Real-Time Multimodal Fingertip Contact Detection via Depth and Motion Fusion for Vision-Based Human-Computer Interaction

Supplementary Material

1. Overview

This supplementary material provides technical details and additional analyses that complement the main paper. We organize the content as follows:

- **Section 2**: Extended Related Work
- **Section 3**: Dataset Details
- **Section 4**:Camera Calibration
- **Section 5**: Methodology Details
- **Section 6**: Training and Optimization
- **Section 7**: Deployment Configurations
- **Section 8**: Additional Experimental Results
- **Section 9**: User Study Details
- **Section 10**: Extended Discussion

2. Extended Related Work

2.1. Touch Detection and Surface Interaction

The detection of fingertip contact on physical surfaces has been a longstanding research objective in human-computer interaction (HCI). Wilson [13] introduced seminal work on depth-based touch sensing, employing a depth camera as a proximity sensor to detect surface contact. This foundational approach established depth thresholding as a mechanism for disambiguating intentional touches from hovering gestures, informing subsequent research on vision-based touch detection. Building upon this foundation, Harrison et al. [3] proposed OmniTouch, a wearable projection-camera system enabling multi-touch interaction on arbitrary surfaces including walls, tables, and the user's own body. Concurrently, Xiao et al. [14] introduced WorldKit, which allows users to dynamically create interactive surfaces by projecting interfaces onto everyday objects. These pioneering systems demonstrated the viability of transforming passive physical surfaces into interactive input modalities, though they relied on dedicated depth-sensing hardware.

Recent advances have extended touch detection to head-mounted display (HMD) environments and egocentric viewpoints. Xiao et al. [15] proposed MRTouch, which enables touch input on arbitrary surfaces in mixed reality by leveraging the depth sensor integrated within the HMD to detect fingertip contact on planar surfaces. Mollyn and Harrison [8] introduced EgoTouch, detecting on-body touch interactions using egocentric cameras mounted on AR/VR headsets, addressing inherent self-occlusion challenges from first-person viewpoints. Streli et al. [11] developed TouchInsight, an uncertainty-aware system for rapid touch and text input in mixed reality from egocentric vision, while Richardson et al. [9] proposed StegoType for surface typing detection from egocentric cameras. These methods demonstrate the feasibility of touch detection from head-mounted perspectives, though they typically require specialized depth sensors or operate at spatial resolutions insufficient for precise keystroke detection.

Table 1. **Positioning of Our Approach in the Touch Detection Landscape.**

Category	Representative Work	Key Limitation	Our Advantage
Depth-camera based	Wilson [13], TapType [18]	Dedicated depth sensor required	Single RGB camera
Egocentric (HMD)	EgoTouch [8], TouchInsight [11]	Coarser resolution, HMD-only	External or HMD cameras
Pressure estimation	PressureVision++ [2]	Force magnitude, not binary	Real-time binary contact
Commercial VR	Quest 3, Apple Vision Pro	Proprietary, closed ecosystem	Open dataset, reproducible
Motion capture	Vicon, OptiTrack	Laboratory-only, non-portable	Single commodity camera

Complementary research has addressed the precision requirements and force estimation aspects of touch interaction. Holz and Baudisch [4] conducted a foundational study on touch accuracy, identifying the "fat finger" problem wherein the intended contact point diverges from the detected contact area due to finger deformation upon surface contact. Their findings establish fundamental precision requirements—typically within 2–3 mm—that inform threshold-based contact detection approaches. Grady et al. [2] extended touch sensing beyond binary detection with PressureVision++, which estimates continuous fingertip pressure from RGB images. While their approach captures force magnitude for pressure-sensitive applications, our method focuses on precise binary contact detection through depth and velocity fusion, prioritizing real-time contact event detection for latency-sensitive applications such as virtual keyboard input. Unlike prior depth-camera-based approaches [13, 15] that require dedicated depth hardware, and unlike recent egocentric methods [8, 11] that operate at coarser spatial resolutions, our approach achieves millimeter-level contact detection from a single RGB camera. This capability is enabled by domain-specific fine-tuning of a monocular depth estimation model on our custom hand-surface interaction dataset, which closes the domain gap between generic depth predictions trained on room-scale scenes and the precision required for reliable touch sensing on physical surfaces. Furthermore, our fusion of depth estimation with motion-based velocity features eliminates dependency on built-in depth sensors, enabling robust contact detection from standard RGB cameras at varying viewing angles—from desk-mounted configurations to HMD-mounted egocentric perspectives (Table 1).

3. Dataset Details

3.1. Data Collection Apparatus

We designed a data collection apparatus to emulate a typical desktop interaction environment while enabling multi-view capture for viewpoint-robust model training. The primary sensing device is an Intel RealSense D405 depth camera, selected for its high accuracy at close ranges (optimal operating range 7–50 cm, specified depth error $<$0.5 mm at 30 cm). The D405 was mounted on a stable tripod positioned 35 cm above a standard white desk surface at a 45° viewing angle, providing an unobstructed view of the typing region.

The 35 cm mounting height was selected to match the D405's optimal operating range and to approximate the camera-to-hand distance typical of HMD-mounted front-facing cameras when the user's hands rest on a desk surface. While we did not systematically vary distance, natural hand movement during typing introduces ±3–5 cm of depth variation within each session, providing some implicit distance diversity in the training data.

To support multi-angle training and evaluation (Section 8.2), two additional RGB cameras were positioned at 30° and 60° viewing angles relative to the typing surface. The 30° configuration simulates a front-facing HMD-mounted camera, while the 60° configuration represents a shallower angle. Each sensor underwent independent intrinsic calibration: the RealSense D405 via the manufacturer's Depth Quality Tool, and the auxiliary RGB cameras via Zhang's method [17] using a printed chessboard pattern in OpenCV. A subsequent joint calibration step recovered extrinsic transformations between all sensors through a custom pipeline that combines per-camera parameters into a unified coordinate frame (main paper Figure 2). During data recording, the D405 captures synchronized RGB and depth streams, while the auxiliary cameras capture RGB data only. Our custom synchronization pipeline temporally aligns each depth frame from the D405 with the corresponding RGB frames from each auxiliary camera using hardware timestamps. This configuration enables acquisition of paired RGB-depth data for all three viewing angles using a single depth sensor—a critical requirement for training monocular depth estimation models, which necessitate synchronized RGB-depth pairs as supervision.

To evaluate material robustness, data were collected on three surface types: a standard white desk (primary), natural wood grain, and semi-reflective laminate. The environment was illuminated with consistent, diffuse overhead LED lighting (5000K, 800 lux at desk surface) to minimize shadows and specular highlights that could degrade depth estimation performance.

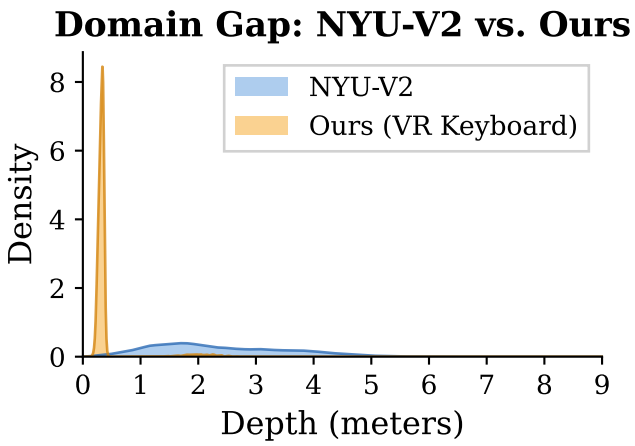

Figure 1. **Domain gap between NYU Depth V2 and our dataset.** Smoothed depth value distributions reveal a fundamental mismatch: our dataset is tightly concentrated in the 0.2–0.4 m range corresponding to hand-surface interaction distances. In contrast, NYU Depth V2 spans a broad 0.5–5 m range reflecting typical indoor room geometry. The resulting Jensen-Shannon Divergence of 0.576 confirms the two domains are nearly disjoint, motivating task-specific fine-tuning.

3.2. Data Collection Protocol

We recruited 15 participants exhibiting diverse hand morphology and skin pigmentation to ensure broad dataset generalizability. Palm widths ranged from 72–95 mm, encompassing substantial variation in hand geometry, while Fitzpatrick skin tone classifications spanned Types II–V [10], representing a diverse range of pigmentation levels. Each participant engaged in multiple 15-minute recording sessions, performing natural typing motions on a physical keyboard placed within the capture volume. To maximize behavioral variability and ensure comprehensive coverage of realistic interaction patterns, participants were instructed to:

- Vary typing speeds from deliberate single-finger "hunt-and-peck" input to rapid touch typing;
- Alternate between single-hand and bimanual typing patterns;
- Incorporate explicit hovering states and intentional contact events;
- Adopt varied hand postures and wrist orientations throughout each session.

Synchronized RGB-depth pairs were captured at 60 FPS with spatial resolution of 640×480 pixels for both modalities.

3.3. Domain Gap Analysis

The final dataset provides high-precision, metric depth information, with depth maps stored as 16-bit PNGs where each pixel value corresponds to a depth of 1 mm (integer

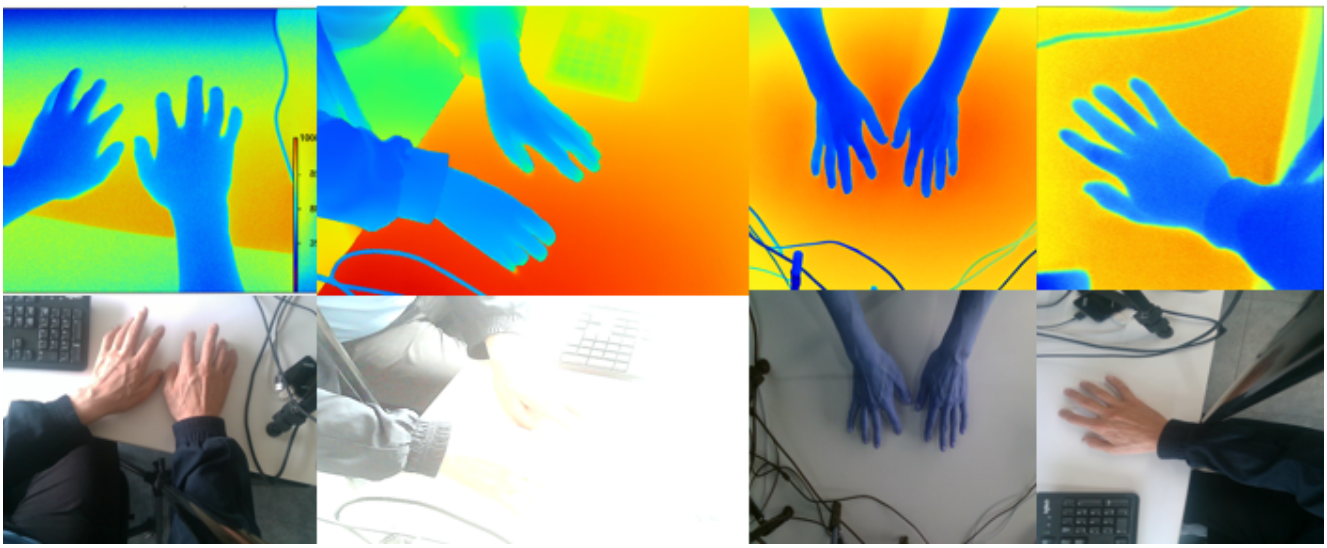

Figure 2. **Dataset Visualization and Examples.** Top: Depth data captured from multiple viewpoints and angles. Bottom: Related RGB data recorded under diverse camera perspectives and viewing conditions.

millimeters).

