

---

# ToF-IP: Time-of-Flight Enhanced Sparse Inertial Poser for Real-time Human Motion Capture

## - Supplementary Materials -

---

Yuan Yao<sup>1</sup>   Shifan Jiang<sup>1</sup>   Yangqing Hou<sup>1</sup>   Chengxu Zuo<sup>1</sup>   Xinrui Chen<sup>1</sup>

Shihui Guo<sup>1\*</sup>

Yipeng Qin<sup>2</sup>

<sup>1</sup> School of Informatics, Xiamen University, China

<sup>2</sup> School of Computer Science & Informatics, Cardiff University, UK

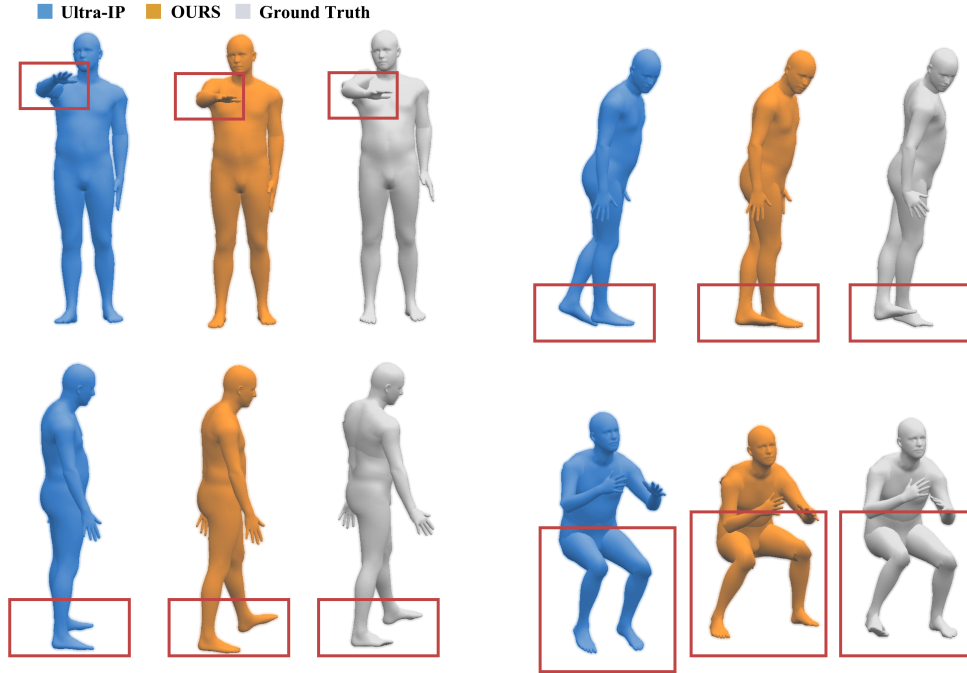


Figure 1: Qualitative comparisons with Ultra-IP on the DIP dataset.

## 1 Additional Experiments

### 1.1 Comparison with Ultra-IP

To better contextualize our method with recent progress in sparse motion capture, we compare our ToF-IP with Ultra-IP [1], a concurrent method that combines sparse IMUs with ultra-wideband (UWB) ranging modules. While both methods aim to enhance inertial-only pose estimation with distance-

---

\*Corresponding author.

