Self-Calibrated Multi-Sensor Wearable for Hand Tracking and Modeling

Nikhil Gosala*, Fangjinhua Wang*, Zhaopeng Cui, Hanxue Liang,
Oliver Glauser, Shihao Wu, Olga Sorkine-Hornung

Abstract—We present a multi-sensor system for consistent 3D hand pose tracking and modeling that leverages the advantages of both wearable and optical sensors. Specifically, we employ a stretch-sensing soft glove and three IMUs in combination with an RGB-D camera. Different sensor modalities are fused based on the availability and confidence estimation, enabling seamless hand tracking in challenging environments with partial or even complete occlusion. To maximize the accuracy while maintaining high ease-of-use, we propose an automated user calibration that uses the RGB-D camera data to refine both the glove mapping model and the multi-IMU system parameters. Extensive experiments show that our setup outperforms the wearable-only approaches when the hand is in the field-of-view and outplays the camera-only methods when the hand is occluded.

Index Terms—hand tracking, wearable sensors.

1 Introduction

Hand modeling and tracking has emerged as an important and highly researched problem in the field of computer graphics with numerous applications in human-computer interaction, ergonomics, bio-mechanics, and mixed reality. Successful application in these domains demands high levels of accuracy in the estimated hand pose, the potential to run in real-time, and the ability to work in a wide range of environments.

Existing hand pose estimation and tracking solutions largely make use of either vision-based or wearable sensors. Vision-based approaches estimate the shape and pose of the hand by using either multiple cameras [1], RGB-D sensors [2], or monocular cameras [3]. However, these methods require having the hand in the field-of-view (FoV) of the camera, and are sensitive to motion blur and poor lighting. Wearable device-based approaches, which leverage passive devices such as inertial measurement units (IMUs) [4], flex sensors [5] or stretch sensors [6, 7], can estimate the hand pose without requiring a controlled environment and a direct line-of-sight between the hand and the sensor. However, these passive devices are unaware of the extrinsic shape parameters and thus require careful and tedious calibration.

Research has been conducted towards fusing the pose outputs from both sensor modalities to overcome the drawbacks of each individual system. Fusion of vision and wearable sensors has been performed in [8] to estimate the position of fingertips; however, this approach does not directly capture the pose of all hand joints, and relies on manual calibration. The approaches in [9] and [10] address the challenge of 3D hand tracking using manually pre-calibrated bone lengths, the estimation of which is highly involved and prone to inaccuracies.

In this paper, we present a novel multi-sensor hand tracking and modeling system based on an RGB-D camera, a set of IMUs and a stretch-sensing glove [7] that leverages the full advantages of both vision-based and wearable sensors for seamless hand tracking and modeling in all conditions. In particular, we exploit both the depth camera and the stretch-sensing glove for consistent local 3D hand modeling. In addition, we design a simplified skeleton model based on only three IMUs, which is further integrated with the observations of the depth camera for robust and accurate global hand tracking.

We employ different existing techniques as building blocks of our system, while our key novelty lies in how we handle the uncertainties of the estimates from the employed modalities. We propose a novel α-weighted approach and exploit the extended Kalman filter for local hand modeling and global tracking, respectively, according to their properties. Our system, including the glove mapping model and the skeleton model parameters, can be automatically calibrated online, which greatly facilitates the use of our setup compared to existing systems [7, 9, 10].

Our multi-sensor setup and the fusion algorithm are integrated with ROS [11] and Unity [12], and are experimentally validated. The proposed automatic body skeleton calibration approach outperforms its manual counterpart and results in superior 3D global pose estimates. Further, the proposed local model fusion algorithm successfully estimates the confidence values of all finger joints and results in seamless transitions between the depth and glove poses during occlusion, as shown in Fig. 1. Qualitative evaluations show that our multi-sensor setup outperforms existing single sensor-based approaches in terms of dealing with both occluded and non-occluded cases using a unified system. Quantitatively, the global hand position is approx. 7% and 11% more accurate than the camera-based and IMU-based methods, respectively. The hand pose is approx. 11% and 15% more accurate than the camera-only and the glove-only setups, respectively.

2 Related Work

The literature on 3D hand modeling and tracking is vast. We briefly discuss the methods based on either vision-based

* *The two authors contributed equally to this paper.*

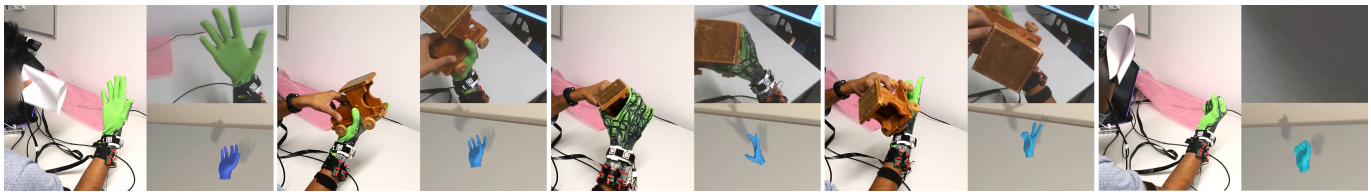


Fig. 1: Our multi-sensor wearable successfully performs hand tracking and modeling with high accuracy. Our approach automatically assesses whether the hand is visible to the camera (dark blue) or occluded (light blue), and combines the advantages of both optical and wearable sensors accordingly.

systems or wearable devices alone, and then review existing works that deal with multi-sensor fusion in the context of hand pose estimation.