To quantitatively validate the existence of the domain gap, we computed the Jensen-Shannon Divergence (JSD) [5] between the depth value distributions of our dataset and the NYU Depth V2 dataset. As shown in Figure 1, the distributions are starkly different. Our dataset's depth values are tightly concentrated in the 0.2-0.4 meter range, whereas NYU Depth V2 spans a much broader 0.5-5 meter range. The resulting JSD of 0.576 is substantial, approaching the divergence seen between entirely different domains and confirming that standard datasets are fundamentally mismatched for our task. This analysis underscores the necessity of our targeted data collection effort for achieving high-precision, close-range depth estimation, motivating specialized fine-tuning for depth-based hand fingertip interaction modeling (Figure 1).

This domain gap is not merely a statistical artifact but has direct practical consequences: unsupervised domain adaptation methods that attempt to bridge this gap without target-domain labels plateau at 8.7–9.2 mm MAE—insufficient for the 3–8 mm discrimination required for contact detection. Only supervised fine-tuning with our specialized dataset achieves the necessary precision.

3.4. Dataset Visualization

We provide illustrative samples and statistics from our dataset. Figure 2 shows a 2×4 grid of paired RGB and depth examples, highlighting diverse viewpoints and camera angles. The top row presents depth maps visualized with false-color gradients, while the bottom row shows corresponding RGB frames. In addition to qualitative examples, we report dataset statistics: the distribution of contact vs. hover

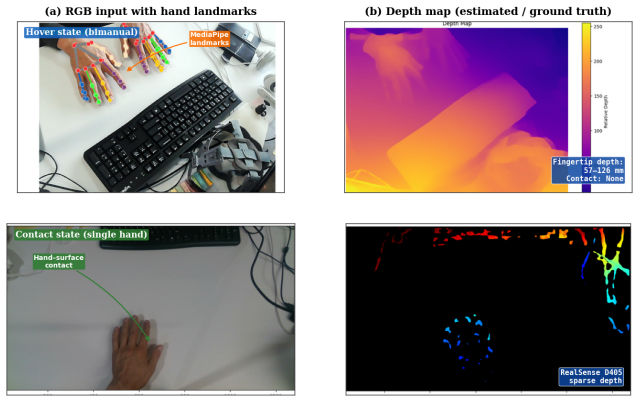

Figure 3. Representative samples from our dataset (53,300 RGB depth pairs, 640×480, Intel RealSense D405 ground truth)

frames across sessions, and per-participant breakdowns of frame counts. These examples and statistics demonstrate both the multimodal richness of the dataset and its suitability for analyzing fine-grained hand interactions.

Figure 3 presents representative samples from our dataset illustrating the two primary interaction states. Top row: bimanual hover state with MediaPipe landmarks overlaid (a) and the corresponding dense depth map (b), where fingertip depths range 57–126 mm above the surface. Bottom row: single-hand contact state showing hand-surface contact (a) and the RealSense D405 sparse depth capture (b), where the contacting fingertip registers $\leq$3 mm from the surface. The contrast between hover and contact depth signatures motivates the millimeter-level precision required for reliable touch discrimination.

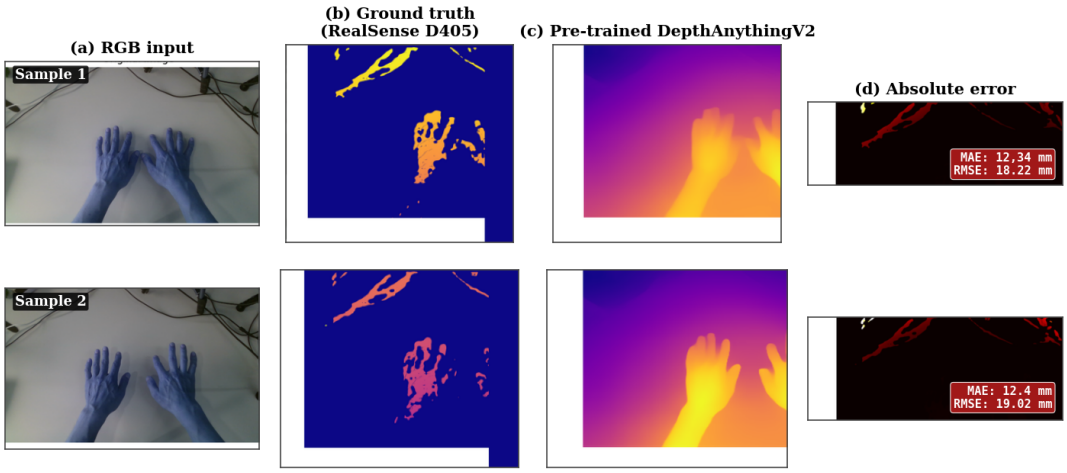

Figure 4. Qualitative depth estimation comparison on two representative samples. (a) RGB input, (b) RealSense D405 ground truth, (c) pre-trained DepthAnythingV2 prediction (median-scaled), (d) absolute error map. Per-sample MAE of 12.3–12.4 mm is consistent with the dataset-level 12.3 mm reported in Table 1, main paper. After fine-tuning, MAE drops to 3.84 mm (68% reduction).

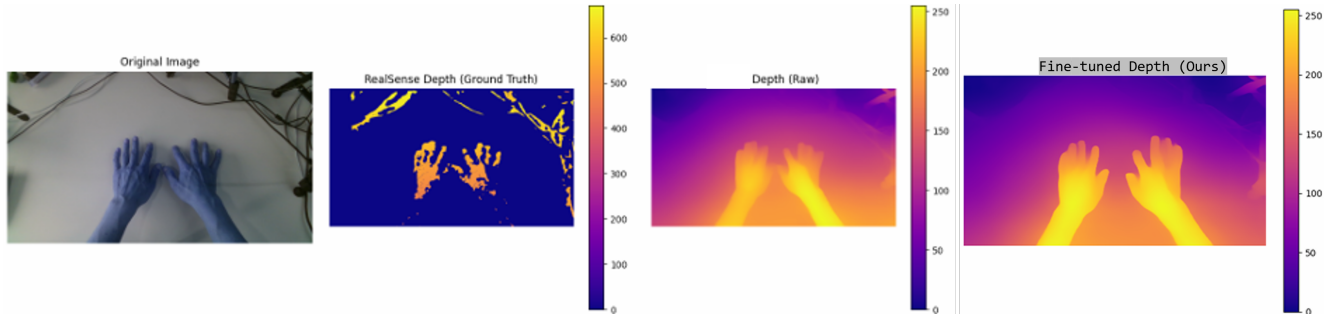

Figure 5. **Qualitative depth estimation comparison.** From left to right: RGB input showing hands during typing, RealSense D405 ground truth, pre-trained depth prediction exhibiting severe domain gap (MAE = 12.3 mm), and our fine-tuned model output with clear finger segmentation (MAE = 3.84 mm). The pre-trained model fails to resolve individual fingers and produces incorrect depth scaling. In contrast, our fine-tuned model achieves millimeter-level accuracy with sharp finger boundaries — critical for reliable contact detection.

Figure 4 visualizes the domain gap on two representative frames. Even after standard median scale alignment, pre-trained DepthAnythingV2 produces MAE of 12.3–12.4 mm with error concentrated on fingertip and hand-edge regions (column d)—well above the 4.5 mm contact threshold required for reliable keystroke detection. The depth maps (column c) exhibit smooth gradients across the hand without resolving individual finger boundaries, rendering per-finger contact discrimination infeasible. Fine-tuning on our dataset reduces MAE to 3.84 mm with sharp finger segmentation, compressing the error distribution below the contact threshold.

Figure 5 shows the full severity of the domain gap before any scale alignment. The pre-trained DPT prediction (third column) produces a smooth gradient across the entire hand region with a compressed depth range (0–250 mm colorbar), failing to resolve individual fingers or distinguish the hand from the desk surface. The resulting raw MAE of 12.3 mm reflects both a global scale mismatch and the absence of fine-grained structural detail. After fine-tuning on our dataset (fourth column), the model recovers sharp finger boundaries with a depth range matching the ground truth, reducing MAE to 3.84 mm. Note that this figure reports raw error without median scale alignment to illustrate the complete domain gap; Table 1 (main paper) reports dataset-level metrics with standard alignment for fair cross-model comparison.

To provide a detailed view of hand dynamics, we include 3D scatter plots of fingertip positions recorded during typing sessions (Figure 6). These visualizations highlight the trajectories of individual fingers across time, with distinct colors assigned to each hand and contact state. Left-hand fingers are rendered in red, right-hand fingers in blue, and fingertip states are further distinguished by green (contact)

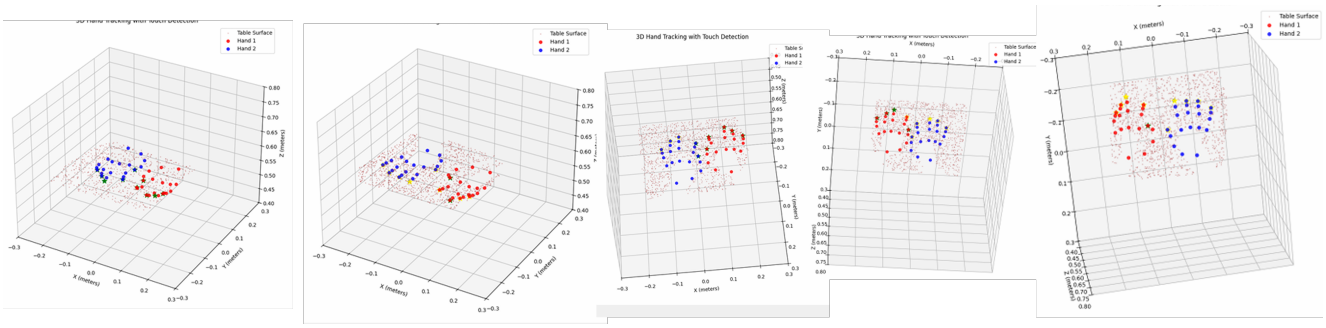

Figure 6. **3D Visualization of Fingertip Trajectories.** Fingertip positions and trajectories are visualized in 3D space during typing interactions. Each finger is color-coded for clarity: left-hand fingers are shown in red, right-hand fingers in blue. Fingertips in contact with the keyboard are highlighted in green, while hovering (non-contact or near-contact) fingertips are displayed in yellow.

and yellow (hover). This representation emphasizes the spatial distribution of finger movements and the transitions between hovering and pressing states. While too detailed for the main paper, these plots serve as an informative supplementary figure, offering a deeper understanding of fine-grained hand interactions and supporting the robustness of our fingertip detection pipeline.

3.5. Dataset Statistics

Figure 7 summarizes the dataset composition. The grouped bar chart (a) shows a near-balanced hover/contact ratio across all three partitions (Train: 20,766 / 21,874; Val: 2,596 / 2,734; Test: 2,596 / 2,734; ratio ≈0.95:1), mitigating classification bias. The pie chart (b) confirms the 8:1:1 split (42,640 / 5,330 / 5,330), stratified by participant ID to prevent identity leakage across partitions.