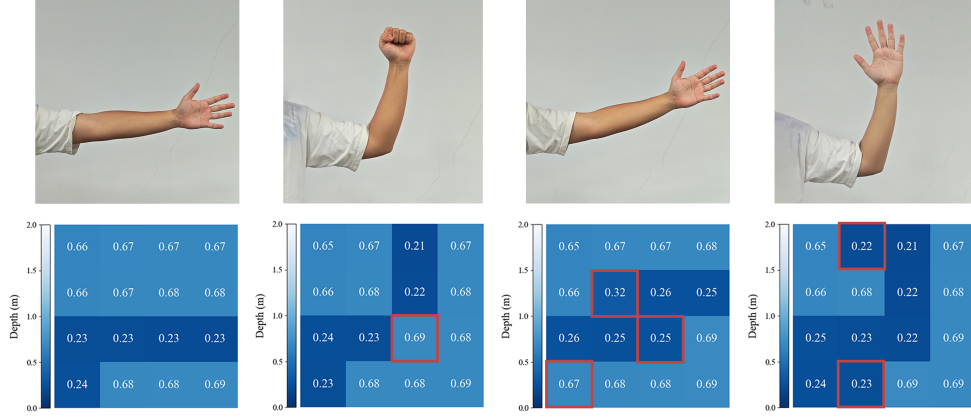


Figure 2: Visual comparison between RGB frames (top) and corresponding ToF depth maps (bottom) during two elbow flexion cycles. Depth artifacts caused by boundary ambiguity and pixel mixing are highlighted with red boxes, where foreground regions are misestimated with background-like depths.

based measurements, they differ fundamentally in sensing philosophy: Ultra-IP is ego-centric only and environment-agnostic, whereas ToF-IP is both ego-centric and environment aware.

Specifically, Ultra-IP leverages UWB modules to measure pairwise distances between sensors at different body joints. These distances are purely relative and invariant to global rigid transformations (e.g., translating/rotating the entire body), as they **only include inter-sensor geometry without environmental interaction**. Our ToF-IP, in contrast, equips with a low-resolution time-of-flight (ToF) sensor that outputs a  $4 \times 4$  depth image representing local environmental geometry. These distance maps **captures directional depth cues relative to both user’s body and environment** (e.g., distances to the ground). This makes it sensitive to egocentric and environmental context and achieve more accurate joint position determination.

For fair evaluation, we follow the data synthesis method provided in Ultra-IP on the DIP dataset: using ground-truth pairwise distances as UWB inputs, while ToF-IP uses synthesized ToF frames rendered from SMPL meshes with matching field-of-view. As shown in Table 1, the environmental awareness of ToF-IP brings better performance on key spatial metrics. Compared to Ultra-IP, it achieves lower joint position error (4.59 cm vs. 5.05 cm), endpoint error (6.65 cm vs. 7.09 cm), and jitter (0.17 vs. 0.24). We notice that the SIP of Ultra-IP is slightly lower than that of ToF-IP. This is because in the motion of some SIP-related joints, there may be a lack of reference objects in the ToF’s field of view. For example, in a "surrender-like" movement where the palm faces forward and is raised, the ToF’s perspective may not include the user’s body or the ground, resulting ToF data ineffective. In contrast, UWB can still provide valid sensor-to-sensor distance measurements. This implies that simultaneously using ToF and UWB to enhance inertial motion capture may bring greater improvements.

## 1.2 Analysis of ToF Sensing Errors

In this work, we employ the VL53L8CX ToF sensor in  $4 \times 4$  multizone mode at a frame rate of 60 Hz. All measurements are collected in controlled indoor environments, minimizing ambient infrared interference and ensuring that target surfaces fall within the sensor’s calibrated reflectance range. The sensor features a wide  $45^\circ$  diagonal field of view (FoV), enabling simultaneous ranging across 16 spatial zones. To maintain consistency with the sensor’s internal calibration and noise modeling, we utilize its native distance outputs without applying additional external filtering. To ensure spatial relevance to human-scale motion, we further truncate all depth values beyond 2 meters, discarding measurements that fall outside this effective sensing range. The onboard histogram-based signal processing and crosstalk compensation effectively reduce spurious measurements, yielding stable performance with typical ranging accuracies of  $\pm 5\%$  for high-reflectivity and  $\pm 8\%$  for low-reflectivity targets under moderate lighting conditions [5].

