it was not necessary for TapLight's operation.

3.4 Virtual Reality

TapLight interfaces with VR frontends implemented using Unity 2021 for the Oculus Quest 2 VR headset. We used web sockets for communication between our tracking system and the VR apps. VR apps receive the hand poses from the Quest 2 at ~70 Hz and forward it to TapLight's processing pipeline along with the headset's global 6D pose in world coordinates. TapLight's processing backend detects input from our sensing subsystems, maintains the list of detected surfaces as well as events on them, and transmits updates and interactions to the VR environment.

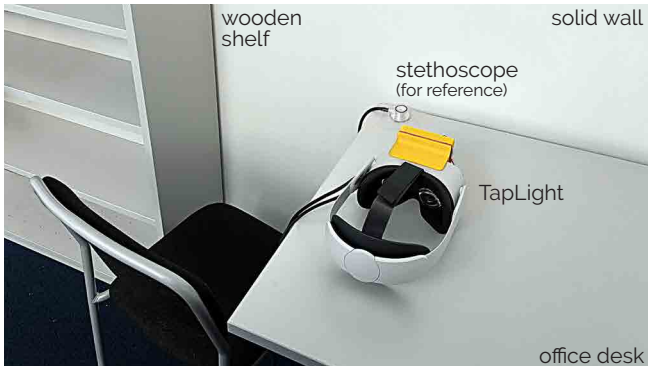

Figure 6: In our technical evaluation, participants provided touch events on these three surfaces. To evaluate TapLight's depth accuracy, the headset was additionally equipped with retroreflective Optitrack markers.

4 TECHNICAL EVALUATION

The purpose of this evaluation was to establish TapLight's efficacy in supporting direct touch input on physical surfaces surrounding a user for interaction in VR. We studied TapLight's accuracy in terms of surface detection and touch detection. Figure 6 shows our environment for both evaluations.

4.1 Depth estimation & surface detection study

In our first evaluation, we verified our sparse surface detection method. For this, we aimed to determine the accuracy of surface angles detected by our pipeline as well as the error of records captured across a single surface.

Procedure. Different materials exhibit distinguishable deformations in response to force due to variances in the density and internal microstructures [38]. To evaluate the performance of TapLight, we used multiple surfaces in our experiments as shown in Figure 6. Our surface test set included a solid wall, painted uniformly with acrylic paint during construction, a contemporary office desk (i.e., medium-density fiberboard (MDF) coated with melamine with a smooth and glossy surface), and a wooden shelf (sanded plywood with a cedar veneer). Testing our system on various surfaces enabled us to evaluate its ability to detect and distinguish between different types of touches on different materials, as well as its versatility and robustness across settings and applications.

Two experimenters conducted this evaluation to account for differences in speed of motion and incident angles. The experimenters held and pointed the VR headset at a variety of angles at the surfaces in 1-minute intervals with five repetitions each.

In a follow-up experiment, we repeated this procedure on two additional surface materials: cardboard and a reflective blue plastic overlay with a thickness of 1.5 mm. Each new surface type was placed on top of the table and the wall.

Interfaces. For reference, we tracked all surfaces and the headset itself with an 8-camera Optitrack Prime 13 system, which reported object positions and orientations with sub-mm accuracy. The cameras surrounded the markers attached to all surfaces and had an unobstructed view of the moving VR headset.

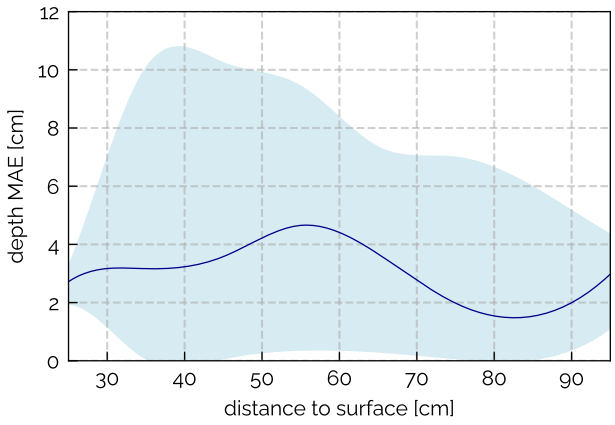

Figure 7: TapLight's mean depth estimation error for varying distances between the headset and a sensed surface point over the reconstructed table, shelf, and wall area (blue line). The blue background indicates the 95% confidence interval.

As the experimental interface, we used TapLight to emit structured laser light through our diffraction grating as described above, record diffused laser reflections through its RGB camera, record speckle using its monochrome camera, and process all data through the pipeline described in Section 3.

Results. We compared the distances between the VR headset and each detected surface between TapLight's surface estimates and the OptiTrack's recordings. We thereby evaluated the accuracy of our algorithm and determined the impact of surface types and angles on the accuracy of measurements.

We obtained a mean absolute error (MAE) of 21.9 mm (table), 45.1 mm (shelf), and 51.1 mm (wall) for the estimated depth distances. Our method reconstructed surface normals with an error of 2.77° (table), 13.8° (shelf), and 5.9° (wall).

Effect of additional surface materials. TapLight's reconstruction error for cardboard on top of the table and wall was 16.9 mm and 47.2 mm, respectively. For the plastic overlay, the error was 19.5 mm for the table and 51.6 mm for the wall.