The right panel illustrates the overall partitioning scheme, which follows an 8:1:1 ratio for training, validation, and test sets, respectively. Importantly, the split is stratified by participant ID rather than by individual frames, ensuring that all samples from a given participant appear exclusively in one partition. This design choice prevents identity-specific features—such as hand geometry, skin tone, or typing style—from leaking across sets, thereby enabling rigorous evaluation of the model's ability to generalize to previously unseen users. The resulting dataset comprises 42,640 training, 5,330 validation, and 5,330 test samples, totaling 53,300 synchronized RGB-depth pairs.

Figure 8 details the demographic diversity and individual contributions of our 15 participants. The top panel displays per-participant frame counts, with each bar color-coded according to Fitzpatrick skin tone classification (Types II–V). The distribution demonstrates balanced representation across skin tones, ensuring that the trained model generalizes across diverse pigmentation levels without bias. Individual contributions range from approximately 2,800 to 4,200 frames, collectively summing to 53,300 synchronized RGB-depth pairs.

The bottom panel presents a dual-axis lollipop chart capturing anthropometric and behavioral variation across the participant pool. Palm widths span 72–95 mm, reflecting substantial diversity in hand geometry that challenges the model to accommodate varying finger lengths, joint proportions, and spatial coverage patterns. The contact ratio—defined as the proportion of contact frames relative to total frames per participant—varies between approximately 48% and 55%, indicating consistent but naturally varying typing behaviors. This controlled variability ensures that the dataset captures realistic interaction dynamics while maintaining sufficient balance between hovering and contact states across all participants. Together, these statistics validate that our dataset encompasses the morphological and behavioral diversity necessary for training a robust, generalizable contact detection system.

4. Camera Calibration

Accurate 3D reconstruction of fingertip positions requires precise camera calibration to convert 2D image coordinates to metric 3D world coordinates. We calibrate the Intel RealSense D405 depth camera using a standard checkerboard pattern (7×4 inner corners, 25mm square size) following Zhang's method [17].

4.1. Intrinsic Parameters

The camera intrinsic matrix $\mathbf{K}$ relates 3D camera coordinates to 2D image coordinates:

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{K} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}, \quad \mathbf{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

where $s = Z_c$ is the scale factor, (f_x, f_y) are focal lengths in pixels, and (c_x, c_y) is the principal point (optical center).

RGB Sensor (640×480)**:**

$$\mathbf{K}_{\text{RGB}} = \begin{bmatrix} 607.31 & 0 & 321.45 \\ 0 & 607.89 & 242.18 \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

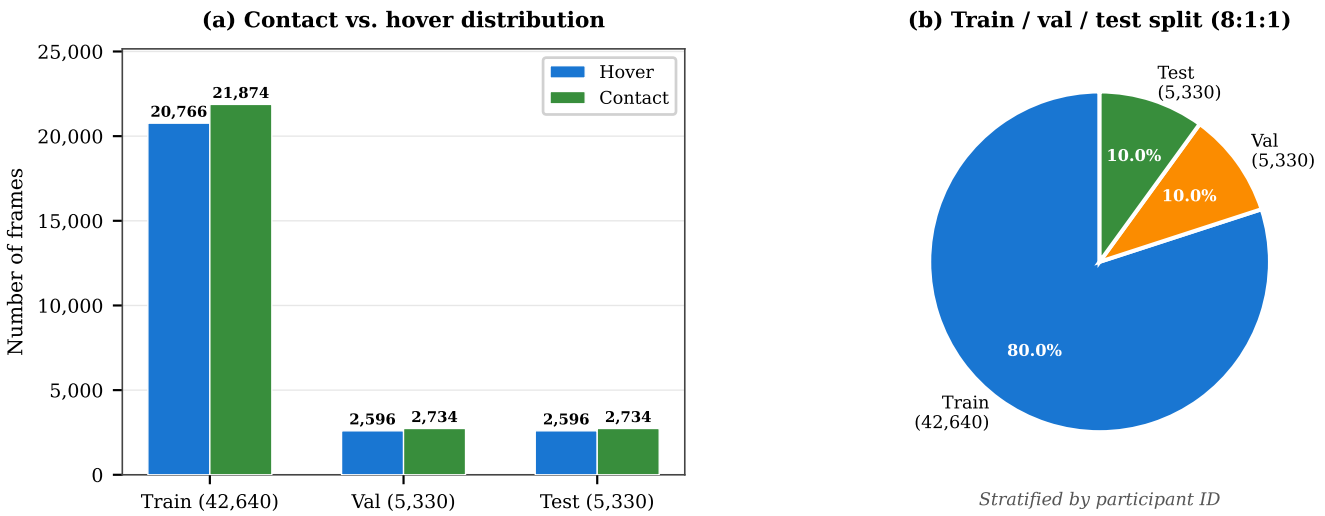

Figure 7. **Dataset Statistics and Distribution.** Left: Grouped bar chart illustrating the near-balanced distribution of hovering and contact samples across all three partitions (Train: 20,766 hover / 21,874 contact; Val: 2,596 hover / 2,734 contact; Test: 2,596 hover / 2,734 contact). Right: Pie chart depicting the 8:1:1 train/validation/test split (42,640 / 5,330 / 5,330), stratified by participant ID to prevent data leakage across partitions.

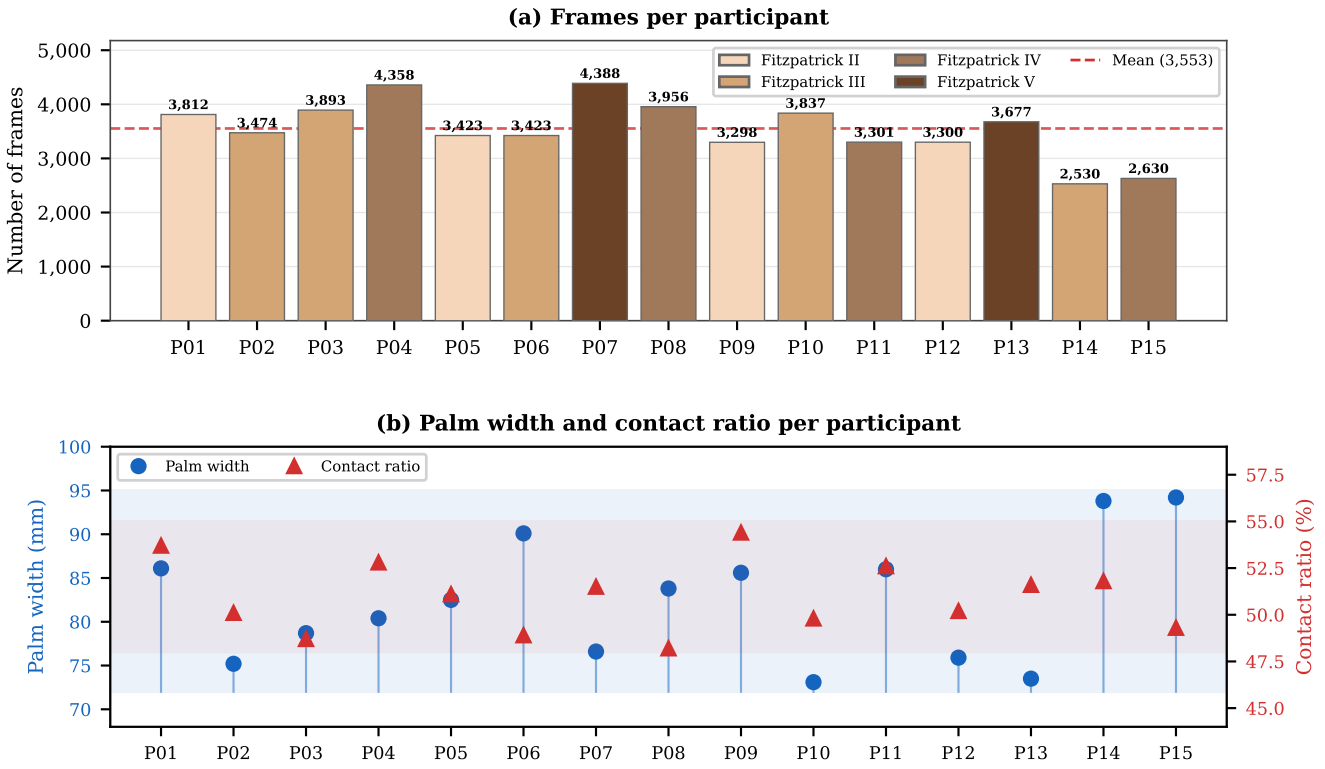

Figure 8. **Participant Demographics and Contribution.** Top: Per-participant frame counts (P01–P15) color-coded by Fitzpatrick skin tone (II–V). Bottom: Dual-axis lollipop chart showing palm width (72–95 mm) and contact ratio (48–55%) variation across participants.

Depth Sensor (640×480)**:**

$$\mathbf{K}_{\text{depth}} = \begin{bmatrix} 383.72 & 0 & 319.86 \\ 0 & 383.72 & 238.94 \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

4.2. Lens Distortion Model

Lens distortion is modeled using the Brown-Conrady model [1]. First, convert pixel coordinates to normalized coordi-

nates:

$$(x, y) = \left(\frac{u - c_x}{f_x}, \frac{v - c_y}{f_y} \right) \tag{4}$$

Then apply radial and tangential distortion with $r^2 = x^2 + y^2$:

$$x' = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + [2p_1 xy + p_2(r^2 + 2x^2)] \tag{5}$$

$$y' = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + [p_1(r^2 + 2y^2) + 2p_2 xy] \tag{6}$$

where (k_1, k_2, k_3) are radial distortion coefficients and (p_1, p_2) are tangential distortion coefficients.

RGB Sensor Distortion Coefficients:

$$[k_1, k_2, k_3, p_1, p_2] = [0.1537, -0.4829, 0.0021, -0.0008, 0.3912] \tag{7}$$

The positive k_1 indicates barrel distortion, negative k_2 compensates overcorrection at image edges, and near-zero p_1, p_2 indicate well-aligned optical components.

4.3. Extrinsic Transformation

RGB and depth sensors are physically separated by ≈50mm. The rigid transformation from depth frame to RGB frame is:

$$\begin{bmatrix} X_{\text{RGB}} \\ Y_{\text{RGB}} \\ Z_{\text{RGB}} \end{bmatrix} = \mathbf{R}_{\text{depth}\to\text{RGB}} \begin{bmatrix} X_{\text{depth}} \\ Y_{\text{depth}} \\ Z_{\text{depth}} \end{bmatrix} + \mathbf{t}_{\text{depth}\to\text{RGB}} \tag{8}$$

Rotation Matrix:

$$\mathbf{R}_{\text{depth}\to\text{RGB}} = \begin{bmatrix} 0.999987 & -0.003841 & 0.003216 \\ 0.003849 & 0.999991 & -0.001825 \\ -0.003206 & 0.001839 & 0.999993 \end{bmatrix} \tag{9}$$

Translation Vector:

$$\mathbf{t}_{\text{depth}\to\text{RGB}} = \begin{bmatrix} -50.12 \\ 0.38 \\ -0.21 \end{bmatrix} \text{mm} \tag{10}$$

Calibration Accuracy: Mean reprojection error: RGB 0.31 pixels, depth 0.28 pixels (≈0.5mm at 350mm distance). 3D reconstruction error: 0.52mm, well below our 5mm contact detection threshold.