Vision-based methods

The methods using vision-based sensors can be classified into optimization-based and learning-based approaches. Optimization-based methods explicitly define a hand model and estimate the hand shape and pose parameters by minimizing the disparity between the depth information and the reconstructed hand [13, 14, 15, 16, 17]. To guarantee generalization of the manually designed model over a wide range of hand shapes, the works in [18, 19, 20] introduce offline calibration procedures, wherein the user has to replicate a set of predefined hand poses, from which the hand model is personalized. Tkach et al. [21] propose online calibration to make the process more intuitive and user-friendly.

Under the broad umbrella of deep neural networks, learning-based approaches have gained a lot of attention [22]. Tompson et al. [23] employ a convolutional neural network (CNN) to generate a probability distribution function for each joint in the form of a heat map, from which the 2D positions of the joints are inferred and propagated to 3D. Oberwerger et al. [24, 25] directly regress 3D joint locations on the input depth image. To address the highly non-linear nature of the 2D-to-3D mapping, Ge et al. [26] convert the input 2D depth images into a 3D point cloud, Moon et al. [27] employ volumetric CNNs, and Baek et al. [28] learn a one-to-one mapping between the depth input and the corresponding 3D hand pose skeleton through generative adversarial networks that enforce the cyclic consistency constraint (CycleGANs). However, these approaches are parameter intensive, which makes them computationally very inefficient [29]. This is addressed by either splitting the pose estimation process into multiple stages [30] or reformulating it as a dense regression problem [31]. The latter approach currently records top performance among learning-based approaches , which we adopt as the backbone of our depth-to-pose model.

To address the situations when the hand is occluded, Mueller et al. [32] train a network to estimate the correspondences and segmentation for the input depth image. Zhou et al. [33] jointly use 2D and 3D annotated real or synthetic image data as well as motion capture data to improve the network performance under occlusion. However, these methods require a large amount of training data and can only handle partial occlusions; we resort to additional wearable devices for better handling of occlusion.

Wearable-based methods

Wearable sensor setups for hand pose estimation are usually constructed in the form of a data glove, where the sensors are either embedded into the glove [6, 34, 35] or attached to its surface [4, 7]. Wearable data gloves employ a variety of sensors such as inertial measurement units (IMUs), flex sensors, magnetic sensors and stretch sensors [36].

IMUs are electronic devices that measure the linear and angular acceleration of objects using a combination of accelerometers and gyroscopes. IMUs have been widely used in human motion capture [37, 38, 39, 40]. However, IMU sensors are often bulky and produce noisy measurements, requiring time-consuming calibration techniques [36, 7]. Being part of the dead-reckoning sensor family, the commercially available and inexpensive IMUs quickly accumulate drift, making them re-calibration demanding.

Flex sensors, also known as flexion, bend, or angular displacement sensors, are used to measure the change in bend, which is subsequently used to compute the angle assumed by a joint [5]. Flex sensors have gained a lot of popularity in commercial products, such as the CyberGlove [41], ManusVR glove [42] and 5DT glove [43]. However, it is difficult to layout a large number of flex sensors in a small area such as human hands.

Stretch sensors, also known as strain sensors, are used to measure the amount of stretch and bend in the object [44, 45]. Resistance-based stretch sensors have been adopted in hand motion capture [35, 46, 34, 47]. The MoCap Pro Glove from StretchSense [48] contains 16 sensing channels and can capture hand motion for an inexperienced user. However, we can not fine-tune, or domain transfers their commercialized regression model using new training data obtained by a vision-based approach. Glauser et al. [7] develop a data-driven glove based on dense stretch sensor arrays to estimate the pose of the hand in real-time [49]. Their glove is thin, soft, low-cost, and accurate; however, it cannot capture the 3D position and the shape of the hand, and requires min-max calibration, which can be unreliable. We address these limitations by making use of the vision-based observations whenever available.

Multi-sensor fusion

Various works investigate the fusion of estimates from cameras and IMUs for improved motion capture [50, 51, 52, 53, 54, 55], and for refining the calibration of wearable sensors using their vision-based counterparts [56], which is helpful when some of the sensors need constant re-calibration due to them being dead-reckoning and noisy [57, 58]. Existing hybrid tracking methods for human poses typically jointly optimize

the pose parameters to fit the observations from different sensors [59], or use weight averaging with a predefined weight [60], while our method considers the uncertainty of the camera observation via Extended Kalman Filter or adaptive alpha-weighting, and is therefore more robust to occlusion. Recently, research has been conducted towards estimating the position of fingertips using a LeapMotion sensor [61] and a flex-sensing glove [8]. However, this approach does not actively estimate the pose of all joints and also does not use any online calibration, which forces the user to either undergo a lengthy calibration procedure or stick to a generic model, which affects the accuracy of the estimate.

We emphasize the flexibility of our sensor fusion setup: it does not require capturing new training data whenever a new modality, e.g., a better vision-to-pose [62] or glove-to-pose [63] model, becomes available.

3 Overview

In this paper, we propose a multi-sensor setup for hand modeling and tracking, as shown in Fig. 2. The setup integrates sensor readings from a stretch-sensing soft glove [7], IMU readings and observations from the RGB-D camera, and outputs the 3D hand shape and poses including the orientation of each joint of the hand, and the orientation and position of the hand in 3D space. Our setup consists of two symbiotic main modules – *local hand modeling* and *global hand tracking*.

The local hand modeling module, detailed in Sec. 4, fuses the hand pose estimates from the depth camera and the stretch-sensing glove, using depth-to-pose and sensor-to-pose models adapted from [31] and [7], respectively. The fusion is performed in an α-weighted fashion, where the weight, α, is estimated from heatmaps – an intermediate result from the depth-to-pose network. Additionally, we exploit the multi-modal nature of our system and use hand poses estimated from the depth camera to implicitly personalise the sensor-to-pose mapping model to each user.