Effect of distance between TapLight and surface. The mean depth error across the reconstructed surfaces of the table, shelf, and wall was stable and < 4.8 cm within 1 m (Figure 7). The error was 2.72 cm from 0.25 m, 4.65 cm from 0.55 m, and 2.97 cm from 0.95 m distance.

4.2 Contact detection study

We investigated the efficacy of TapLight's touch detection on several surface types. Since structured light speckle samples the motion and vibration of the user's hand upon touch but also the microvibrations propagating through the surface, signal magnitude and detection reliability may depend on the rigidity of the touch surface.

Task. Participants sat in an office chair, not leaning against its rest, and were surrounded by three surfaces: an office table (horizontal), a solid wall (vertical), and wooden shelf (vertical). Participants' task was to put on our TapLight prototype, which showed a virtual representation of the surface. The experimenter instructed participants to repeatedly touch the surface with an intensity and interaction speed similar to how they would interact with an iPad or public touchscreen. To touch, participants used their left index finger or their right index finger.

Procedure. The experimenter began this evaluation with a short introduction of TapLight and its purpose and then recorded participants' demographic and anatomic details. During the evaluation, the experimenter instructed participants on the finger to use for touch input as well as the surface to touch. Participants repeated 60 touches with one index finger, as instructed, and produced another 60 touches with the other index finger. Participants then moved on to another surface, completing all three. Each participant completed the evaluation in under 20 minutes.

We used a stethoscope as a surface microphone to record ground-truth moments of contact. The recorded audio signal shows distinct and easily detectable peaks that enable the reliable detection of tap events in environments without strong background noise.

Similar to our study of depth estimation, we evaluated TapLight's performance on other surface types in a follow-up study. Again, we evaluated it on the cardboard overlay as well as on the 1.5 mm thick piece of plastic.

Participants. We recruited 8 participants from places around our institution (3 female, 5 male, ages 23–36, mean=26.6 years). We measured three anatomical characteristics in participants to account for how their hands may absorb vibrations: 1) length of middle finger (70–89 mm, mean=81 mm), 2) height (158–190 cm, mean=174 cm), and 3) width of the middle finger at the distal joint (15–21 mm, mean=17.4 mm). Participants received a small gratuity for their time after the evaluation.

Results. We compared the touch events TapLight detected with the reference events from the surface microphone. Our method achieved a recall of 0.937 (SD=0.03) and precision of 0.977 (SD=0.01) across all participants, hence an F_1-score of 0.953 (SD=0.02). We considered a touch event detected by our method correct if it was the closest detected touch event to a reference event and the latency between the two events was smaller than 300 ms.

Effect of surface type. For touch events on the table, TapLight achieved a recall of 0.938, a precision of 0.976, and an F_1-score of 0.952. On the wall, the recall, precision, and F_1-score was 0.930, 0.994, and 0.960, respectively. On the shelf, our method detected taps with a recall of 0.943, precision of 0.961, and an F_1-score of 0.950. On cardboard, TapLight achieved a precision of 1.000, a recall of 0.926, and an F_1-score of 0.962. On the plastic overlay, precision was 0.963, recall was 0.867, and the F_1-score was 0.912.

Effect of distance between TapLight and surface. In another follow-up experiment, we evaluated the signal-to-noise ratio (SNR) at varying distances from the table. We computed SNR as the mean of the squared offset between the peaks of the moving window sum divided by the mean of the squared unfiltered noise during periods without touches. At a distance of 0.45 m from the table, the SNR was 44.9 dB (precision=1.000, recall=1.000, F_1-score=1.000). At a distance of 0.6 m, the SNR dropped to 32.1 dB, with a corresponding precision,