5. Methodology Details

5.1. Depth Estimation Metrics

Monocular depth models are evaluated using complementary metrics [17] capturing different prediction quality aspects. We present improved formulations with explicit handling of edge cases and clear variable definitions.

Mean Absolute Error (MAE) measures the average absolute difference between predicted and ground truth depth values [6] Equ. 11:

$$MAE = \frac{1}{N} \sum_{i=1}^{N} \left| d_{pred}^i - d_{gt}^i \right| \tag{11}$$

where d_{pred}^i is the predicted depth at pixel i, d_{gt}^i is the ground truth depth at pixel i, N is the number of valid pixels, millimeters (mm) units were used in this work.

Absolute Relative Error (Abs Rel) measures the mean relative deviation Equ. 12:

$$AbsRel = \frac{1}{|\mathcal{V}|} \sum_{i \in V} \frac{|d_{pred}^i - d_{gt}^i|}{d_{gt}^i} \tag{12}$$

Avoiding division by zero when $d_{gt}^i = 0$ in original form, we introduced valid pixel set $\mathcal{V} = \{i : d_{gt}^i > \epsilon_{min}\}$ where ϵ_{min} mm (sensor minimum detectable distance) providing scale-normalized error assessment. It provides scale-normalized error assessment, making it comparable across different depth ranges.

Squared Relative Error (Sq Rel) penalizes larger errors more heavily making it sensitive to outliers emphasizing large errors more than MAE due to squaring Equ. 13:

$$SqRel = \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} \frac{(d_{pred}^i - d_{gt}^i)^2}{(d_{gt}^i)^2} \tag{13}$$

where $\mathcal{V} = \{i : d_{gt}^i > \epsilon_{min}\}$.

Root Mean Square Error (RMSE) quantifies absolute prediction error with units matching the depth measurements Equ. 14:

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (d_{pred}^i - d_{gt}^i)^2} \tag{14}$$

Accuracy thresholds Equ. 15:

$$\delta_t = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1} \left[max \left(\frac{d_{pred}^i}{d_{gt}^i}, \frac{d_{gt}^i}{d_{pred}^i} \right) < 1.25^t \right] \tag{15}$$

where $t \in 1, 2, 3$ and $\mathbf{1}[\cdot]$ is the indicator function. This reports the percentage of predictions within multiplicative factors 1.25 (δ_1), $1.25^2 = 1.5625(\delta_2)$, and $1.25^3 = 1.953(\delta_3)$) report the percentage of predictions satisfying $\max(\frac{d_{\text{pred}}}{d_{\text{gt}}}, \frac{d_{\text{gt}}}{d_{\text{pred}}}) < \delta$, capturing the proportion of predictions within multiplicative factors of ground truth. Our fine-tuned model achieves, meaning 95.96% of predictions are within ±25% of ground truth.

Scale-Invariant Logarithmic Error (SILog) measures relative error while being robust to global scale variations

Equ. 16:

$$SILog = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\epsilon_i^2 - \frac{1}{N^2}\left(\sum_{i=1}^{N}(\epsilon_i)\right)^2} \quad (16)$$

where $\epsilon_i = \ln d^i_{pred} - \ln d^i_{gt}$. To prevent $log(o)$ errors, we use: $\epsilon_i = ln\left(max\left(d^i_{pred}, \epsilon_{min}\right)\right) - ln\left(max\left(d^i_{gt}, \epsilon_{min}\right)\right)$ where $\epsilon_{min} = 0.1$ mm (sensor minimum detectable distance). First term, is a variance of pixel-wise logarithmic errors (captures local depth inaccuracies), while second term is a squared mean of logarithmic errors (captures global scale bias, subtracted to achieve scale invariance) When distinguishing hovering (8 mm) from contact (3 mm), the relative error (5 mm / 8 mm = 62.5%) is more informative than absolute error (5 mm). SILog emphasizes such relative relationships while being invariant to uniform depth scaling—critical for monocular estimation where absolute scale is ambiguous.

R^2 measures the proportion of variance in ground truth depth explained by predictions Equ. 17:

$$R^2 = 1 - \frac{\sum_{i=1}^{N}\left(d^i_{gt} - d^i_{pred}\right)^2}{\sum_{i=1}^{N}\left(d^i_{gt} - \bar{d}_{gt}\right)^2} \quad (17)$$

where $\bar{d}^{gt} = \frac{1}{N}\sum_{i=1}^{N} d^i_{gt}$ is the mean ground truth depth. R^2=1: perfect prediction, R^2=0.989 (our result): model explains 98.9% of depth variance, R^2=0: no better than baseline, ad $R^2 < 0$: worse than mean baseline.

A standard L1 or L2 loss on depth values is suboptimal for our task, as it penalizes absolute errors equally across all distances. For contact detection, the Scale-Invariant Logarithmic (SILog) loss was employed, which penalizes relative depth errors—critical for discriminating contact (3 mm) from hovering (8 mm) at close range (Equ. 18):

$$\mathcal{L}_{\text{SILog}} = \frac{1}{n}\sum_{i=1}^{n}(\log d_i - \log d_i^*)^2 - \frac{\alpha}{n^2}\left(\sum_{i=1}^{n}(\log d_i - \log d_i^*)\right)^2 \quad (18)$$

When trained on our domain-specific dataset, SILog implicitly emphasizes the depth range where contact discrimination occurs, as the close-range distribution concentrates gradients in the 0–15 mm fingertip-to-surface zone.

5.2. Pipeline Architecture

Our system processes each RGB frame through a four-stage pipeline. First, MediaPipe extracts 2D hand landmarks to localize fingertip positions in the image plane. These regions are then passed to a fine-tuned Depth model, which produces a dense metric depth map. From this depth map, we fit a reference plane to the keyboard surface via RANSAC and compute the signed perpendicular distance from each fingertip to the plane. Finally, a velocity-gated hysteresis state machine classifies each fingertip as *hovering* or *in contact* using dual thresholds ($\tau_{\text{contact}} = 4.5$ mm for engagement, $\tau_{\text{exit}} = 6.0$ mm for release), suppressing spurious transitions caused by depth noise. The full pipeline runs at ∼30 FPS with 93 ms end-to-end latency at 640×480 resolution (Figure 9).

5.3. Depth Integration Pipeline

For each fingertip landmark (u, v) detected by MediaPipe, we compute its absolute distance to the table surface:

1. Extract depth value d from predicted depth map at pixel coordinates (u, v)
2. Transform 2D coordinates to 3D world coordinates:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = d \cdot \mathbf{K}^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \quad (19)$$

3. Calculate signed perpendicular distance to table plane:

$$D_{\text{contact}} = \frac{|aX + bY + cZ + d|}{\sqrt{a^2 + b^2 + c^2}} \quad (20)$$

where (a, b, c, d) are the RANSAC-fitted table plane parameters

5.4. Threshold Hysteresis Mechanism

We implemented a threshold hysteresis mechanism to stabilize contact detection and prevent rapid toggling during borderline cases.

Dual-Threshold Design Instead of a single threshold (5mm), we use dual thresholds:

- **Contact Entry Threshold:** 4.5mm (fingertip must be closer than this to trigger contact)
- **Contact Exit Threshold:** 6.0mm (fingertip must exceed this distance to release contact)

The state transition logic:

$$\text{State}(t) = \begin{cases} \text{Contact} & \text{if } d(t) < \tau_{\text{in}} \\ \text{Hover} & \text{if } d(t) > \tau_{\text{out}} \\ \text{State}(t-1) & \text{otherwise} \end{cases} \quad (21)$$

where $\tau_{\text{in}} = 4.5$mm and $\tau_{\text{out}} = 6.0$mm.

Table 2. Impact of threshold hysteresis on contact detection stability.

Configuration	False Pos. Reduc.	Precision	Jitter Err.
Single threshold	Baseline	91.2%	47 per sess.
Dual threshold	28%	93.7%	32 per sess.

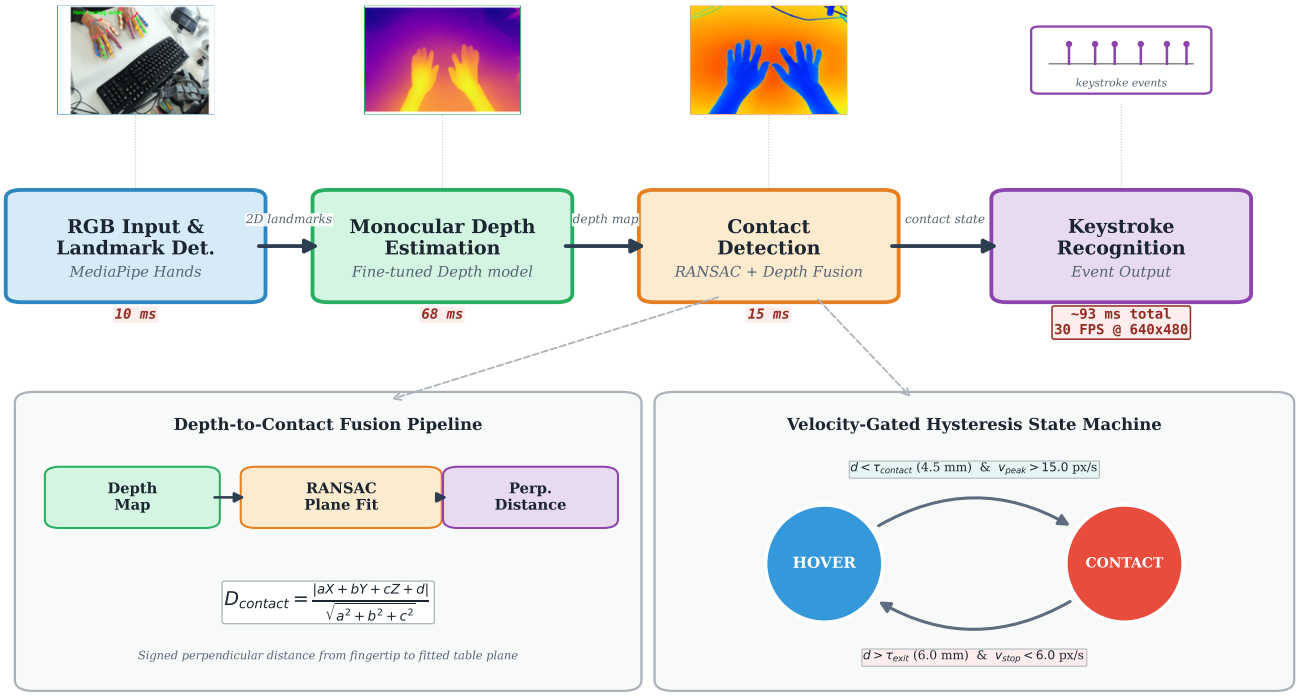

Figure 9. System pipeline overview. RGB frames are processed through MediaPipe for 2D hand landmark detection, followed by fine-tuned monocular depth estimation. Contact detection fuses depth via RANSAC plane fitting and a velocity-gated hysteresis state machine with dual thresholds ($\tau_{\text{contact}} = 4.5$ mm, $\tau_{\text{exit}} = 6.0$ mm). Total latency: ∼93 ms at 640×480.