The global hand tracking module, described in Sec. 5, is based on a simplified skeleton model. It fuses the global pose estimates from the vision as well as the multi-IMU system and the calibrated skeleton model with known lengths for body segments to improve hand tracking accuracy. Furthermore, instead of a much more involved manual calibration, which is prone to measurement errors, we propose an online calibration that uses the visual 3D tracking output to automatically estimate the lengths of the body segments.

4 Local hand modeling

In order to capture the local hand pose and geometry consistently, we propose to utilize multiple types of sensors, i.e., the depth camera and the stretch-sensing glove. This allows to handle the shortcomings of each modality and estimate the hand poses robustly and accurately in all conditions. As shown in Fig. 2, two models, i.e., depth-to-pose and sensor-to-pose, are first utilized to generate individual hand poses given the depth image and sensor readings, which are then fused together to generate the final hand shape and pose. In order to fuse the output from different modalities, we design a confidence estimation for hand pose capture from visual information in Section 4.1. Moreover, we introduce a calibration method for the sensor-to-pose model of the stretch-sensing wearable sensors using the RGB-D information in Section 4.2. At last, we fuse the 3D hand pose from different sensors with a simplified hand model adaptation.

4.1 Heatmap-based confidence estimation

For the hand pose estimation from depth, we adopt the denseReg network [31] due to its accuracy and efficiency. Given the input depth image, the pixel-wise 2D/3D joint heat maps will be first estimated for each joint, and aggregated with the clustering algorithm into a global estimate. We further exploit the estimated 2D heat maps to estimate the confidence of the estimated 3D hand pose.

The first-row of Fig. 3 shows examples of 2D heatmaps obtained from the depth-based network arranged in the decreasing order of their confidence. It can be observed that a joint regression is reliable when the likelihood-spread in its corresponding heatmap is small and well-defined, whereas the regression is ambiguous when its likelihood spread is large and has multiple maxima. We capitalize upon such an observation and subsequently propose a feature-based approach that uses traditional image processing techniques coupled with the domain expertise to compute the confidence of the generated heatmaps. Several features are proposed in our method to capture the confidence of a given heatmap H as below.

Background color

The background colour provides a high-level estimate of the likelihood-spread in the heatmap. This metric, denoted by f_{bbg}, computes the natural logarithm of the mean of the heatmap pixel values (re-scaled to $[0, 65535]$), i.e.,

$$f_{bbg} = \log\left(\frac{1}{|H|}\sum_{(x,y)\in H} I(x,y)\right), \tag{1}$$

where $|H|$ denotes the number of pixels in the heatmap and $I(x,y)$ represents the intensity of the heatmap at coordinate (x,y).

Cluster count

The number of clusters, n_c gives an estimate of the number of regions where the joint could have been regressed. Since each heatmap corresponds to only one joint, the confidence of a heatmap decreases with an increase in the number of clusters. The heatmap is clustered using a density-based clustering algorithm DBSCAN [64] that groups densely packed points together and marks points in low-density regions as outliers.

Cluster variance

The spread of a cluster is quantified by computing the sum of the principal diagonal entries of its covariance matrix:

$$f_{cov} = \sum_{i=1}^{n_c}\left(\sigma_i^{(0,0)} + \sigma_i^{(1,1)}\right), \tag{2}$$

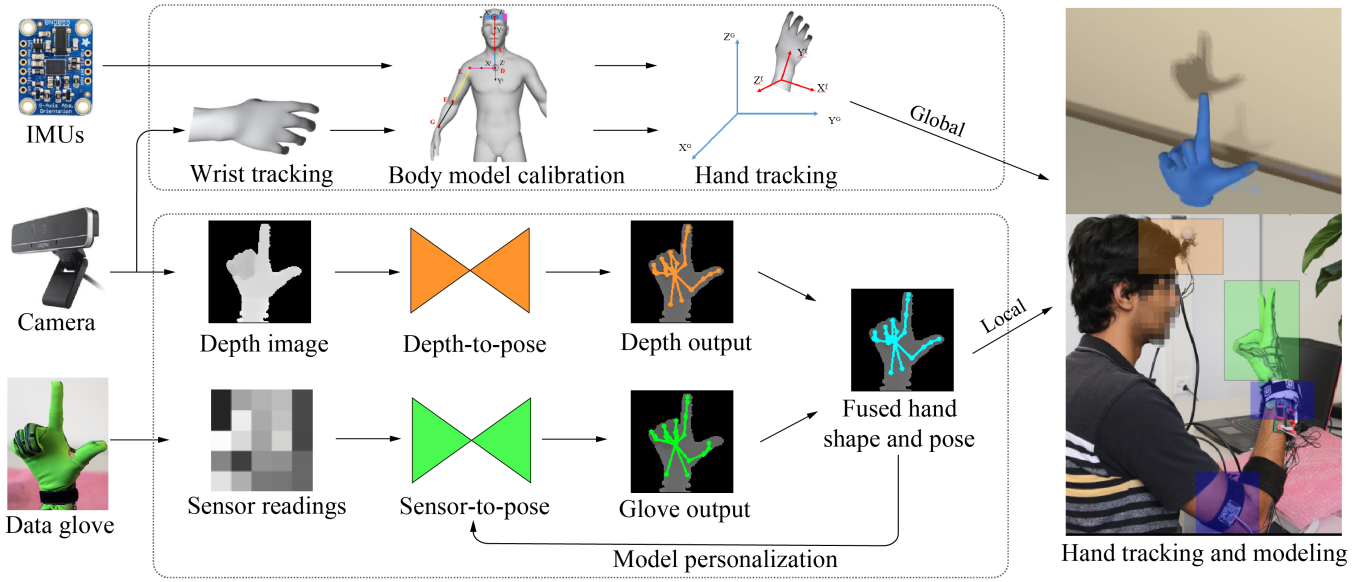


Fig. 2: Our multi-sensor device consists of a depth camera, a set of arm-tracking inertial measurement units (IMUs), and a hand-tracking soft glove. Our system includes global hand tracking and local hand modeling modules. When the hand is well-positioned in front of the camera, we combine the arm and hand pose estimations of multiple sensors and use the computed shape information to optimize our wearable modules. When the hand is moving out of FOV or being occluded, our optimized wearable seamlessly takes over the motion capture tasks until the camera detects the hand again.

where f_{cov} denotes the cluster covariance, n_c represents the number of clusters in the heatmap, and $\sigma_i^{(l,m)}$ represents the value in position (l, m) of the covariance matrix of the i^{th} cluster.