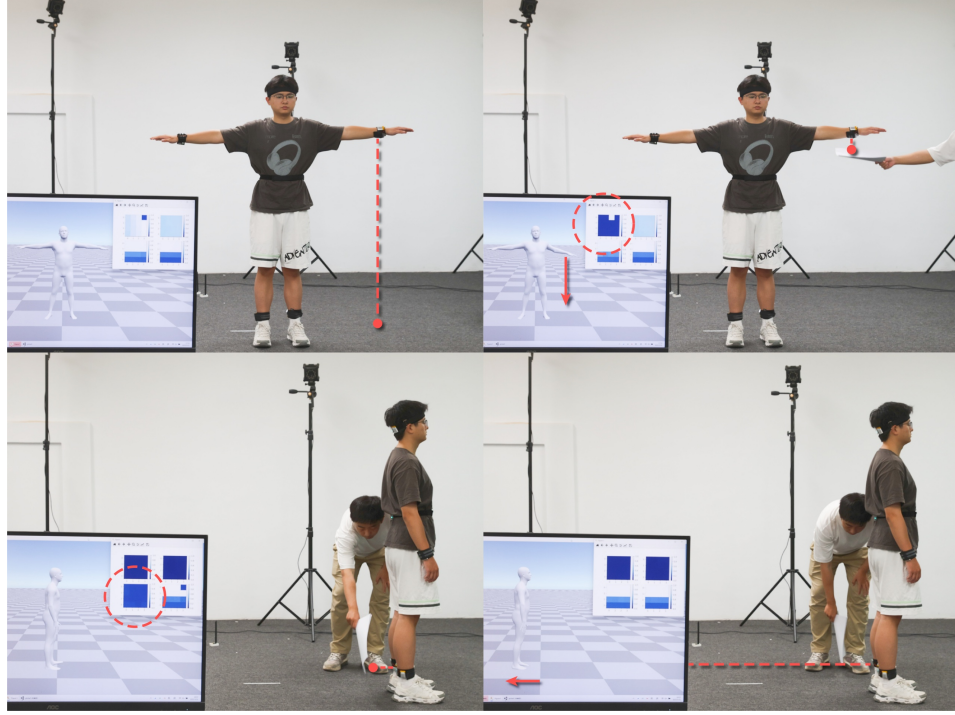


Figure 3: Illustration of two representative failure cases in ToF sensing.

Although our ToF-IP framework benefits from direct depth measurements, the low spatial resolution of commercial ToF sensors inherently introduces sensing noise and structural ambiguities, especially near motion boundaries. To better understand the limitations of ToF measurements in our system, we analyze representative examples captured during an elbow flexion cycle. Fig. 2 shows a set of synchronized optical and ToF depth frames sampled across a flexion motion. The optical images (top row) serve as ground truth visual references, while the bottom row displays the corresponding  $4 \times 4$  ToF depth grids. We observe that while general limb configurations are preserved, several critical artifacts occur at occlusion and silhouette boundaries.

Specifically, the red boxes highlight regions where the ToF sensors fail to differentiate between body and background, misattributing near-field pixels with far-field values. These errors are primarily caused by the wide field-of-view of each ToF zone, which introduces pixel-level ambiguity at object boundaries, and by mixed-pixel effects where foreground and background surfaces fall within the same zone, resulting in biased or fused depth values. Despite these limitations, our Dynamic Spatial Positional Encoding learns to attenuate such inconsistencies through temporal context and inter-node correlations, as evidenced by the overall robustness of our pose predictions.

### 1.3 Failure Case Analysis

To further examine the role of ToF depth in our framework, we analyze two representative failure scenarios involving partial or intermittent occlusion of ToF sensors, as illustrated in Fig. 3. Additional dynamic examples and motion sequences are provided in the supplementary video for further qualitative understanding.

**Case 1: Wrist Occlusion and Reversible Local Errors.** In the top row of Fig. 3, a flat object occludes the wrist-mounted ToF sensor, leading to a sudden drop in measured depth. The model interprets this as a lowering of the arm. Once the occlusion is removed, ToF readings stabilize, and the predicted pose quickly recovers. This reversible effect confirms that wrist ToF directly informs local limb estimation and that transient disruptions can be compensated once reliable depth returns.

Table 1: Comparison of methods on the DIP dataset. ToF-IP is evaluated with synthesized ToF signals, and Ultra-IP uses synthesized perfect distances.