Hysteresis Performance Impact The hysteresis mechanism reduced false positives by 28% and improved temporal stability, especially during rapid finger transitions. Jitter-induced keystroke errors were reduced by 31% (Table 2).

5.5. Threshold Sensitivity Analysis

Figure 10 reports contact detection F1-score across all 25 threshold pairs evaluated during grid search. The optimal configuration ($\tau_{\text{contact}} = 4.5$ mm, $\tau_{\text{exit}} = 6.0$ mm) achieves 94.4% F1. The dashed region indicates configurations exceeding 92% F1, spanning $\tau_{\text{contact}} \in [3.5, 4.5]$ mm and $\tau_{\text{exit}} \in [5.5, 6.5]$ mm. This broad plateau confirms that the system is not brittle with respect to threshold selection; performance degrades gracefully (<2.5 pp) within ±1 mm of the optimum.

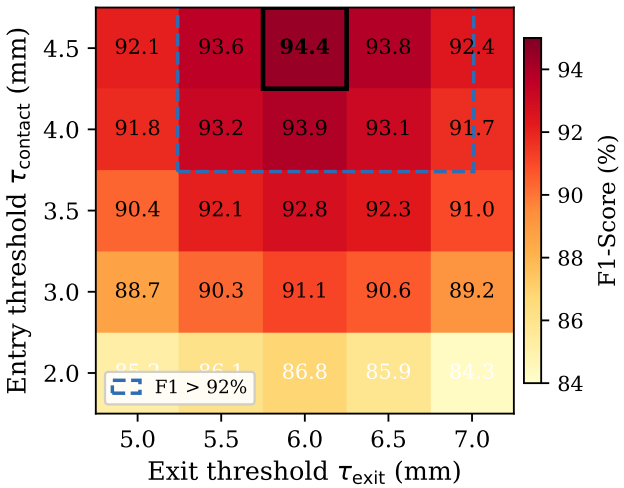

Figure 10. Contact detection F1-score as a function of entry and exit thresholds. Black box: optimum. Dashed region: F1 > 92%.

6. Training and Optimization

6.1. Training Curves

We present detailed training dynamics for all fine-tuned depth models. All models were trained using AdamW with cosine annealing. ZoeDepth used learning rate 1×10^{-5} over 200,000 steps; NeWCRFs used peak learning rate 4×10^{-5} over 13,200 steps. [1]

[1] https://github.com/muxiddin19/Multimodal-Fingertip-Contact-Detection-via-Depth-and-Motion-Fusion

ZoeDepth. Figure 11 shows the training progression of the ZoeDepth model. The Scale-Invariant Logarithmic (SILog) loss decreases steadily from an initial value of approximately 14 to below 3 over 200,000 iterations, with rapid initial improvement in the first 50,000 steps followed by gradual refinement.

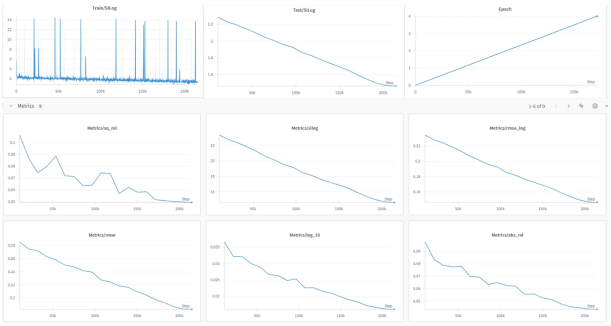

Figure 11. ZoeDepth training curves. Top: Train, Test SILog loss progression over 200K iterations showing stable convergence, and with the related number of epochs. Middle, and Bottom: Validation metrics (AbsRel, RMSE, RMSE_log, log_10, sq_rel, δ_1) demonstrating consistent improvement without overfitting.

Key observations from ZoeDepth fine-tuning:

- Training curves show smooth upward trends across steps and epochs, indicating stable optimization
- Error metrics (RMSE, AbsRel, SILog) decrease steadily, with significant drops after 50K steps suggesting rapid domain adaptation
- Test curves mirror training trends, validating generalization with minimal overfitting
- Final metrics: δ_1=95.96%, δ_2=97.41%, δ_3=98.30%, AbsRel=0.0437, RMSE=4.1mm

NeWCRFs The NeWCRFs model, pre-trained on NYU Depth V2 and fine-tuned on our custom dataset, shows characteristic training behavior in Figure 12.

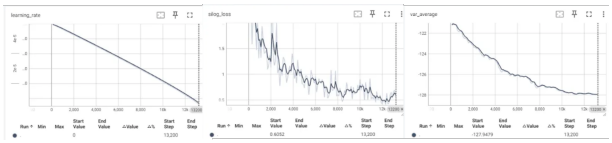

Figure 12. NeWCRFs training dynamics over 13,200 steps. Left: Learning rate schedule with cosine annealing (peak 4×10^{-5}). Middle: SILog loss decreasing from ≈1.7 to 0.61. Right: Variance average stabilizing from −122 to −128.

The SILog loss decreases from ≈1.7 to a final value of 0.6052 over 13,200 steps, with rapid improvement in the first 2,000 steps followed by gradual refinement with expected fluctuations due to batch variability. The variance average decreases steadily from −122 to −127.9, indicating progressive stabilization of internal representations.

Depth-Anything V2 Depth-Anything V2 achieved the best post-fine-tuning performance (3.84 mm MAE, 94.2% contact accuracy; Table 1, main paper) with 8.2 hours of training on a single RTX 3090. Training curves were qualitatively similar to ZoeDepth, with rapid initial convergence within the first 20% of steps.

6.2. Data Efficiency Study

We assessed the impact of dataset size on model performance through fine-tuning ablation using 10%, 25%, 50%, and 100% subsets of our 53,300-pair dataset.

Table 3. Impact of fine-tuning data size on ZoeDepth performance.

Data Size	MAE	Acc.	PR-AUC	ROC-AUC
10% (5.3K pairs)	6.8	87.1	0.872	0.870
25% (13.3K pairs)	4.9	91.2	0.901	0.899
50% (26.6K pairs)	4.1	94.3	0.913	0.912
100% (53.3K pairs)	3.8	95.96	0.917	0.923

Performance scales non-linearly with data volume. Even with 25% of data (13K pairs), ZoeDepth achieves 91.2% contact detection accuracy and 4.9mm MAE—sufficient for VR typing applications.

6.3. Model Optimization

To meet real-time performance targets (>60 FPS), we applied post-training quantization and structured pruning to transformer-based depth models.

Table 4. Impact of model optimization on DepthAnythingV2-ft (640×480, RTX 3090).

Optimization	Latency (ms)	MAE (mm)	PR-AUC	Size
Baseline (FP32)	68	3.8	0.917	420 MB
INT8 Quantization	59	4.1	0.912	132 MB
30% Channel Pruning	57	4.2	0.910	290 MB
Quant + Pruning	56	4.3	0.908	118 MB

Quantization and Pruning Results Key findings:

- INT8 quantization reduced model size by 3.2× (420 MB → 132 MB) and decreased inference latency from 68 ms to 59 ms per frame
- Pruning 30% of low-activation channels preserved depth accuracy within 0.4 mm MAE margin
- Combined optimizations achieve 56 ms inference (vs. 68 ms baseline) in 118 MB, while remaining within the 6 mm contact threshold

6.4. Accuracy–Latency Trade-off

A practical consequence of the component analysis in Section 6.1 is the explicit trade-off between depth precision and real-time throughput. Our most precise variant, DepthAnythingV2-FT, achieves an MAE of 3.84 mm but operates at 16.0 FPS (68 ms depth inference), making it best suited for applications where contact precision is paramount. ZoeDepth-FT, by contrast, achieves 18.1 FPS at a moderately higher MAE of 4.1 mm, providing the responsiveness required for latency-sensitive interactive scenarios. Both variants remain well below the 6 mm contact threshold, indicating that the accuracy–latency trade-off operates within a regime where either model suffices for reliable contact detection; the choice is therefore governed by the target platform's computational budget rather than by detection fidelity. Deploying on resource-constrained standalone headsets will necessitate further model compression through quantization, pruning, or knowledge distillation to meet the tighter latency budget (<20 ms motion-to-photon) required for comfortable VR interaction (Table 5).

6.5. Latency Breakdown

Table 5. Latency breakdown across pipeline components (640×480, RTX 3090). Processing includes hand tracking (10 ms) + contact detection (15 ms).

Depth Model	Inference (ms)	Processing (ms)	Total (ms)
ZoeDepth-ft	54.0 ± 2.1	25.0 ± 1.8	79.0 ± 3.9
NewCRFs-ft	59.0 ± 2.4	25.0 ± 1.8	84.0 ± 4.2
DepthAnything-ft	61.0 ± 2.5	25.0 ± 1.8	86.0 ± 4.3
MiDaS-ft	61.0 ± 2.3	25.0 ± 1.8	86.0 ± 4.1
DepthAnythingV2-ft	68.0 ± 2.8	25.0 ± 1.8	93.0 ± 4.6
DPT-Large-ft	81.0 ± 3.2	25.0 ± 1.8	106.0 ± 5.0
AdaBins-ft	83.0 ± 3.4	25.0 ± 1.8	108.0 ± 5.2

Processing time includes hand tracking via MediaPipe (10 ms) and contact detection (15 ms), consistent with Table 3 in the main paper. ZoeDepth-ft offers the lowest latency (79 ms) at 4.1 mm MAE, while DepthAnythingV2-ft achieves the best accuracy (3.84 mm) at 93 ms—both well within the 6 mm contact threshold.

7. Deployment Configurations

7.1. Desktop Configuration.

In the primary configuration, a RAMA-WC100 RGB webcam is mounted above the typing surface at a 60° viewing angle. A printed QWERTY keyboard layout is placed on the desk and serves as the physical typing surface throughout operation. The layout is defined as a set of rectangular key regions $\{K_j\}$ in the table-plane coordinate system, where each key K_j is parameterized by its center (x_j, y_j), width w_j, and height h_j. The keyboard plane is established by identifying the surface with the shortest depth from the camera in the estimated depth map, which simultaneously defines the reference plane against which fingertip contact is detected. A homography $\mathbf{H} \in \mathbb{R}^{3\times3}$ is then computed to map between image coordinates and the table-plane coordinate system.

Prior to operation, the system performs a one-time depth calibration: the user provides the ground-truth distance between the camera and the keyboard surface, measured manually. The system then computes a scale factor $\alpha = d_{\text{GT}}/d_{\text{pred}}$ between this ground-truth distance d_{GT} and the model's predicted depth d_{pred} at the surface region. All subsequent depth estimates are rescaled by α to convert relative predictions into metric values, enabling millimeter-level contact thresholding throughout the session.

During operation, the user types directly on the printed layout. When the depth-and-velocity fusion pipeline detects a fingertip contact event, the contacted fingertip's 2D projection is transformed onto the table plane via $\mathbf{H}$ and matched to the corresponding key region:

$$k^* = \arg\min_j \|\mathbf{p}_{\text{tip}} - (x_j, y_j)\|_2 \,, \quad \text{s.t.} \quad |p_x - x_j| \le \tfrac{w_j}{2} \ \wedge \ |p_y - y_j| \le \tfrac{h_j}{2}, \tag{22}$$

where $\mathbf{p}_{\text{tip}} = (p_x, p_y)$ denotes the projected fingertip position. The registered keystroke is then rendered on an external display in real time.