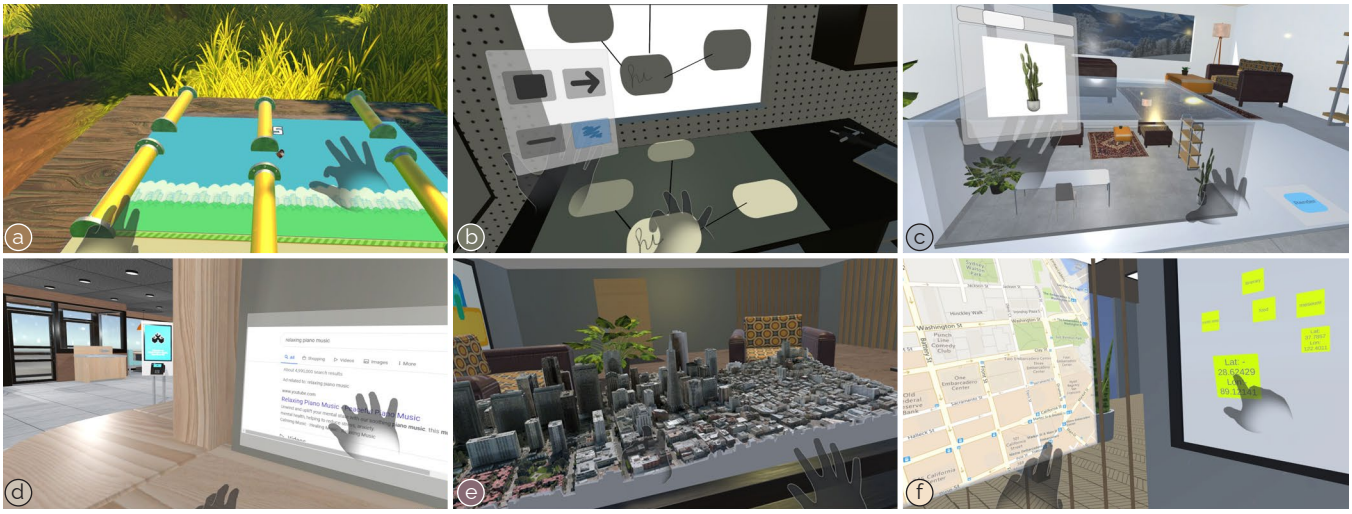

Figure 8: Our demos show TapLight's integration with VR environments to provide touch interaction with surface-aligned content: (a) Tappy Bird, (b) a diagram editor in a craft workshop, (c) a world-within-world room design app that renders the miniature setting around the user, (d) a web view embedded in VR with conventional touch support, (e) an interactive 2.5D city map with pan and zoom control, and (f) a 2D map for city planners that allows annotating locations on the whiteboard.

recall, and F_1-score value of 0.962, 1.000, and 0.981, respectively. At a distance of 0.9 m, the SNR decreased to 11.4 dB, with a precision, recall, and F_1-score value of 0.634, 0.765, and 0.693, respectively.

Latency of contact detection. TapLight's per-frame processing took ~2 ms and computing coarseness accounted for most of this. We used a 240 fps iPhone 12 Pro camera to measure the end-to-end latency from multiple touches (moment of contact to the computer display's response, including processing and communication). We found that TapLight's mean latency was 50.4 ms end-to-end.

4.3 Discussion

Our evaluation showed the efficacy of our method and TapLight's implementation during dynamic situations in a real-world office environment. As expected, surface types affected distance accuracy, which is likely due to surface finish and reflective properties.

Depth estimation. Most promisingly, our depth reconstruction and normal estimation of a horizontal surface produced the lowest errors. This supports our initial motivation that TapLight could support touch input on passive surfaces in immersive scenarios. Despite the sparsity of our sensing method, the low-resolution cameras, and the use of off-the-shelf laser and diffraction grating, TapLight achieved accuracies that are comparable to those obtained with stationary consumer depth cameras on horizontal surfaces.

Touch detection & latency. Also promising was our method's capability of detecting rapid touch contact with physical surfaces—reliably and with low latency in particular. TapLight's latency of 50.4 ms is lower than in related systems, including optical (69.2 ms in Fan and Xiao's depth-based approach [16], ~200 ms in Dante's depth camera [40], ~200 ms on a Hololens [61]) and body-worn sensors close to the finger (60–70 ms in TapID's wrist IMU [35]).

Sensing in motion. It is worth pointing out that throughout our evaluation, TapLight was moving around, which deliberately introduced considerable motion artifacts into our sensing evaluation. Although participants were sitting, they still moved and turned their heads during interaction. During the depth evaluation, the experimenters also produced substantial motion while recording.

Overall, the results of our study support our approach to touch-input sensing as a practical complement for the inside-out tracking on consumer VR headsets to detect input events during regular interaction and without requiring additional body-worn sensors.

5 APPLICATIONS

TapLight enables a wide variety of interactive VR scenarios that benefit from surface interaction. We prototyped six use cases to showcase TapLight capabilities, all running based on the input cues from TapLight's surface and touch detection as well as the Quest 2 headset tracking. Our apps span games, explorative, and productivity scenarios, all of which benefit from prolonged interaction that is supported by passive haptic feedback.

Touch down & up detection. Our demo apps support drag, pan, and zoom for bimanual interaction within the user's field of view. Our system determines the touching hand from the proximity to the detected surface and the head-gaze center. Release events are detected when the tip of the index finger, tracked by the VR headset, exits a band of 4 cm above the surface. This simple 2-state touch processing enables all our interaction modalities.

Demonstration apps. All apps afford situated interaction (Figure 8), either in front of an empty table or a wall in the real world, immersing the user and surrounding them with virtual content.

(a) Tappy Bird. Since TapLight detects quick touch events, it can power rapid input games, such as our take on the popular Flappy Bird game. Tappy Bird can be played on any surface in VR.

(b) Diagram editor. Users can select diagram elements from a menu that is attached to their left hand and diagrams of cells, connecting lines, and arrows on the surface. Users may also draw annotations. Diagram elements can be rearranged through touch & drag, as common in editors on traditional multitouch devices.

(c) Room design editor. Our room designer allows placing 3D elements from a menu into the room on the table in front of the user through touch. Elements can be rotated and dragged. Once a design has been created, users can render it in the space around them to experience the design in a situated setting to scale.

(d) VR web view. Our web view allows embedding any web page into virtual environments. The view can be navigated through the common touch controls to select links, scroll, zoom, etc.

(e) 2.5D city map. In this immersive geo renderer, we stream live data from Bing Maps and visualize the 2.5D building structure of a city. The map can be operated through touch controls, including panning and zooming using both hands.

(f) City planning. Finally, we designed a simple 2D map view that allows users to inspect locations and extract them for further annotation on the adjacent whiteboard. This demo works in situated settings that include two walls, such as the corner of a room, yet gives the impression of a much larger creative space.