Method	DIP				
	SIP Err	Ang Err	Pos Err	EndPos Err	Jitter
Ultra-IP	<b>13.20</b>	7.87	5.05	7.09	0.24
ToF-IP (Ours)	13.62	<b>6.75</b>	<b>4.59</b>	<b>6.65</b>	<b>0.17</b>

**Case 2: Ankle Occlusion and Irreversible Global Drift.** In contrast, the bottom row shows a cyclic occlusion of the ankle-mounted ToF sensor. The intermittent signal corruption introduces fluctuations in the inferred foot-to-ground velocity, which is subsequently integrated to estimate global root translation. This misleads the model into perceiving unintended motion, resulting in cumulative drift of the global body position. Unlike Case 1, the error persists even after signal restoration, as the velocity integration has already propagated incorrect translation.

**Discussion** These failure modes validate our core design rationale. The contrasting outcomes highlight the asymmetric sensitivity of our system: ToF signals on the upper limbs influence pose estimation locally and reversibly, whereas ToF at the feet directly governs global translation through velocity integration, making its stability critical. Rather than undermining our method, these failures underscore the indispensable contribution of ToF depth in our pipeline. In our intended use case—unobstructed indoor environments—ToF provides consistent, directional geometric cues that significantly enhance both local pose fidelity and global stability.

## 2 Proof of IMU Acceleration Accumulated Error

Under a sparse IMU configuration, only a subset of joint orientations can be directly measured. To compensate for missing measurements, acceleration data is commonly used as an additional input [4], as it carries implicit cues about joint positions that can aid in inferring unobserved joint orientations. In principle, joint positions can be obtained by double-integrating the acceleration signals over time. However, in practice, real-world acceleration measurements are prone to various sources of error, such as sensor noise and signal drift, inevitably resulting in significant error accumulation over time:

**Proposition 2.1** (Error Accumulation Analysis). *Following standard statistical practice, we assume that the acceleration measurement error at any timestamp  $\tau \in (0, t)$  follows a normal distribution  $\epsilon_a(\tau) \sim \mathcal{N}(\mu, \sigma^2)$ . Then, we have:*

- *Distribution of joint velocity error:  $\epsilon_v(t) \sim \mathcal{N}(\mu t, \sigma^2 t)$*
- *Distribution of joint position error:  $\epsilon_s(t) \sim \mathcal{N}(\frac{1}{2}\mu t^2, \frac{1}{2}\sigma^2 t^2)$*

*Proof.* We derive the error distributions through integration of stochastic processes, leveraging properties of normal distributions and Itô calculus.

**1. Velocity Error Distribution** By definition, velocity is the integral of acceleration over time:

$$\epsilon_v(t) = \int_0^t \epsilon_a(\tau) d\tau \quad (1)$$

Given  $\epsilon_a(\tau) \sim \mathcal{N}(\mu, \sigma^2)$  for all  $\tau \in (0, t)$ , the integral of independent and identically distributed (i.i.d.) Gaussian processes results in:

$$\epsilon_v(t) \sim \mathcal{N}\left(\int_0^t \mu d\tau, \int_0^t \sigma^2 d\tau\right) = \mathcal{N}(\mu t, \sigma^2 t) \quad (2)$$

This follows from the linearity of expectation and variance for i.i.d. Gaussian variables.

**2. Position Error Distribution** Position is the double integral of acceleration:

$$\epsilon_s(t) = \int_0^t \int_0^\tau \epsilon_a(s) ds d\tau \quad (3)$$

We apply Fubini's theorem to change the order of integration:

$$\epsilon_s(t) = \int_0^t \epsilon_a(s) \cdot (t - s) ds \quad (4)$$

The mean and variance of  $\epsilon_s(t)$  are computed as:

$$\mathbb{E}[\epsilon_s(t)] = \int_0^t \mu \cdot (t - s) ds = \mu \cdot \frac{1}{2}t^2 \quad (5)$$

$$\text{Var}[\epsilon_s(t)] = \int_0^t \sigma^2 \cdot (t - s)^2 ds = \sigma^2 \cdot \frac{1}{3}t^3 \quad (6)$$