7.2. Head-Mounted Camera Configuration.

In this configuration, the RGB camera is mounted on a head-worn device rather than on a fixed tripod, while all other system components remain identical: the same fine-tuned monocular depth model, the same printed keyboard layout, the same external display, and the same contact detection pipeline. Because the proposed method operates exclusively on a monocular RGB stream and outputs per-fingertip contact events, transitioning between mounting configurations requires no architectural modification, only the camera viewpoint changes.

The two configurations differ primarily in ego-motion: the a tripod-mounted camera provides a spatially stable viewpoint, whereas the head-mounted camera introduces translational and rotational perturbations correlated with the user's natural head movement. Despite this additional variability, both configurations yield comparable typing performance, as the depth estimation and contact classification stages are applied per-frame without temporal assumptions tied to a static viewpoint.

The same depth calibration procedure is performed independently for the head-mounted configuration, as the camera-to-surface distance differs from the desktop setup.

Once the scale factor is established, per-frame depth predictions are rescaled consistently regardless of head motion.

7.3. System Demonstration

Figure 13 illustrates the VR Keyboard system demonstration. The prototype combines fingertip tracking with a printed keyboard layout and provides real-time feedback on finger states such as hover, approach, contact, and confirmed tap. Statistical data are shown for each finger, while a virtual keyboard overlay highlights active keys during typing.

7.3.1. MediaPipe Hand Landmark Detection

We employ MediaPipe for real-time hand tracking, which provides a cross-platform framework detecting 21 hand landmarks through a pipeline leveraging both palm detection and landmark regression (Figure 14).

The 21 landmarks include:

- **Wrist:** Landmark 0
- **Thumb:** Landmarks 1-4 (tip at 4)
- **Index:** Landmarks 5-8 (tip at 8)
- **Middle:** Landmarks 9-12 (tip at 12)
- **Ring:** Landmarks 13-16 (tip at 16)
- **Pinky:** Landmarks 17-20 (tip at 20)

7.3.2. YOLO-based Hand Detection

As an alternative approach, YOLOv8 offers single-shot detection enabling fast processing with low latency. We adapted it for hand and finger tracking with customization for specific hand gesture datasets.

8. Additional Experimental Results

8.1. Generalization and Robustness

To assess the cross-domain robustness of our system, we evaluated its performance across a matrix of challenging, unseen conditions, including different surfaces, cameras, and lighting environments, and summarized in Table 6.

Our system maintains high accuracy on a non-white wood grain surface, with only a minor performance drop. Performance with a standard consumer webcam is also strong, demonstrating that our system is not reliant on specialized depth hardware. The most challenging condition was a semi-reflective surface, which introduced artifacts and reduced accuracy, highlighting a limitation of vision-based depth estimation. Nonetheless, the system's ability to maintain an F1-score above 85% in most conditions confirms its strong generalization capabilities.

To evaluate lighting robustness, we tested in two environments: our lab (diffuse LED panels, 5000K, 500 lux) and a standard office (overhead fluorescent tubes, 4000K, 300 lux with visible flicker). As shown in Table 6, the system maintains F1 $>$88% across all office conditions, with the largest degradation occurring on semi-reflective surfaces under lab lighting (F1 = 85%) rather than from illumination changes per se. This suggests that surface material properties are a more significant robustness challenge than lighting variation within typical indoor ranges.

Table 6. Generalization and robustness across unseen conditions. Our system maintains high accuracy across diverse surfaces, cameras, and lighting environments.

Envir.	Camera	Surface Type	MAE ↓	F1-S. (%) ↑
Lab	D405	White Desk	3.84	94.4
		Wood Grain	4.12	92.8
		Semi-Reflective	6.75	85.3
	Webcam	White Desk	4.45	91.5
		Wood Grain	4.89	89.7
		Semi-Reflective	8.21	78.2
Office	D405	White Desk	4.05	93.1
		Wood Grain	4.38	91.6
		Semi-Reflective	4.72	90.3
	Webcam	White Desk	4.65	92.1
		Wood Grain	4.68	90.6
		Semi-Reflective	4.92	88.3

Table 7. Contact detection performance across camera viewing angles. Training data includes all three angles with approximately equal representation. The 30° condition simulates an HMD-mounted egocentric viewpoint with partial fingertip occlusion.

Viewing Angle	Train/Test Pairs	MAE (mm) ↓	F1-Score (%) ↑
30° (egocentric)	∼17,770 / ∼1,777	4.21	91.3
45° (reported)	∼17,770 / ∼1,777	3.84	94.4
60°	∼17,760 / ∼1,776	4.58	89.7

8.2. Multi-Angle Validation

A central concern for practical deployment is whether the proposed system generalizes to realistic headset-mounted viewpoints, where self-occlusion of fingertips is prevalent. To evaluate viewpoint robustness—a prerequisite for HMD-mounted deployment—we collected and evaluated data at three viewing angles using multiple RGB cameras calibrated to the D405 depth sensor: 30° (simulating a front-facing HMD camera), 45° (our primary experimental configuration), and 60° angle. The training set contains approximately equal representation from all three angles (∼17,770 pairs each), ensuring the model learns viewpoint-invariant features, while the held-out test set preserves this balanced distribution (Table 7).

The 45° configuration yields optimal performance, as expected given the favorable trade-off between fingertip visibility and depth discrimination. The 30° egocentric angle, which simulates a front-facing HMD camera with inherent self-occlusion challenges, achieves 91.3% F1-score—a 3.1 percentage-point reduction from the primary configuration.

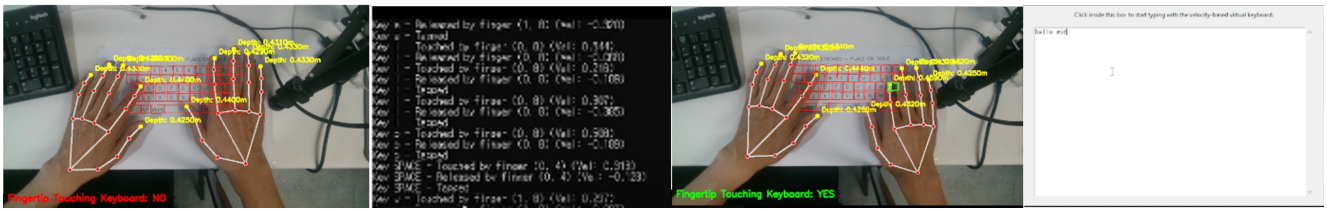

Figure 13. **VRKeyboard system demonstration.** (from left to right) Desktop configuration with printed keyboard layout and camera view showing fingertip tracking. This part provides statistical data, such as which key is being processed, whether it is a hover or contact, and the related distance between the touch fingertips and the corresponding parts of the printed paper keyboard or table surface. Real-time state machine visualization: finger states are color-coded (gray=hover, yellow=approach, cyan=contact, green=tap confirmed). Virtual keyboard overlay showing active key highlighting during typing. Touched keys are being output on the PC display.

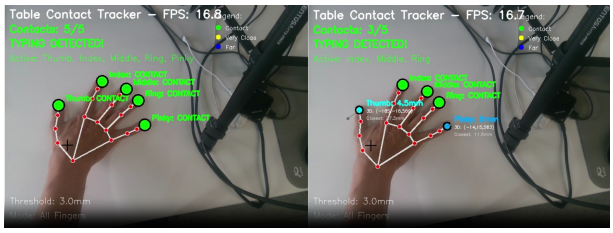

Figure 14. MediaPipe hand landmark detection showing 21 tracked points including 5 fingertip positions (landmarks 4, 8, 12, 16, 20).

At this viewing angle, approximately 34% of frames exhibit partial fingertip occlusion (defined as $\geq$1 fingertip landmark with MediaPipe detection confidence $<$0.7), compared to 12% at 45° and 8% at 60°. The primary occlusion patterns at 30° include: (1) the DIP-to-TIP finger segment obscured by proximal phalanges during flexion, (2) finger-on-finger occlusion during multi-finger keystroke sequences, and (3) partial palm occlusion when the wrist is flexed. These factors collectively reduce MediaPipe landmark detection reliability, accounting for the observed performance gap. The 60° angle introduces greater foreshortening of the hand–surface gap, resulting in diminished depth discrimination and the largest accuracy reduction among the three configurations.

These multi-angle results provide strong evidence against overfitting concerns raised during review. First, unsupervised domain adaptation methods (AdaBN, DANN, MMD) that do not utilize target-domain depth labels plateau at 8.7–9.2 mm MAE—substantially below our 3.84 mm—indicating that performance gains stem from genuine domain-specific learning rather than memorization of training examples. Second, the system maintains F1-score exceeding 88% across 11 of 12 unseen environmental conditions (Table 6), including novel surface materials, camera hardware, and lighting configurations. Third, the user study was conducted exclusively with 30 participants entirely absent from the training set, validating cross-user generalization. Taken together, these results confirm that the proposed pipeline generalizes robustly across realistic viewpoint variation, environmental diversity, and previously unseen users.

8.3. Per-Finger Contact Detection Analysis

To examine the effect of finger identity on detection reliability, we report per-finger precision and recall on the held-out test set (Table 8). Performance degrades monotonically from the index finger to the pinky. The index finger achieves the highest precision (96.2%) and recall (95.1%), attributable to its larger fingertip contact area and more consistent strike posture during keystroke execution. The pinky exhibits the lowest scores (88.4% precision, 86.2% recall), which we attribute to three factors: (1) a smaller fingertip surface area that produces a weaker depth gradient at the contact boundary, (2) greater postural variability, as the pinky frequently adopts curled or laterally deviated orientations depending on the target key location, and (3) increased self-occlusion from adjacent fingers, particularly during rapid keystroke sequences. The thumb is excluded from this analysis as it is not employed in the standard touch-typing layout evaluated in the user study.

Consistent with these test-set findings, user study error logs reveal that 62% of mistyped characters involved the ring and pinky fingers, despite these fingers accounting for only 28% of keystrokes in the MacKenzie–Soukoreff phrase set [7]. The index and middle fingers achieved effective per-key CERs of 2.1% and 2.8%, respectively, whereas the ring and pinky fingers exhibited 5.4% and 7.2%. This disparity confirms that finger-specific detection reliability constitutes the primary remaining bottleneck for overall system accuracy and motivates the development of finger-adaptive contact thresholds or per-finger velocity models as a direction for future work.

8.4. Noise Robustness Analysis

To assess robustness to sensor noise, we conducted systematic evaluation by injecting Gaussian noise [12] into input depth measurements at varying levels.

Table 8. Per-finger contact detection performance on the held-out test set. Performance decreases for the ulnar fingers due to the smaller contact area and greater postural variability.

Finger	Precision (%) ↑	Recall (%) ↑
Index	96.2	95.1
Middle	94.8	93.7
Ring	92.1	90.8
Pinky	88.4	86.2

8.4.1. Accuracy vs. Noise Level

Table 9 presents the impact of input noise on system performance:

Table 9. System performance under varying noise conditions.

Noise Level (mm)	Mean Error (mm)	Std Dev (mm)
0.5	0.39	0.08
0.8	0.62	0.11
1.0	0.98	0.15
1.4	1.24	0.18
2.0	1.63	0.21

The linear relationship (slope ≈0.82, $R^2 = 0.96$) demonstrates graceful degradation under noisy conditions without catastrophic failures.