6 LIMITATIONS

The use of a laser. TapLight was operated in closed environments by a single user only. The VR headset covered the user's eyes, which blocked laser reflections during motion. TapLight could substitute the visible laser with an infrared emitter, such as those embedded in structured light cameras (e.g., Kinect, Intel RealSense) and mobile phones (e.g., iPhone 12 Pro, Huawei P30, Google Pixel 4).

Operation range. TapLight's depth estimation, plane extraction, and touch detection reliably works within arm's length—the range where touch input is needed. While the signal diminishes for farther surfaces, TapLight's range of 1 m suffices for situated interaction.

Field of view & sparse depth. TapLight's current diffraction grating limits the spatial resolution of depth to be sparse. Since we require five reflections, TapLight can detect empty surfaces larger than 20 cm^2 (30 cm away) or 80 cm^2 (60 cm away). Our method rejects non-empty surfaces such as cluttered desks based on the plane's residual error, thereby preventing undesirable collisions.

Index fingers only. TapLight currently limits touch to the index fingers and cannot disambiguate individual fingers as done in our systems TapID or TapType, which used a wrist-worn sensor [35, 50].

Situated interaction. While TapLight reliably operates during motion, which is beneficial for integration into a headset, app users and study participants were seated, which limited their head motion. Using TapLight while standing or even walking may introduce additional motion. We guard against false positives by relating finger positions to detected surfaces but have not evaluated this.

7 IMPLICATIONS & FUTURE WORK

The main implication of our work is that structured light speckle is a viable method to complement depth sensing with minute motion and vibration sensing of objects in view—using vision as the one sensing modality and operating from a single vantage point. Thus, our work opens up the opportunity for rich action and interaction sensing on *one* integrated device—phenomena that may have so far required (additional) wearable inertial sensors or motion sensor-instrumented environments to resolve. While the focus in this paper has been on detecting hand-object interactions, in particular touch input on physical surfaces, our approach can generalize beyond this input modality to capturing a wider set of behaviors and events.

7.1 Rich camera-based touch & contact sensing

Structured light speckle shows a path forward that could allow vision-based touch detection to approach the interactive rates and detection accuracy common on surface-based touch sensors [22]. This includes recognizing touch in realistic configurations, i.e., reliably detecting events that are quick [16], subtle [33], and ambiguous [50]. Establishing this sensing capability using cameras will allow touch to become a recognized input modality as part of the interaction vocabulary in everyday surroundings—not just in VR [10, 53, 54] but also the wider Mixed Reality ecosystem.

Capturing the precise moment of touch contact will also improve spatial input accuracy. Since egocentric sensing platforms align well with users' mental models of precise input [30], the temporal certainty of input events can help already accurate sensing methods (e.g., 4.8 mm [60] corrected for systematic offsets, 9.8 mm [43] at 3 m) to further reduce errors towards the precision of touchscreens [29]—without the need for dwelling or signal aggregation and while supporting a fully movable sensing platform.

Building on our method's moment of contact detection, vision-based approaches also have the opportunity to better estimate touch *shape*. While inferring and interpreting shape has been an active area of research in touch sensing [1, 6, 49], this task has become popular in camera-based methods only recently due to advances in deep learning, which scale beyond fingers and hands [21] to human-level contact [17, 64]. We believe that sensing moments of physical contact and interaction will further aid these tasks.

Combined with previous work on speckle-based remote sensing [66, 70], our method also has implications for capturing touch *force* [38], especially when integrated into fluid interaction in Mixed Reality. Remote vibrometry could thereby offer a powerful input signal for learning-based methods that aim to detect touch and pressure from monocular cameras [20] and facilitate their operation on non-stationary sensing systems.

7.2 Transfer inertial methods to depth cameras

A final implication of our work is that structured light speckle can benefit the research on camera-based scene understanding, creating a bridge between the body of work on signal processing for inertial sensors and the computer vision community. We see an opportunity for future work to adapt processing methods for inertial sensors for the use in remote vibrometry to aid tasks such as human action recognition, robotics, and telemedicine. (See Chen et al.'s survey for an overview of possibilities [8].)

8 CONCLUSION

We have presented Structured Light Speckle, a method that fuses structured light-based depth estimation with speckle-based imaging to resolve remote vibrometry, such as during human-object interaction. Our prototype TapLight, an add-on system to current Mixed Reality headsets, implements this method specifically for discovering real-world surfaces that surround the user and for reliably detecting touch input on them. The use of structured light speckle enables TapLight to integrate both into a single device, forgoing the need for additional wearable sensors or instrumentation of the environment. TapLight complements camera-based hand tracking in current immersive systems and extends their interaction modes with reliable touch sensing, allowing surrounding physical affordances to be adopted in immersive interactive scenarios to provide passive haptic feedback upon input.

Our evaluation of TapLight showed its efficacy in detecting horizontal surfaces with an error of 22 mm in distance and 2.8° in normal error, sufficient for real-time integration during regular interaction. Our system registers interaction at speeds, intensities, and with a reliability that is common on touch devices, performing all tracking from a single apparatus, and robustly operating while in motion during regular wear.

ACKNOWLEDGMENTS

We sincerely thank Berken Utku Demirel and Andy Kong for helpful discussions and comments. We are further grateful to NVIDIA for the provision of computing resources through the NVIDIA Academic Grant. We thank the anonymous reviewers and all participants of our user studies.