Local maxima

Local maxima represent areas with local 2D peaks. Subsequently, the distribution of local maxima in conjunction with their count gives an estimate of the confidence of a heatmap. We observe that the heatmap confidence decreases with an increase in the number of local maxima. This is due to the presence of multiple plausible locations where the final joint can be regressed, which makes selecting the correct location challenging. Further, we note that for heatmaps having the same number of local maxima, the intensity at each maximum also determines their confidence. For instance, the confidence of a heatmap having multiple peaks with similar intensities is lower than that of a heatmap where one peak significantly outweighs the others. This is because in the former case there is an equal chance for the joint to be regressed at any of the similarly weighted peaks, which reduces the probability of the joint being regressed at the correct peak. In contrast, in the latter scenario, the probability of the joint being regressed at correct high-intensity peak is much larger than at the other local maxima due to the significant difference in their peak intensities. The *weighted local maxima count* (f_{lm}) is thus formulated to account for both these observations. The base of the exponent comprises the local maxima count, which determines the confidence of a heatmap. The exponent term weights the different local maxima intensity distributions and ranks the different heatmaps having the same number of local maxima.

Specifically,

$$\rho = 10^{m-1} \cdot \prod_{i=1}^{m} \frac{I(x_i, y_i)}{\sum_{j=1}^{m} I(x_j, y_j)}, \tag{3}$$

$$f_{lm} = m^{1+\rho}, \tag{4}$$

where m is the number of local maxima in the heatmap and $I(x_i, y_i)$ represents the intensity at the i^{th} local maximum. Additionally, ρ measures the combined weight of all local maxima, scaled by a factor of 10^{m-1} to ensure that the product of weights does not become too small when multiple local maxima exist.

Furthermore, it is observed that the distribution of local maxima locations determines the confidence of the heatmaps. For instance, a heatmap with multiple local maxima in close proximity is more confident than a heatmap where the same number of local maxima are spread throughout the image. This observation is accounted for by defining *average local maxima distance* (f_d), which is defined as

$$f_d = \begin{cases} \log\left(\frac{1}{m}\sum_{i=1}^{m}\sum_{j=i+1}^{m}\|p_i - p_j\|_2\right), & \text{if } m > 1 \\ 1, & \text{otherwise,} \end{cases} \tag{5}$$

where m is the number of local maxima in the heatmap and p_i is the position of the i^{th} local maximum.

Considering all the above-defined metrics, the ambiguity of a heatmap, κ, is computed using

$$\kappa = f_{bbg} \cdot n_c \cdot f_{cov} \cdot f_{lm} \cdot f_d, \tag{6}$$

where $\kappa \in (0, \infty)$. The value κ is then normalised to $[0, 1]$ to compute the confidence α, which is in turn used in the pose fusion algorithm:

$$\alpha = \begin{cases} \frac{1}{\sqrt{\frac{\kappa}{k}}}, & \text{if } \kappa > k \\ 1, & \text{otherwise,} \end{cases} \tag{7}$$

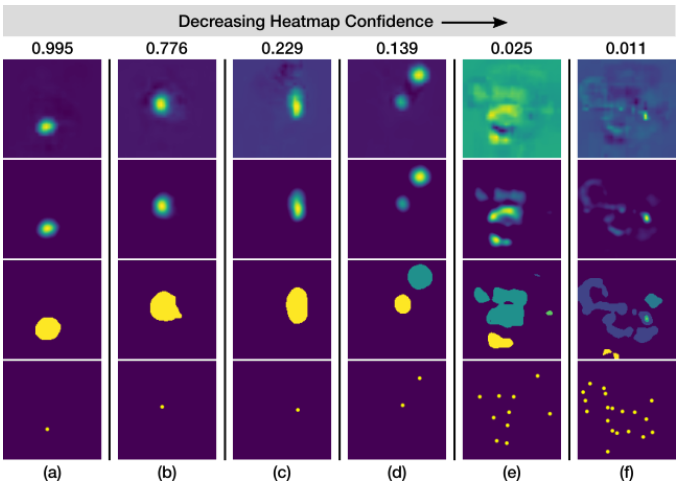


Fig. 3: Heatmaps along with the intermediate stages obtained when computing their confidence. The heatmaps are arranged in the decreasing order of their confidence, and the estimated confidence is recorded above each heatmap. The first row shows the raw input heatmap, which is directly used to compute the background color. The second row shows the filtered heatmap, which is obtained by repeatedly blurring and re-scaling the input. The third row shows the output when the DBSCAN algorithm is applied on the filtered heatmap. The output from DBSCAN is used to compute the cluster count and cluster covariance. The filtered heatmap is used to compute the local maxima map, shown in the last row, wherein the yellow circles denote the positions of the local maxima, used to compute the weighted local maxima count and the average local maxima distance.

where k is a threshold set to 1000 in our implementation, and α is the final, $[0, 1]$-bounded confidence estimate of the heatmap.