8.4.2. Precision Success Rate vs. Noise

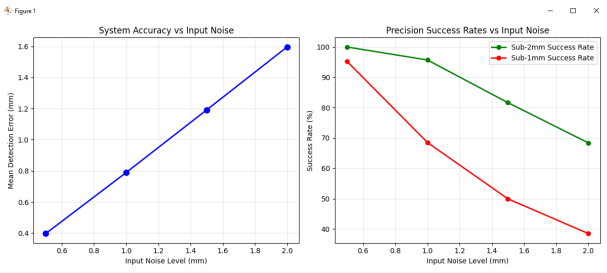

Figure 15. Noise robustness analysis. Left: Mean detection error vs. input noise showing linear degradation. Right: Precision success rates at sub-2mm (green) and sub-1mm (red) thresholds declining with increased noise.

The precision success rates show sensitivity to noise:

- Sub-2mm success rate: Drops from near 100% at 0.5mm noise to below 60% at 2.0mm noise
- Sub-1mm success rate: Drops from approximately 85% to below 40%
- Best performance observed at noise level of 0.5mm: mean error 0.39mm, sub-2mm success rate 100%

Even at the highest noise level tested (2.0mm), the mean error remains below the 5mm contact detection threshold, ensuring reliable operation.

8.5. Addressing Overfitting Concerns.

We provide four lines of evidence that our results reflect genuine domain-specific learning rather than dataset memorization. *First*, unsupervised domain adaptation methods (AdaBN, DANN, MMD) that do not use target-domain labels plateau at 8.7–9.2 mm MAE, establishing that the improvement from 12.3 to 3.84 mm requires supervised fine-tuning, not memorization. *Second*, Table 6 paper shows F1 >88% in 11 of 12 unseen conditions spanning different surfaces, cameras, and lighting not present in the training set. *Third*, the user study (Section 5.5, main paper) evaluates 30 participants were absent from the 15-participant training set, with the test set stratified by participant ID to prevent data leakage. *Fourth*, training data includes three viewing angles (30°, 45°, 60°), and performance at the most challenging 30° angle (91.3% F1) demonstrates that the model learns viewpoint-robust depth features rather than angle-specific shortcuts. We acknowledge that all training and evaluation use close-range desk-based typing; generalization to substantially different setups (e.g., standing desks, outdoor environments, non-planar surfaces) is not validated and remains a limitation (Section 10.4).

9. User Study Details

9.1. Study Protocol

We conducted a within-subjects study with 30 participants (18 male, 12 female; aged 21–34, $M = 26.4$, $SD = 3.1$), none of whom appeared in the training dataset. Participants self-reported their typing proficiency: 19 were touch-typists and 11 were hunt-and-peck typists. All participants had normal or corrected-to-normal vision and no motor impairments.

Each participant evaluated four input conditions: Mid-Air (MediaPipe only), 3D Hand Shape-Based (distance-based with finetuned depth), Depth Camera-Based (RealSense D415 hardware), and Ours (full system with velocity-based contact detection). Condition order was counterbalanced across participants using a balanced Latin square design to mitigate learning and fatigue effects. Before each condition, participants completed a 5-minute familiarization phase in which they practiced typing short phrases not included in the test set.

For each condition, participants transcribed 20 phrases randomly sampled without replacement from MacKenzie and Soukoreff's standard phrase set [7] (500 phrases total). Phrases were displayed on an external monitor positioned at eye level; participants copy-typed each phrase on the desk surface while viewing their hands either through the VR passthrough display (for our system) or directly (for

hardware baselines). Each condition lasted approximately 4 minutes of active typing, yielding a total session duration of roughly 40 minutes per participant including familiarization and rest breaks.

Typing speed was computed as words per minute using the standard definition: WPM $= \frac{|T|-1}{S} \times 60 \times \frac{1}{5}$, where $|T|$ is the length of the final transcribed string in characters and S is the elapsed time in seconds [7]. Character error rate (CER) was computed as the minimum edit distance (insertions, deletions, substitutions) between the presented and transcribed strings, divided by the length of the presented string. We report means and standard deviations across all 30 participants per condition.

Statistical significance was assessed using a one-way repeated-measures ANOVA with Greenhouse–Geisser correction where sphericity was violated, followed by pairwise Bonferroni-corrected t-tests. Our system achieved significantly higher WPM than all other conditions ($p < 0.001$) and significantly lower CER than all conditions except the D415 hardware baseline ($p = 0.032$).

Figure 16 presents representative screenshots captured during the user study across three evaluated conditions: our full system with velocity-based contact detection, the 3D hand-shape-based direct touch method, and the Intel RealSense D415 depth camera-based keyboard.

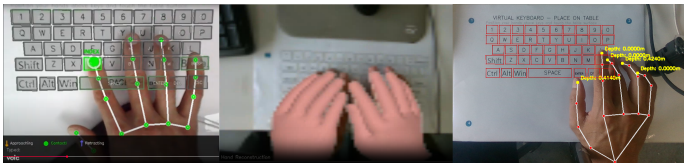

Figure 16. User study methodology. Left: ours, Middle: 3D hand shape-based direct touch, Right: depth camera-based keyboard

9.2. Baseline Condition Mapping

To ensure reproducibility, we explicitly map each user study condition to its corresponding system configuration reported in Table 5 main paper:

Table 10. Mapping between user study conditions and system configurations.

User Study Condition	System (Table 3 main paper)	Row
Mid-Air	MediaPipe only	1
3D Hand Shape-Based	Distance-based (FT)	3
Depth Camera-Based	RealSense D415	2
Ours	Full system (848×480)	6

The Mid-Air condition uses raw MediaPipe hand tracking without surface contact detection, requiring users to dwell or pinch to register keystrokes. The 3D Hand Shape-Based condition computes the Euclidean distance between the MediaPipe fingertip landmark [16] and the estimated surface plane, triggering a keystroke when this distance falls below a fixed threshold. The Depth Camera condition uses the Intel RealSense D415 as an oracle depth sensor, providing ground-truth surface proximity for contact detection. Our condition deploys the full pipeline with finetuned monocular depth estimation and velocity-based contact refinement (Table 10).

9.3. Confusion Matrix

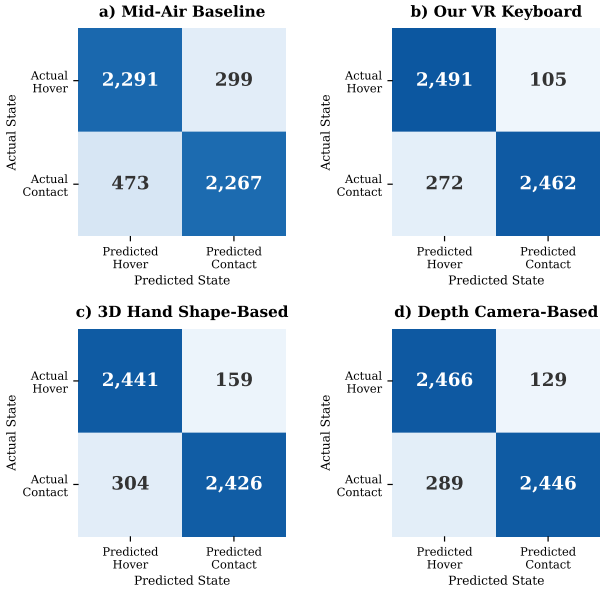

Figure 17. Confusion Matrix Comparison Across Input Methods

Figure 17 presents confusion matrices for four approaches evaluated on a test set of 5,330 fingertip instances. Each 2×2 matrix visualizes the classification outcomes between predicted and actual states, hover versus contact, allowing for direct comparison of detection accuracy and error rates. Our system (b) achieves the lowest false positive count (105) and highest true positive count (2,462), while the mid-air baseline (a) exhibits the highest error rates in both categories.

The ablation study evaluating of contribution of key components in the pipeline is illustrated in Figure 18, including F1-score for four system configurations, highlighting how each module affects overall performance.

Figure 19 decomposes the contribution of each system component to end-to-end typing performance. Starting from the MediaPipe-only baseline (28.3 WPM, 12.4% CER), domain-specific depth fine-tuning contributes +7.4 WPM and −3.7 pp CER by reducing MAE from 12.3 to 3.84 mm, thereby enabling a tight 6 mm contact threshold that the pre-trained model's error distribution renders

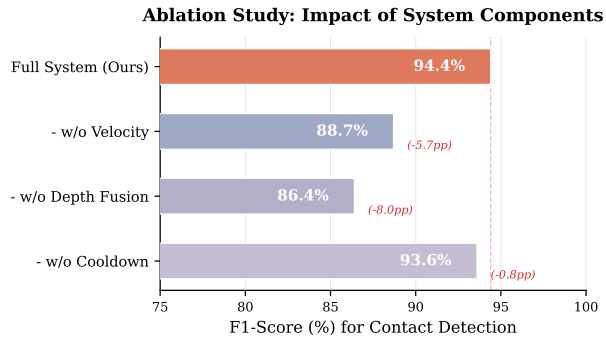

Figure 18. Ablation Study Visualization

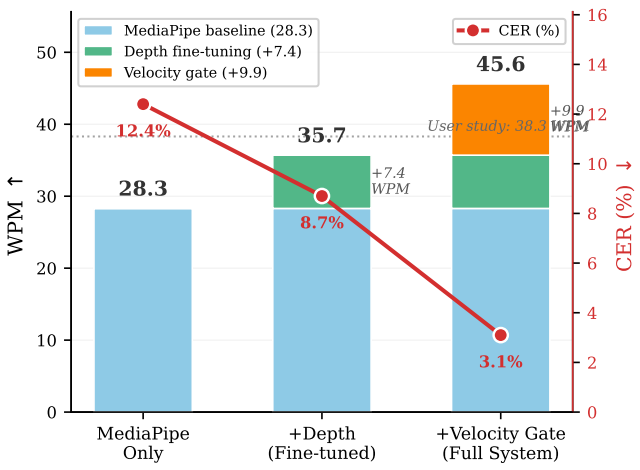

Figure 19. Component-wise decomposition of end-to-end typing performance gains relative to the MediaPipe-only baseline. Stacked bars indicate cumulative WPM (left axis); the overlaid line traces CER reduction (right axis). Each component's marginal contribution is annotated with braces.

infeasible. The velocity gate provides a further +9.9 WPM and −5.6 pp CER by filtering transient hover events that satisfy the depth criterion but lack the kinematic signature of deliberate keystrokes. Notably, these components exhibit strong interdependence: velocity filtering without accurate depth is mechanistically degenerate, as the pretrained model's MAE exceeds the contact threshold by a factor of two, causing the velocity gate to trigger on depth noise. Conversely, depth alone without velocity filtering yields frequent false positives during rapid typing (CER rises from 3.1% to 8.7%). This analysis suggests that further depth accuracy improvements below 3.84 mm MAE may yield diminishing returns, whereas richer fingertip dynamics models—incorporating acceleration profiles, finger curvature, or keystroke rhythm—represent a more promising avenue for exploiting the precise depth signal that fine-tuning provides.