REFERENCES

[1] Karan Ahuja, Paul Streli, and Christian Holz. 2021. TouchPose: Hand Pose Prediction, Depth Estimation, and Touch Classification from Capacitive Images. In *The 34th Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) *(UIST '21)*. Association for Computing Machinery, New York, NY, USA, 997–1009.

[2] Verena Biener, Snehanjali Kalamkar, Negar Nouri, Eyal Ofek, Michel Pahud, John J Dudley, Jinghui Hu, Per Ola Kristensson, Maheshya Weerasinghe, Klen Čopič Pucihar, et al. 2022. Quantifying the effects of working in VR for one week. *IEEE Transactions on Visualization and Computer Graphics* 28, 11 (2022), 3810–3820.

[3] Doug A Bowman, Sabine Coquillart, Bernd Froehlich, Michitaka Hirose, Yoshifumi Kitamura, Kiyoshi Kiyokawa, and Wolfgang Stuerzlinger. 2008. 3d user interfaces: New directions and perspectives. *IEEE computer graphics and applications* 28, 6 (2008), 20–36.

[4] Doug A. Bowman and Larry F. Hodges. 1997. An Evaluation of Techniques for Grabbing and Manipulating Remote Objects in Immersive Virtual Environments. In *Proceedings of the 1997 Symposium on Interactive 3D Graphics* (Providence, Rhode Island, USA) *(I3D '97)*. ACM, USA, 35–ff. https://doi.org/10.1145/253284.253301

[5] G. Bradski. 2000. The OpenCV Library. *Dr. Dobb's Journal of Software Tools* (2000).

[6] Xiang Cao, Andrew D Wilson, Ravin Balakrishnan, Ken Hinckley, and Scott E Hudson. 2008. ShapeTouch: Leveraging contact shape on interactive surfaces. In *2008 3rd IEEE International Workshop on Horizontal Interactive Human Computer Systems*. IEEE, 129–136.

[7] Jongeun Cha, Seung-man Kim, Ian Oakley, Jeha Ryu, and Kwan H. Lee. 2005. Haptic Interaction with Depth Video Media. In *Advances in Multimedia Information Processing - PCM 2005*, Yo-Sung Ho and Hyoung Joong Kim (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 420–430.

[8] Chen Chen, Roozbeh Jafari, and Nasser Kehtarnavaz. 2017. A survey of depth and inertial sensor fusion for human action recognition. *Multimedia Tools and Applications* 76 (2017), 4405–4425.

[9] Lung-Pan Cheng, Eyal Ofek, Christian Holz, Hrvoje Benko, and Andrew D Wilson. 2017. Sparse haptic proxy: Touch feedback in virtual environments using a general passive prop. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 3718–3728.

[10] Yi Fei Cheng, Christoph Gebhardt, and Christian Holz. 2023. InteractionAdapt: Interaction-driven Workspace Adaptation for Situated Virtual Reality Environments. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology* (San Francisco, CA, USA) *(UIST '23)*. Association for Computing Machinery, New York, NY, USA, Article 25. https://doi.org/10.1145/3586183.3606717

[11] Yi Fei Cheng, Tiffany Luong, Andreas Rene Fender, Paul Streli, and Christian Holz. 2022. ComforTable User Interfaces: Surfaces Reduce Input Error, Time, and Exertion for Tabletop and Mid-air User Interfaces. In *2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 150–159.

[12] HTC Corporation. 2023. *Vive, Discover Virtual Reality Beyond Imagination.* https://www.vive.com/us/

[13] Microsoft Corporation. 2019. *Microsoft HoloLens, Mixed Reality Technology for Business.* https://www.microsoft.com/en-us/hololens

[14] Xavier de Tinguy, Claudio Pacchierotti, Anatole Lécuyer, and Maud Marchal. 2020. Capacitive sensing for improving contact rendering with tangible objects in VR. *IEEE Transactions on Visualization and Computer Graphics* 27, 4 (2020), 2481–2487.

[15] Mustafa Doga Dogan, Steven Vidal Acevedo Colon, Varnika Sinha, Kaan Akşit, and Stefanie Mueller. 2021. SensiCut: Material-Aware Laser Cutting Using Speckle Sensing and Deep Learning. In *The 34th Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) *(UIST '21)*. Association for Computing Machinery, New York, NY, USA, 24–38. https://doi.org/10.1145/3472749.3474733

[16] Neil Xu Fan and Robert Xiao. 2022. Reducing the Latency of Touch Tracking on Ad-Hoc Surfaces. *Proc. ACM Hum.-Comput. Interact.* 6, ISS, Article 577 (nov 2022), 11 pages. https://doi.org/10.1145/3567730

[17] Mihai Fieraru, Mihai Zanfir, Elisabeta Oneata, Alin-Ionut Popa, Vlad Olaru, and Cristian Sminchisescu. 2020. Three-dimensional reconstruction of human interactions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7214–7223.

[18] Masaaki Fukumoto and Yoshinobu Tonomura. 1997. "Body coupled FingerRing" wireless wearable keyboard. In *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*. 147–154.

[19] Jun Gong, Aakar Gupta, and Hrvoje Benko. 2020. Acustico: Surface Tap Detection and Localization Using Wrist-Based Acoustic TDOA Sensing. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) *(UIST '20)*. Association for Computing Machinery, New York, NY, USA, 406–419. https://doi.org/10.1145/3379337.3415901

[20] Patrick Grady, Chengcheng Tang, Samarth Brahmbhatt, Christopher D Twigg, Chengde Wan, James Hays, and Charles C Kemp. 2022. Pressurevision: Estimating hand pressure from a single rgb image. In *European Conference on Computer Vision*. Springer, 328–345.