The uncertainty of each regressed joint is subsequently used to compute the confidence of each finger and the hand as a whole. Empirically, a finger in the depth pose is classified as ambiguous when at least 2 of the 4 finger joints have confidence less than 0.4, or when the mean confidence of all joints in the finger is less than 0.6. Similarly, the entire hand is said to be ambiguous when more than 4 fingers are classified as ambiguous or when more than 14 of the 23 joints in the depth pose have confidence less than the pre-defined threshold of 0.4. When a finger is classified as ambiguous, the depth pose estimate for that finger is discarded, and the final pose of the finger is assumed to be the pose obtained from the glove. Likewise, when the entire hand is ambiguous, the depth pose estimate is completely discarded, and the pose estimate from the glove is assumed to be the final pose of the hand.

4.2 Sensor-to-pose model calibration

To compute the hand pose from the stretch-sensing glove, we adopt the sensor-to-pose model similar to [7]. The input of the model is the mapped stretch sensor readings and the output is the hand pose parameters. The stretch sensors require calibration as users have a large variety of sizes and shapes of hands. In order to handle this problem, we propose a new auto-calibration method based on the visual information for the sensor-to-pose model, which includes i) estimating a personalized hand skeleton model, e.g., bone lengths; ii) domain transfer for the stretch-to-pose network. Note that we consider the unknown hand shape of a new user as the domain gap for the pre-trained base network. Fine-tuning such a model using the available camera information can be viewed as calibration.

First, from the depth-to-pose estimation, we can easily get the bone lengths by computing the average over multiple observations with a high confidence estimate. Secondly, we can use the estimated pose [31] with high confidence for domain transfer of the stretch-to-pose network. To promote fast training, we replace the UNet-based network proposed in [7] with a significantly efficient architecture consisting of only 2 convolutional and 4 fully-connected layers, as shown in Fig. 4(a).

While the conventional fine-tuning method can improve the performance of a base model on an unseen hand, we instead use side-tuning as inspired from [65]. Model calibration using side-tuning involves the training of an additional network from scratch whose output is concatenated to the output of the generic network to generate the final personalized pose estimate, as shown in Fig. 4. To this end, our model-personalization network contains three networks, namely, (1) the base network from Fig. 4(a), (2) a side network containing 1 convolution and 2 fully-connected layers to account for the errors caused by the base network, and (3) a post-processing network consisting of 2 convolutional layers to merge the outputs from the aforementioned networks. During training, the parameters of the base network are fixed, and the side network along with the post-processing network learn to encode the characteristics pertaining to the new user. In our experiments, we train our new base model until convergence using 265,204 training and 33,298 validation samples. We then adapt the model to an individual user using $\sim$500 training and $\sim$8,700 validation samples obtained using high-confidence predictions from our depth-to-pose network.

4.3 Multi-sensor fusion for hand modeling

As shown in Fig. 2, the pose estimates from depth and stretch-sensing glove are finally fused to generate robust and accurate local hand shape and pose. One major challenge of multi-sensor fusion is the mismatch between the data representations of multiple sensor modality pipelines. In this section, we address this model mismatch challenge and fuse the pose estimates with the joint confidence estimated in Sec. 4.1.

4.3.1 Hand model alignment

The pose estimates from the depth and glove pipelines use different hand representations, as shown in Fig. 5, which makes direct fusion of their outputs challenging. We denote the hand skeleton representation used by the glove-based and depth-based models as $\mathcal{H}_G$ and $\mathcal{H}_D$, respectively. $\mathcal{H}_G$ uses the 39-centre model from [21] (Honline) and predicts the flexion, abduction and rotation angles of joints in the hand. $\mathcal{H}_D$ regresses 23 joints on the input depth map based on [66]'s NYU hand pose dataset. Furthermore, the $\mathcal{H}_D$

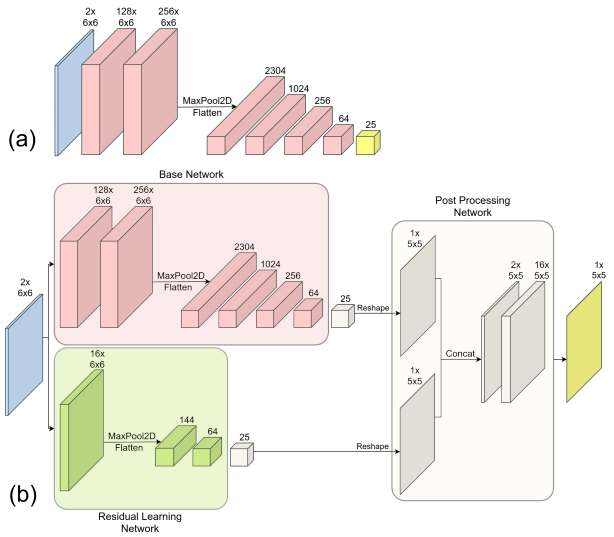


Fig. 4: The architectures used to (a) map the sensor readings from the wearable to the estimated hand pose, and (b) calibrate the sensor-to-pose mapping model using the side-tuning approach. In the figures, the input and output are depicted in blue and yellow respectively, while the base network is depicted in red, the side-tuning approach in green and the post-processing network is coloured in gray.