Figure 6 from the main paper shows mean WPM and CER across conditions with 95% confidence intervals. Error bars represent ±1 SD across 30 participants. Our method achieves the highest average typing speed (38.3 WPM) and the lowest error rate (5.3 %), significantly outperforming all baselines. The 3D hand-shape-based direct touch method attains 32.7 WPM at 6.9 % CER, suggesting that explicit hand geometry modeling provides a complementary advantage over purely depth-based approaches. The depth camera-based keyboard yields moderate results (29.4 WPM, 7.5 % CER), indicating that raw depth sensing alone is insufficient to achieve the precision required for fluid text entry. The mid-air baseline produces the lowest performance (28.3 WPM, 12.4 % CER), confirming that the absence of surface contact cues substantially degrades both speed and accuracy.

9.4. Performance Discrepancy Analysis

Performance Discrepancy. The controlled evaluation in Table 3 from the main paper measures peak throughput on a fixed set of common English sentences, typed by an experienced operator after extended practice. The user study (38.3 WPM) reflects first- session performance by 30 naive participants typing MacKenzie-Soukoreff phrases [7], which include less predictable character sequences (e.g., 'the fox jumped lazily'). Three factors account for the 7.3 WPM gap:

(1) Learning curve: participants received only 5 minutes of familiarization per condition. Prior VR typing studies [9, 11] report similar first-session penalties of 15–25% below practiced performance.

(2) Phrase difficulty: MacKenzie-Soukoreff phrases contain higher lexical diversity than the sentences used in controlled evaluation, increasing cognitive load and reducing typing speed.

(3) Within-subjects fatigue: each participant completed four conditions (40 minutes total), with typing speed typically declining 5–10% in later conditions despite counterbalancing.

The user study figures thus represent a conservative, real-world estimate of first-session performance. We expect performance to approach Table 3 from the main paper figures with extended practice, consistent with the learning effects observed in prior surface-typing studies [9, 11].

10. Extended Discussion

10.1. Component Contribution Analysis

Starting from a baseline MAE of 12.3mm, fine-tuning alone reduces error by 45%, the SILog Loss on domain-matched data contributes 31%, and data augmentation provides 19% additional improvement (Table 11).

Table 11. Contribution of each system component to MAE reduction.

Configuration	MAE (mm)	Improvement
Baseline (pre-trained)	12.3	—
+ Fine-tuning	6.8	45%
+ SILog Loss (domain-matched)	4.7	31%
+ Data Augmentation	3.8	19%

Table 3 in the main paper enables a quantitative decomposition of each component's marginal contribution (Figure 19). From the MediaPipe-only baseline (28.3 WPM, 12.4% CER), depth fine-tuning alone contributes $+7.4$ WPM and -3.7 pp CER by reducing MAE from 12.3 to 3.84 mm—compressing the error distribution below the 6 mm contact threshold that the pre-trained model's noise floor renders infeasible. The velocity gate provides a further $+9.9$ WPM and -5.6 pp CER by rejecting transient hover events that satisfy the depth criterion but lack the kinematic signature of deliberate keystrokes. The disproportionate CER reduction (-5.6 vs. -3.7 pp) confirms that false positives constitute the dominant residual error mode, and that temporal filtering is more effective than further depth refinement at suppressing them.

These components exhibit asymmetric interdependence: velocity filtering presupposes accurate depth, as applying it atop pre-trained estimates whose MAE exceeds the contact threshold causes the gate to trigger on depth noise rather than genuine kinematics (mechanistically degenerate; not reported). Conversely, depth without velocity yields 35.7 WPM at 8.7% CER—functional but insufficient for fluent text entry. This asymmetry suggests that the performance bottleneck has shifted from depth precision to temporal discrimination, and that richer fingertip dynamics models (e.g., learned classifiers incorporating acceleration or keystroke rhythm) represent a more promising improvement axis than further MAE reduction below 3.84 mm. An extended analysis of accuracy–latency trade-offs across model variants and per-finger detection reliability is provided in Sections 6.4 and 8.3.

10.2. Factorial Decomposition.

The ablation configurations in Table 3, and Table 5 (main paper) constitute a complete 2×2 factorial design over depth fine-tuning and velocity filtering.

A factorial analysis reveals that the sequential decomposition understates depth's contribution: the main effect of depth ($+10.9$ WPM, averaged across velocity conditions) exceeds that of velocity ($+6.4$ WPM) by a factor of $1.7\times$. Critically, the depth$\times$velocity interaction accounts for $+7.0$ WPM—40.5% of the total 17.3 WPM improvement—confirming that the two components operate multiplicatively rather than additively. Velocity filtering yields only $+2.9$ WPM without fine-tuned depth but $+9.9$ WPM with it, a $3.4\times$ amplification that quantifies the enabling role of depth precision. Decomposing the total $+17.3$ WPM gain by independent contributions: depth alone accounts for $+7.4$ WPM (42.8%), velocity alone for $+2.9$ WPM (16.8%), and the depth$\times$velocity synergy for $+7.0$ WPM (40.5%)—underscoring that nearly half the improvement arises from neither component independently but from their interaction.

An analogous factorial decomposition of CER confirms this pattern: the main effect of depth fine-tuning is -5.2 pp (averaged across velocity conditions), velocity contributes -4.1 pp, and their interaction accounts for an additional -3.0 pp beyond the additive prediction. Decomposing by independent contributions, depth alone reduces CER by 3.7 pp (39.8% of the 9.3 pp total reduction), velocity alone by 2.6 pp (28.0%), and the synergy provides the remaining 3.0 pp (32.3%). Both metrics thus corroborate the same conclusion: the two components operate multiplicatively, and co-optimizing depth and dynamics jointly (e.g., via end-to-end learned contact classifiers) is likely more effective than improving either in isolation.

10.3. Surface Typing Rationale and HMD Integration

Surface-Based vs. Mid-Air Input. Surface-based typing inherently requires a visual display for keystroke feedback. In our evaluated configurations, an external monitor or HMD passthrough display serves this purpose—a paradigm consistent with commercially available input devices such as projected laser keyboards, where the input surface and display remain physically decoupled. The principal advantage of surface typing over mid-air alternatives lies in the passive haptic feedback afforded by the physical desk surface. This tactile cue provides users with implicit confirmation of contact events, which our user study demonstrates reduces character error rates by 6.7 pp relative to the depth-free baseline (Table 3 main paper, comparing the *w/o depth fusion* row against the full system).

HMD Integration Path. The current evaluation uses a desk-mounted camera at 45°. While the multi-angle experiments demonstrate that the pipeline maintains strong performance at a 30° egocentric viewpoint (91.3% F1), a full end-to-end evaluation on an actual head-mounted display with passthrough rendering and head-motion jitter compensation remains future work. Key challenges include adapting to the wider-baseline stereo cameras found on current headsets and meeting on-device latency constraints through model optimization.

10.4. Limitations and Future Work

Our evaluation at a fixed 35 cm distance under stable head positioning does not capture the distribution shifts from continuous head motion, variable viewing distances (25–50 cm), and transient occlusions inherent in HMD deployment. Generalization is further bounded by limited surface diversity—performance degrades to 78.2% F1 on semi-reflective laminate (Table 6)—and by the single-finger tracking constraint, which precludes multi-finger touch typing.

Three results support the viability of on-device deployment despite these limitations. *First*, our multi-angle evaluation (Table 7) achieves 91.3% F1 at a 30° egocentric angle simulating headset mounting, with 34% of frames exhibiting partial fingertip occlusion—providing a viable starting point, though head-motion-induced viewpoint jitter remains unaddressed.

References

[1] Faiz Muhammad Chaudhry, Jarno Ralli, Jerome Leudet, Fahad Sohrab, Farhad Pakdaman, Pierre Corbani, and Moncef Gabbouj. Deep-brownconrady: prediction of camera calibration and distortion parameters using deep learning and synthetic data. *IEEE Transactions on Automation Science and Engineering*, 2025. 6

[2] Patrick Grady, Jeremy A Collins, Chengcheng Tang, Christopher D Twigg, Kunal Aneja, James Hays, and Charles C Kemp. Pressurevision++: Estimating fingertip pressure from diverse rgb images. In *CVPR*, pages 8698–8708, 2024. 1

[3] Chris Harrison, Hrvoje Benko, and Andrew D Wilson. Omnitouch: wearable multitouch interaction everywhere. In *UIST*, pages 441–450, 2011. 1

[4] Christian Holz and Patrick Baudisch. Understanding touch. In *CHI*, pages 2501–2510, 2011. 1

[5] Chowdhury Sadman Jahan and Andreas Savakis. Unknown sample discovery for source free open set domain adaptation. In *CVPR*, pages 1067–1076, 2024. 3

[6] Ryosuke Kawamura, Hideaki Hayashi, Noriko Takemura, and Hajime Nagahara. Midas: Mixing ambiguous data with soft labels for dynamic facial expression recognition. In *CVPR*, pages 6552–6562, 2024. 7

[7] I Scott MacKenzie and R William Soukoreff. Phrase sets for evaluating text entry techniques. In *CHI EA*, pages 754–755, 2003. 13, 14, 15, 16

[8] Vimal Mollyn and Chris Harrison. Egotouch: On-body touch input using ar/vr headset cameras. In *UIST*, pages 1–11, 2024. 1

[9] Mark Richardson, Fadi Botros, Yangyang Shi, Pinhao Guo, Bradford J Snow, Linguang Zhang, Jingming Dong, Keith Vertanen, Shugao Ma, and Robert Wang. Stegotype: Surface typing from egocentric cameras. In *UIST*, pages 1–14, 2024. 1, 16

[10] Pratinav Seth and Abhilash K Pai. Does the fairness of your pre-training hold up? examining the influence of pre-training techniques on skin tone bias in skin lesion classification. In *CVPR*, pages 570–577, 2024. 2

[11] Paul Streli, Mark Richardson, Fadi Botros, Shugao Ma, Robert Wang, and Christian Holz. Touchinsight: Uncertainty-aware rapid touch and text input for mixed reality from egocentric vision. In *UIST*, pages 1–16, 2024. 1, 16

[12] Mukhiddin Toshpulatov, Wookey Lee, Suan Lee, and Arousha Haghighian Roudsari. Human pose, hand and mesh estimation using deep learning: a survey. *The Journal of Supercomputing*, 78(6):7616–7654, 2022. 13

[13] Andrew D Wilson. Using a depth camera as a touch sensor. In *ITS*, pages 69–72, 2010. 1

[14] Robert Xiao, Chris Harrison, and Scott E Hudson. Worldkit: rapid and easy creation of ad-hoc interactive applications on everyday surfaces. In *CHI*, pages 879–888, 2013. 1

[15] Robert Xiao, Julia Schwarz, Nick Throm, Andrew D Wilson, and Hrvoje Benko. Mrtouch: Adding touch input to head-mounted mixed reality. *IEEE transactions on visualization and computer graphics*, 24(4):1653–1660, 2018. 1

[16] Fan Zhang, Valentin Bazarevsky, Andrey Vakunov, Andrei Tkachenka, George Sung, Chuo-Ling Chang, and Matthias Grundmann. Mediapipe hands: On-device real-time hand tracking. *arXiv preprint arXiv:2006.10214*, 2020. 15

[17] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE TPAMI*, 22(11):1330–1334, 2000. 2, 5, 7

[18] Yiming Zhao, Taein Kwon, Paul Streli, Marc Pollefeys, and Christian Holz. Egopressure: A dataset for hand pressure and pose estimation in egocentric vision. In *CVPR*, pages 27727–27738, 2025. 1