[21] Patrick Grady, Chengcheng Tang, Christopher D Twigg, Minh Vo, Samarth Brahmbhatt, and Charles C Kemp. 2021. Contactopt: Optimizing contact to improve grasps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1471–1481.

[22] Tobias Grosse-Puppendahl, Christian Holz, Gabe Cohn, Raphael Wimmer, Oskar Bechtold, Steve Hodges, Matthew S Reynolds, and Joshua R Smith. 2017. Finding common ground: A survey of capacitive sensing in human-computer interaction. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 3293–3315.

[23] Yizheng Gu, Chun Yu, Zhipeng Li, Weiqi Li, Shuchang Xu, Xiaoying Wei, and Yuanchun Shi. 2019. Accurate and Low-Latency Sensing of Touch Contact on Any Surface with Finger-Worn IMU Sensor. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans, LA, USA) *(UIST '19)*. Association for Computing Machinery, New York, NY, USA, 1059–1070. https://doi.org/10.1145/3332165.3347947

[24] Sean Gustafson, Christian Holz, and Patrick Baudisch. 2011. Imaginary Phone: Learning Imaginary Interfaces by Transferring Spatial Memory from a Familiar Device. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology* (Santa Barbara, California, USA) *(UIST '11)*. Association for Computing Machinery, New York, NY, USA, 283–292. https://doi.org/10.1145/2047196.2047233

[25] 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* (Santa Barbara, California, USA) *(UIST '11)*. Association for Computing Machinery, New York, NY, USA, 441–450. https://doi.org/10.1145/2047196.2047255

[26] Jeremy Hartmann, Christian Holz, Eyal Ofek, and Andrew D. Wilson. 2019. RealityCheck: Blending Virtual Environments with Situated Physical Reality. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. ACM, USA, 1–12. https://doi.org/10.1145/3290605.3300577

[27] Devamardeep Hayatpur, Seongkook Heo, Haijun Xia, Wolfgang Stuerzlinger, and Daniel Wigdor. 2019. Plane, Ray, and Point: Enabling Precise Spatial Manipulations with Shape Constraints. In *Proceedings of the 32nd Annual ACM Symposium*

on User Interface Software and Technology (New Orleans, LA, USA) *(UIST '19)*. ACM, USA, 1185–1195. https://doi.org/10.1145/3332165.3347916

[28] Steven Henderson and Steven Feiner. 2010. Opportunistic Tangible User Interfaces for Augmented Reality. *IEEE Transactions on Visualization and Computer Graphics* 16, 1 (2010), 4–16. https://doi.org/10.1109/TVCG.2009.91

[29] Christian Holz and Patrick Baudisch. 2010. The generalized perceived input point model and how to double touch accuracy by extracting fingerprints. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Atlanta, Georgia, USA) *(CHI '10)*. Association for Computing Machinery, New York, NY, USA, 581–590.

[30] Christian Holz and Patrick Baudisch. 2011. Understanding Touch. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vancouver, BC, Canada) *(CHI '11)*. ACM, USA, 2501–2510. https://doi.org/10.1145/1978942.1979308

[31] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, and Andrew Fitzgibbon. 2011. 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* (Santa Barbara, California, USA) *(UIST '11)*. Association for Computing Machinery, New York, NY, USA, 559–568. https://doi.org/10.1145/2047196.2047270

[32] Susan Jang, Jonathan M Vitale, Robert W Jyung, and John B Black. 2017. Direct manipulation is better than passive viewing for learning anatomy in a three-dimensional virtual reality environment. *Computers & Education* 106 (2017), 150–165.

[33] Wolf Kienzle, Eric Whitmire, Chris Rittaler, and Hrvoje Benko. 2021. Electroring: Subtle pinch and touch detection with a ring. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–12.

[34] Meta Technologies LLC. 2023. *Meta Quest 2: Our Most Advanced All-in-One VR Headset.* https://www.meta.com/quest-2/

[35] Manuel Meier, Paul Streli, Andreas Fender, and Christian Holz. 2021. TapID: Rapid Touch Interaction in Virtual Reality using Wearable Sensing. In *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*. IEEE, 519–528.

[36] Tim Menzner, Alexander Otte, Travis Gesslein, Jens Grubert, Philipp Gagel, and Daniel Schneider. 2019. A capacitive-sensing physical keyboard for vr text entry. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 1080–1081.

[37] Jiapu Pan and Willis J Tompkins. 1985. A real-time QRS detection algorithm. *IEEE transactions on biomedical engineering* 3 (1985), 230–236.

[38] Siyou Pei, Pradyumna Chari, Xue Wang, Xiaoying Yang, Achuta Kadambi, and Yang Zhang. 2022. ForceSight: Non-Contact Force Sensing with Laser Speckle Imaging. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology* (Bend, OR, USA) *(UIST '22)*. Association for Computing Machinery, New York, NY, USA, Article 25, 11 pages. https://doi.org/10.1145/3526113.3545622

[39] Hugo Romat, Andreas Fender, Manuel Meier, and Christian Holz. 2021. Flashpen: A High-Fidelity and High-Precision Multi-Surface Pen for Virtual Reality. In *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*. IEEE, 306–315.