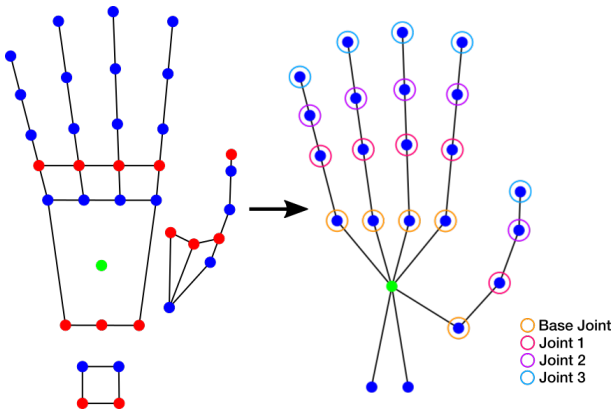


Fig. 5: The hand pose representation $\mathcal{H}_G$ used by the glove-based model [21] (left), and $\mathcal{H}_D$ used by the depth-based model [31] (right). We define a mapping converting $\mathcal{H}_G$ to $\mathcal{H}_D$ to enable the fusion of hand pose estimations. The retained, deleted and added nodes in $\mathcal{H}_G$ are color-coded in blue, red and green, respectively. We classify nodes based on their function. The nodes circled in orange, pink, violet, and blue are referred to as *"Base Joint"*, *"Joint 1"*, and *"Joint 2"*, *"Joint 3"*, respectively.

directly regresses the 3D positions of the joints, whereas the $\mathcal{H}_G$ works in the angle space to predict the angles between phalanges.

Ideally, the phalange angles in the depth pose should be the cosine inverse of the normalized dot product between the two adjacent phalange vectors. However, this approach cannot be employed in practice because the depth network independently predicts each 3D joint position without accounting for physical constraints such as the collinearity of joints in each finger. Such ambiguity makes the

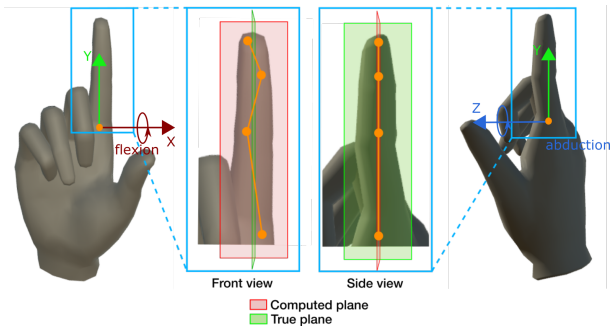


Fig. 6: Depiction of the chosen plane along which the angle between phalanges is computed by the dot product when the collinearity constraint of the regressed finger-joint is violated. The plane obtained by simple dot-product is shown in red, and the true plane along which the angle should be computed is shown in green. We also illustrate the XYZ axes of one joint.

computation of angles from 3D joints non-trivial. To avoid such an ambiguity, we need to find the true planes for angle computation, as shown in Fig. 6. First, we rigidly align the $\mathcal{H}_D$ with the $\mathcal{H}_G$ using seven nodes, including the base joints of each of the five fingers and the two wrist nodes.

After the affine transformation, we compute the merged base joints (orange circle in Fig. 5) using the joint confidence estimates,

$$c_{\text{merged}} = \alpha \cdot c_{\text{depth}} + (1 - \alpha) \cdot c_{\text{glove}}, \tag{8}$$

where c denotes the 3D position of the base joints, α is the corresponding heatmap confidence. The same merging principle is applied to joint angles fusion.

5 Global Hand Tracking

To track the global 3D hand motion in challenging scenarios, including highly occluded and complicated environments, we propose a sensor fusion algorithm that combines data from an RGB-D camera and multiple IMUs, as shown in Fig. 2. We use a simplified skeleton model for the upper human body, which allows our model to track the 3D hand motion using only three IMUs (BOSCH BNO055). At the same time, from the RGB-D camera, we can also track the 3D hand position but only as long as the hand is visible in the camera. To make the 3D tracking robust and continuous, we fuse both visual and wearable observations in an extended Kalman filter (EKF). For ease of use, we exploit the visual observations to automatically calibrate the body model instead of a tedious manual calibration as employed by [9] and [10].

5.1 Simplified skeleton model for upper body

Due to the noisy measurements of IMUs, Zhang et al. [67] model the human upper limb as a skeleton structure with two segments (upper arm and forearm) linked by a revolute joint (elbow joint). In addition to the upper limb, Peppoloni et al. [9] also include the clavicle, which connects the shoulder and thorax and present a novel 7 DoFs model that allows for the reconstruction of the human upper limb kinematics. Since

we utilize the information from RGB-D camera mounted on the head to target AR/VR applications, we also include the head as part of our model.

We model the human upper body as a simplified skeleton structure that consists of several joints and rigid segments as shown in Fig. 7. The head is modeled as a rigid segment connected to the trunk with a 3 DoF spherical joint at the neck. The upper arm is connected to the trunk with a 3 DoF spherical joint at the shoulder. The forearm is connected to the upper arm with a 1 DoF revolute joint at the elbow. The hand is connected to the forearm with a 3 DoF spherical joint at the wrist. Note that we focus on tracking the right hand in this paper, but the principle behind tracking two hands is the same.

Using the simplified skeleton model, we only need three IMUs to track the 3D hand. As shown in Fig. 7, we attach the first IMU to the RGB-D camera, the second at the lateral side of the upper arm, and the third at the lateral and flat side of the forearm near the wrist. Three coordinate systems are defined in our model as follows. (1) *Global coordinate frame* ***G***: the reference coordinate frame, defined by the direction of gravity and the magnetic north pole. (2) *Body coordinate frame* ***B***: this frame is attached to the body segment at the head (B^h), trunk (B^t), upper arm (B^u) and forearm (B^f) as depicted in Fig. 7. Currently, to simplify the system and use fewer sensors, we make the trunk to be static while the head and arm can be moved freely. This implies that B^t, the frame of trunk, is fixed during tracking and its pose w.r.t. the global coordinate frame G is measured offline. [1] (3) *Sensor coordinate frame* ***S***: for the IMUs, the accelerometer, gyroscope and magnetometer inside share the same sensor coordinate frame ***S***. S^h, S^u and S^f represent the senor coordinate frames for IMUs attached to the camera, the upper arm, and the forearm, respectively. For the RGB-D camera, the coordinate frame is denoted as S^c.