However, as inertial motion capture is discrete-time sampling system, the position error variance is commonly approximated as  $\frac{1}{2}\sigma^2t^2$  instead of the continuous-time result  $\frac{1}{3}\sigma^2t^3$ . This approximation arises due to:

- **Correlated Measurement Noise:** Unlike ideal white noise, real inertial sensors exhibit temporal correlations in errors (e.g., bias drift), leading to error accumulation resembling a random walk process where variance grows quadratically with time.
- **Engineering Practicality:** The  $\frac{1}{2}\sigma^2t^2$  form provides a conservative yet tractable estimate, balancing accuracy and analytical simplicity for control system design and error budgeting.
- **Sampling Effects:** Discrete integration over finite time steps introduces error propagation patterns that deviate from continuous models, making the quadratic approximation more representative of actual system behavior.

This approximation is widely adopted in inertial navigation systems (INS) and motion tracking applications [2, 3] to efficiently model error accumulation while maintaining compatibility with real-world sensor characteristics. Thus, under the given proposition's assumptions, the position error distribution is approximated as:

$$\epsilon_s(t) \sim \mathcal{N}\left(\frac{1}{2}\mu t^2, \frac{1}{2}\sigma^2 t^2\right) \quad (7)$$

□

### 3 Hardware Design

All hardware in our system is designed and implemented based on commercially available, off-the-shelf components. To this end, we developed a wearable sensing system consisting of six wireless sensor nodes, categorized into two functional types. The first type includes distal leaf nodes, which are mounted on the wrists and ankles, each integrating a 9-axis IMU and a time-of-flight (ToF) depth sensor. The second type comprises proximal relay nodes, which are positioned on the head and pelvis and are equipped with IMUs only. ToF sensors are intentionally omitted from the head and pelvis nodes because these locations seldom provide useful depth data, either from other body parts or from the ground. Instead, we adopt a 2-to-1 communication-relay architecture in which the head node receives and processes data from both wrist-mounted leaf nodes, while the pelvis node handles data from both ankle-mounted nodes. Each relay node performs lightweight local preprocessing, integrates its own sensor data, and transmits the aggregated information to a host machine using the ESP-NOW wireless protocol. This hierarchical topology mitigates 2.4 GHz signal occlusion caused by the human body and enables reliable data transmission at a synchronized frame rate of 60 Hz. All sensor nodes operate in real-time and maintain temporal alignment across the network.

Practical power measurements were conducted on a complete sensing node comprising an MCU, IMU, ToF sensor, and wireless transmission module, powered by a 1000 mAh 3.7 V Li-ion battery. Total

system consumption during wireless transmission and continuous ToF operation was 0.97–1.00 W, yielding a measured runtime of approximately 4.38 hours per charge. These results indicate that the hardware achieves a balanced trade-off between sensing fidelity and energy efficiency, maintaining practical battery life under continuous sparse ToF operation.

## 4 Implementation Details

### 4.1 ToF Sim2Real

Virtual cameras are placed at fixed vertices on an SMPL human mesh, with all shape parameters set to zero. These cameras are configured to match the physical properties of real ToF sensors, including a 45-degree horizontal and vertical field of view, a near plane of 0.01 m, and a far plane of 2 m. The orientation of each virtual camera is aligned with the local surface geometry of the body mesh. Specifically, the optical axis is set to the normalized normal vector of a triangle formed by the selected vertex and two of its immediate neighbors. To construct a full orthonormal camera frame, an upward reference direction is first defined based on anatomical context: for wrist-mounted cameras, it points from the wrist to the fingertips, while for ankle-mounted cameras, it points from the ankle to the heel. The rightward direction is then computed as the normalized cross product between the upward vector and the optical axis, and the upward direction is subsequently redefined to ensure orthogonality. These three orthogonal axes—right, up, and forward—form the camera’s rotation matrix used for rendering. During the geometric stage of Unity’s rendering pipeline, the skinned human model vertices are transformed into the homogeneous coordinate system of the above-defined camera space  $(x, y, z, w)$ . The value  $z/w$  is stored in an 8-bit grayscale depth texture (values ranging from 0 to 255) with a resolution of  $300 \times 300$ . Each frame of motion is rendered into its corresponding depth texture and preprocessed. Specifically, bilinear interpolation is applied to downsample the high-resolution distance maps to match the resolution of actual ToF sensors.