[40] Elliot N. Saba, Eric C. Larson, and Shwetak N. Patel. 2012. Dante vision: In-air and touch gesture sensing for natural surface interaction with combined depth and thermal cameras. In *2012 IEEE International Conference on Emerging Signal Processing Applications*. 167–170. https://doi.org/10.1109/ESPA.2012.6152472

[41] Munehiko Sato, Shigeo Yoshida, Alex Olwal, Boxin Shi, Atsushi Hiyama, Tomohiro Tanikawa, Michitaka Hirose, and Ramesh Raskar. 2015. SpecTrans: Versatile Material Classification for Interaction with Textureless, Specular and Transparent Surfaces. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (Seoul, Republic of Korea) *(CHI '15)*. Association for Computing Machinery, New York, NY, USA, 2191–2200. https://doi.org/10.1145/2702123.2702169

[42] Maximilian Schrapel, Florian Herzog, Steffen Ryll, and Michael Rohs. 2020. Watch My Painting: The Back of the Hand as a Drawing Space for Smartwatches. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI EA '20)*. Association for Computing Machinery, New York, NY, USA, 1–10. https://doi.org/10.1145/3334480.3383040

[43] Vivian Shen, James Spann, and Chris Harrison. 2021. FarOut Touch: Extending the Range of Ad Hoc Touch Sensing with Depth Cameras. In *Proceedings of the 2021 ACM Symposium on Spatial User Interaction* (Virtual Event, USA) *(SUI '21)*. Association for Computing Machinery, New York, NY, USA, Article 5, 12 pages. https://doi.org/10.1145/3485279.3485281

[44] Yilei Shi, Haimo Zhang, Jiashuo Cao, and Suranga Nanayakkara. 2020. VersaTouch: A Versatile Plug-and-Play System That Enables Touch Interactions on Everyday Passive Surfaces. In *Proceedings of the Augmented Humans International Conference* (Kaiserslautern, Germany) *(AHs '20)*. Association for Computing Machinery, New York, NY, USA, Article 26, 12 pages. https://doi.org/10.1145/3384657.3384778

[45] Yilei Shi, Haimo Zhang, Kaixing Zhao, Jiashuo Cao, Mengmeng Sun, and Suranga Nanayakkara. 2020. Ready, Steady, Touch! Sensing Physical Contact with a Finger-Mounted IMU. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 4, 2, Article 59 (jun 2020), 25 pages. https://doi.org/10.1145/3397309

[46] Yi Chang Shih, Abe Davis, Samuel W. Hasinoff, Frédo Durand, and William T. Freeman. 2012. Laser speckle photography for surface tampering detection. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*. 33–40. https://doi.org/10.1109/CVPR.2012.6247655

[47] Adalberto L Simeone, Eduardo Velloso, and Hans Gellersen. 2015. Substitutional reality: Using the physical environment to design virtual reality experiences. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. 3307–3316.

[48] Paul Streli, Rayan Armani, Yi Fei Cheng, and Christian Holz. 2023. HOOV: Hand Out-Of-View Tracking for Proprioceptive Interaction Using Inertial Sensing. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI '23)*. Association for Computing Machinery, New York, NY, USA, Article 310, 16 pages. https://doi.org/10.1145/3544548.3581468

[49] Paul Streli and Christian Holz. 2021. CapContact: Super-resolution Contact Areas from Capacitive Touchscreens. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21, Article 289)*. Association for Computing Machinery, New York, NY, USA, 1–14.

[50] Paul Streli, Jiaxi Jiang, Andreas Rene Fender, Manuel Meier, Hugo Romat, and Christian Holz. 2022. TapType: Ten-finger text entry on everyday surfaces via Bayesian inference. In *CHI Conference on Human Factors in Computing Systems*. 1–16.

[51] Ryo Takahashi, Masaaki Fukumoto, Changyo Han, Takuya Sasatani, Yoshiaki Narusue, and Yoshihiro Kawahara. 2020. TelemetRing: A Batteryless and Wireless Ring-Shaped Keyboard Using Passive Inductive Telemetry. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) *(UIST '20)*. Association for Computing Machinery, New York, NY, USA, 1161–1168. https://doi.org/10.1145/3379337.3415873

[52] Rafael Veras, Gaganpreet Singh, Farzin Farhadi-Niaki, Ritesh Udhani, Parth Pradeep Patekar, Wei Zhou, Pourang Irani, and Wei Li. 2021. Elbow-Anchored Interaction: Designing Restful Mid-Air Input. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. ACM, USA, Article 737, 15 pages. https://doi.org/10.1145/3411764.3445546

[53] Raquel Viciana-Abad, Arcadio Reyes Lecuona, and Matthieu Poyade. 2010. The Influence of Passive Haptic Feedback and Difference Interaction Metaphors on Presence and Task Performance. *Presence* 19, 3 (2010), 197–212. https://doi.org/10.1162/pres.19.3.197

[54] Peng Wang, Xiaoliang Bai, Mark Billinghurst, Shusheng Zhang, Dechuan Han, Mengmeng Sun, Zhuo Wang, Hao Lv, and Shu Han. 2020. Haptic Feedback Helps Me? A VR-SAR Remote Collaborative System with Tangible Interaction. *International Journal of Human–Computer Interaction* 36, 13 (2020), 1242–1257. https://doi.org/10.1080/10447318.2020.1732140 arXiv:https://doi.org/10.1080/10447318.2020.1732140

[55] Diane Watson, Mark Hancock, Regan L. Mandryk, and Max Birk. 2013. Deconstructing the Touch Experience. In *Proceedings of the 2013 ACM International Conference on Interactive Tabletops and Surfaces* (St. Andrews, Scotland, United Kingdom) *(ITS '13)*. ACM, USA, 199–208. https://doi.org/10.1145/2512349.2512819

[56] Wikipedia, The Free Encyclopedia. 2023. Face ID. http://en.wikipedia.org/wiki/Face_ID [Online; accessed 05-April-2023].