The IMUs are used to compute the relative rotation information between the body coordinate frame ***B*** and the global coordinate frame ***G***. If the length of each body segment is further known, we can capture the motion of each body segment as well as the hand in a given coordinate frame, *e.g.*, S^c.

5.2 3D hand tracking with RGB-D camera

When the hand is totally or even partially visible in the camera's view, we can track its 3D position in the camera coordinate frame. As discussed in Sec. 4.1, the denseReg network [31] is a state-of-the-art hand pose estimation method based on the depth image. To simplify the system, we directly reuse the hand pose estimation results from the denseReg network. Specifically, we take the average 3D position of two bottom joints, shown in Fig. 5 (b), as the position of the hand. To quantitatively describe the noise of observation, we utilize the confidence α in Eq. 7 of these two joints.

5.3 Body model auto-calibration

Different users have different body shapes. Therefore a body model calibration is required beforehand for the 3D hand tracking with IMUs. We propose an auto-calibration method for our simplified upper body model with the help of the RGB-D information.

1. Our system can be easily extended by attaching another IMU on the trunk to track the change of B^t, allowing the subject to move freely.

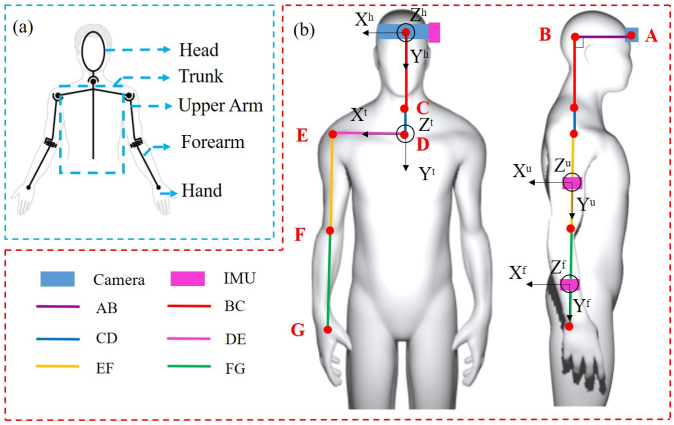


Fig. 7: (a) Simplified skeleton model for upper body. (b) Detailed skeleton model and layout of multiple sensors at N-pose (stand upright on a horizontal surface and arm straight alongside body vertically and thumbs forward). A represents the origin of the camera coordinate frame; C, E, G represent the 3-DoF spherical joint connecting the head and trunk, trunk and upper arm, and forearm and hand, respectively. F represents 1-DoF revolute joint connecting the forearm and upper arm.

5.3.1 Orientation calibration between body and sensor coordinate frame

First, we calibrate the orientation between body coordinate frames B^h, B^u, B^f and sensor coordinate frames S^h, S^u, S^f, *i.e.*, $R^{S^h}_{B^h}$, $R^{S^u}_{B^u}$ and $R^{S^f}_{B^f}$. The user is asked to keep in N-pose, which is shown in Fig 7, for a few seconds. The rotations from B^h, B^u, B^f to B^t, represented as $R^{B^t}_{B^h}$, $R^{B^t}_{B^u}$, $R^{B^t}_{B^f}$, can be calculated in the N-pose because of the alignment of coordinate axes as shown in Fig 7, *e.g.* $R^{B^t}_{B^h} = I$. With the absolute orientation $R^{S^h}_G$, $R^{S^u}_G$, $R^{S^f}_G$ from three IMUs, we get the calibration results as,

$$R^{S^h}_{B^h} = R^{S^h}_G R^G_{B^t} R^{B^t}_{B^h},\ R^{S^u}_{B^u} = R^{S^u}_G R^G_{B^t} R^{B^t}_{B^u},\ R^{S^f}_{B^f} = R^{S^f}_G R^G_{B^t} R^{B^t}_{B^f}, \tag{9}$$

As B^t is fixed in the current setup, $R^G_{B^t}$ can be easily measured offline with an IMU aligning its coordinate frame with B^t.

5.3.2 Body segment length calibration using Kalman filter

Manual measurement of body segments, such as AB, BC, CD depicted in Fig. 7, are difficult to measure and thus, prone to measurement errors. Therefore, we calibrate the body segment lengths automatically based on the information provided by the RGB-D camera and the IMUs. We employ a Kalman Filter with which we can estimate the lengths in real-time and update the current state recursively.

1) Process model: Let $l_1, l_2, l_3, l_4, l_5, l_6$ denote the lengths of *AB, BC, CD, DE, EF, FG*. We define the state vector x as

$$x = [l_1, l_2, l_3, l_4, l_5, l_6]^T. \tag{10}$$

In our model, the body segments are rigid and their lengths are constant. Therefore, the process model in Kalman filter can be given by

$$x_t = x_{t-1} + v_{t-1}, \tag{11}$$

where v_{t-1} is the process noise. Here, we assume v_{t-1} to be zero mean additional Gaussian white noise with covariance matrix Q.

2) Measurement model: The measurement model relates the measurement value z to the value of state vector x. The RGB-D camera provides an observation of hand's 3D position. Three IMUs provide absolute orientations w.r.t. the global coordinate frame ***G***, which are also included in the measurement model.