During the geometric stage of Unity’s rendering pipeline, the skinned human model vertices are transformed into the homogeneous coordinate system of the above-defined camera space, i.e.,

$$\begin{bmatrix} x' \\ y' \\ z' \\ w \end{bmatrix} = \mathbf{P} \cdot \mathbf{V} \cdot \mathbf{M} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

where  $\mathbf{M}$ ,  $\mathbf{V}$ , and  $\mathbf{P}$  denote the model, view, and projection matrices, respectively. The normalized device coordinate (NDC) depth value is obtained by computing  $z/w$ , which is then stored in an 8-bit grayscale depth texture (with values ranging from 0 to 255) and a spatial resolution of  $300 \times 300$ . Each frame of motion is rendered into its corresponding depth texture and preprocessed. Specifically, bilinear interpolation is applied to downsample the high-resolution distance maps to match the resolution of actual ToF sensors.

### 4.2 Node-centric Integration (NCI) Module

The NCI Module includes: 1) Linear Tokenize Layer that mapping each sensing node data to a 64-dim token; 2) One-stack Transformer Encoder with  $d_{\text{model}} = 64$  and FFN size= 128; 3) Linear Output Layer that mapping each output token to a 28-dim vector.

### 4.3 Position Estimation Function $f_{\phi}^n$

We implement  $f_{\phi}^n$  via a 2-layer MLP with 128 hidden size and LeakyReLU activation. The output of MLP are scaled with Tanh and  $\pi$  as follow:

$$f_{\phi}^n(\cdot) = \pi \cdot \text{Tanh}(\text{MLP}(\cdot)) \quad (8)$$

This design ensure a numerical constrains of  $\phi_n \in [-\pi, \pi]$ .

### 4.4 Motion Estimators

Each of motion estimator  $f_v$ ,  $f_p$ , and  $f_{\phi}$  are implement via 2-stack LSTM with 256 hidden size and zero dropout. The  $f_v$ ,  $f_p$  provide 3D volocity and position of all 24 SMPL joints, and  $f_{\phi}$  provide 6D rotation of 18 SMPL joints without hands and foots.

## 4.5 Runtime and Power Efficiency

The exported ONNX model on an Intel i7-13700K CPU achieved an average latency of  $0.65 \pm 0.69$  ms per frame ( $311.73 \pm 78.34$  FPS) over 18,000 frames. To ensure temporal consistency with time-differentiated acceleration, inference was capped at 60 FPS, under which latency remained  $1.10 \pm 1.05$  ms and the measured frame rate was  $60.44 \pm 4.57$  FPS. These results confirm real-time performance on a standard CPU without a dedicated GPU.

## References

- [1] Rayan Armani, Changlin Qian, Jiayi Jiang, and Christian Holz. Ultra inertial poser: Scalable motion capture and tracking from sparse inertial sensors and ultra-wideband ranging. In *SIGGRAPH 2024 Conference Papers*. Association for Computing Machinery, 2024.
- [2] Robert Grover Brown Brown and Patrick YC Hwang. *Introduction to random signals and applied Kalman filtering: with MATLAB exercises fourth ed.* Wiley & Sons, 2012.
- [3] Mohinder S Grewal and Angus P Andrews. *Kalman filtering: Theory and Practice with MATLAB*. John Wiley & Sons, 2014.
- [4] Yinghao Huang, Manuel Kaufmann, Emre Aksan, Michael J Black, Otmar Hilliges, and Gerard Pons-Moll. Deep inertial poser: Learning to reconstruct human pose from sparse inertial measurements in real time. *ACM Transactions on Graphics (TOG)*, 37(6):1–15, 2018.
- [5] STMicroelectronics. *VL53L8CX Time-of-Flight Ranging Sensor Datasheet*. STMicroelectronics, 2024. URL <https://www.st.com/en/imaging-and-photonics-solutions/vl53l8cx.html>. DS14161, Rev. 11, October 2024.