[57] Andrew D. Wilson. 2010. Using a Depth Camera as a Touch Sensor. In *ACM International Conference on Interactive Tabletops and Surfaces* (Saarbrücken, Germany) *(ITS '10)*. Association for Computing Machinery, New York, NY, USA, 69–72. https://doi.org/10.1145/1936652.1936665

[58] Andrew D. Wilson and Hrvoje Benko. 2010. Combining Multiple Depth Cameras and Projectors for Interactions on, above and between Surfaces. In *Proceedings of ACM UIST 2010* (New York, New York, USA) *(UIST '10)*. ACM, USA, 273–282. https://doi.org/10.1145/1866029.1866073

[59] Chung-Ming Wu and Yung-Chang Chen. 1992. Statistical feature matrix for texture analysis. *CVGIP: Graphical Models and Image Processing* 54, 5 (1992), 407–419.

[60] Robert Xiao, Scott Hudson, and Chris Harrison. 2016. DIRECT: Making Touch Tracking on Ordinary Surfaces Practical with Hybrid Depth-Infrared Sensing. In *Proceedings of the 2016 ACM International Conference on Interactive Surfaces and Spaces*. 85–94.

[61] Robert Xiao, Julia Schwarz, Nick Throm, Andrew D. Wilson, and Hrvoje Benko. 2018. MRTouch: Adding Touch Input to Head-Mounted Mixed Reality. *IEEE Transactions on Visualization and Computer Graphics* 24, 4 (2018), 1653–1660. https://doi.org/10.1109/TVCG.2018.2794222

[62] Zihan Yan, Yuxiaotong Lin, Guanyun Wang, Yu Cai, Peng Cao, Haipeng Mi, and Yang Zhang. 2023. LaserShoes: Low-Cost Ground Surface Detection Using Laser Speckle Imaging. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI '23)*. Association for Computing Machinery, New York, NY, USA, Article 853, 20 pages. https://doi.org/10.1145/3544548.3581344

[63] Hui-Shyong Yeo, Juyoung Lee, Andrea Bianchi, Alejandro Samboy, Hideki Koike, Woontack Woo, and Aaron Quigley. 2020. WristLens: Enabling Single-Handed Surface Gesture Interaction for Wrist-Worn Devices Using Optical Motion Sensor. In *Proceedings of the Augmented Humans International Conference* (Kaiserslautern,

Germany) *(AHs '20)*. Association for Computing Machinery, New York, NY, USA, Article 27, 8 pages. https://doi.org/10.1145/3384657.3384797

[64] Yifei Yin, Chen Guo, Manuel Kaufmann, Juan Jose Zarate, Jie Song, and Otmar Hilliges. 2023. Hi4d: 4d instance segmentation of close human interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 17016–17027.

[65] Pietro Zanuttigh, Giulio Marin, Carlo Dal Mutto, Fabio Dominio, Ludovico Minto, Guido Maria Cortelazzo, et al. 2016. Time-of-flight and structured light depth cameras. *Technology and Applications* (2016), 978–3.

[66] Yang Zhang, Gierad Laput, and Chris Harrison. 2018. Vibrosight: Long-Range Vibrometry for Smart Environment Sensing. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology* (Berlin, Germany) *(UIST '18)*. Association for Computing Machinery, New York, NY, USA, 225–236. https://doi.org/10.1145/3242587.3242608

[67] Yang Zhang, Sven Mayer, Jesse T Gonzalez, and Chris Harrison. 2021. Vibrosight++: City-scale sensing using existing retroreflective signs and markers. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–14.

[68] Daniel Zielasko, Marcel Krüger, Benjamin Weyers, and Torsten W. Kuhlen. 2019. Menus on the Desk? System Control in DeskVR. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. 1287–1288. https://doi.org/10.1109/VR.2019.8797900

[69] Daniel Zielasko, Benjamin Weyers, Martin Bellgardt, Sebastian Pick, Alexander Meibner, Tom Vierjahn, and Torsten W. Kuhlen. 2017. Remain seated: towards fully-immersive desktop VR. In *2017 IEEE 3rd Workshop on Everyday Virtual Reality (WEVR)*. 1–6. https://doi.org/10.1109/WEVR.2017.7957707

[70] Jan Zizka, Alex Olwal, and Ramesh Raskar. 2011. SpeckleSense: Fast, Precise, Low-Cost and Compact Motion Sensing Using Laser Speckle. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology* (Santa Barbara, California, USA) *(UIST '11)*. Association for Computing Machinery, New York, NY, USA, 489–498. https://doi.org/10.1145/2047196.2047261