We use V^F to denote the vector V represented in the coordinate frame F. From Fig. 7, we can obtain the following equations,

$$\begin{aligned} AC^{B^h} &= AB^{B^h} + BC^{B^h} = \begin{bmatrix} 0 & l_2 & -l_1 \end{bmatrix}^T, \\ CE^{B^t} &= CD^{B^t} + DE^{B^t} = \begin{bmatrix} l_4 & l_3 & 0 \end{bmatrix}^T, \\ EF^{B^u} &= \begin{bmatrix} 0 & l_5 & 0 \end{bmatrix}^T, \qquad FG^{B^f} = \begin{bmatrix} 0 & l_6 & 0 \end{bmatrix}^T. \end{aligned} \tag{12}$$

Since the measurement is the 3D position of wrist in the camera's coordinate frame S^c, the measurement model is given by

$$\begin{aligned} z_t =& R_G^{S^c}(R_{B^h}^G AC^{B^h} + R_{B^t}^G CE^{B^t} + R_{B^u}^G EF^{B^u} + R_{B^f}^G FG^{B^f}) + \omega_t \\ =& R_{S^h}^{S^c} R_G^{S^h}(R_{S^h}^G R_{B^h}^{S^h} AC^{B^h} + R_{B^t}^G CE^{B^t} + R_{S^u}^G R_{B^u}^{S^u} EF^{B^u} \\ &+ R_{S^f}^G R_{B^f}^{S^f} FG^{B^f}) + \xi_t, \end{aligned} \tag{13}$$

where ξ_t is the measurement noise. We assume ξ_t to be zero mean additional Gaussian white noise with covariance matrix Σ. $R_{S^h}^{S^c}$ is the relative rotation between the sensor coordinate frame of IMU on head to the camera coordinate frame. Since the IMU is attached rigidly to the camera, it only needs to be calibrated offline once. Specifically, we print a grid of evenly spaced AprilTags [68] on a paper and attach an IMU, whose sensor coordinate frame is denoted as S^p. X^p, Y^p are aligned with the grid and Z^p is vertical to the paper. The 3D position of each tag can be easily read in S^p and detected in S^c. Then in S^p and S^c, we can get matrices A and B, where each column of the matrix is the 3D coordinate of each tag subtracted by the centroid of all tags. We use SVD to find rotation [69] $R_{S^c}^{S^p}$ between S^c and S^p as

$$[U, S, V] = SVD(B^T A) \qquad R_{S^c}^{S^p} = VU^T. \tag{14}$$

From two IMUs, we can get $R_{S^h}^G$ and $R_{S^p}^G$ respectively. So we get $R_{S^c}^{S^h}$ as

$$R_{S^c}^{S^h} = {R_{S^h}^G}^T \cdot R_{S^p}^G \cdot R_{S^c}^{S^p}. \tag{15}$$

To this end, we can see that z_t is a linear combination of x_t. Therefore, the measurement model can be written as,

$$z_t = Hx_t + \omega_t. \tag{16}$$

The model is only updated when the hand's 3D position is detected by RGB-D camera. During the calibration, we do not require the user to do any specific calibration movements, meaning that the user can move randomly as long as the hand is visible in camera's view. The calibration stops when a pre-defined number of iterations (1,000 by default) is reached.

5.4 Multiple sensor fusion for 3D hand tracking

Since we can track the 3D hand position with both the optical and the orientation sensors, we utilize all available information for the 3D hand tracking based on the multiple sensor fusion. For this part, as the system becomes non-linear, we apply an extended Kalman filter (EKF) to track the states in real-time.

1) Process model: For each IMU, we define a vector y which consists of the following three parameters:

$$y = [\bar{q}^T, \omega^T, b^T]^T, \tag{17}$$

where $\bar{q} = [q_0, q_1, q_2, q_3]^T$ represents the rotation from the global coordinate frame ***G*** to the sensor coordinate frames, $\omega = [\omega_x, \omega_y, \omega_z]^T$ represents the tri-axis instantaneous angular rates expressed in the sensor coordinate frame and $b = [b_x, b_y, b_z]^T$ represents the drift bias because the gyroscopes are subject to error terms such as noise and drift. We also define $p = [p_x, p_y, p_z]^T$ as the 3D position of hand expressed in the camera coordinate frame S^c. The final state vector x is given by

$$x = [p^T, {y^h}^T, {y^u}^T, {y^f}^T]^T, \tag{18}$$

where y^h, y^u and y^f represent the y vector for the IMUs attached to the camera, the upper arm and the forearm.

The general process model f is given by

$$x_t = f(x_{t-1}) + e_{t-1} = f(x_{t-1}) + [e_{t-1}^p, e_{t-1}^h, e_{t-1}^u, e_{t-1}^f]^T, \tag{19}$$

where e_{t-1} is the process noise. We assume e_{t-1} to be zero additional Gaussian white noise with covariance matrix Q. For e_{t-1}^h, e_{t-1}^u and e_{t-1}^f, there are three components including the quaternion noise e_{t-1}^q, the angular velocity noise e_{t-1}^ω, and the bias noise e_{t-1}^b.

The hand position p, similar to Equation 13, is given by the kinematic model of our skeleton structure as

$$\begin{aligned} p_t =& R_G^{S^c}(R_{B^h}^G AC^{B^h} + R_{B^t}^G CE^{B^t} + R_{B^u}^G EF^{B^u} + R_{B^f}^G EF^{B^f}) \\ &+ e_{t-1}^p. \end{aligned} \tag{20}$$

For each IMU, we use first order Runge-Kutta to update the quaternion as follows:

$$\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}_t = \left(I + \frac{\Delta t}{2} \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix} \right) \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}_{t-1} + e_{t-1}^q, \tag{21}$$

where Δt is the sampling time (about 0.015s in our implementation). Since the sampling time Δt is quite short, we model motion with constant angular velocity as

$$\omega_t = \omega_{t-1} + e_{t-1}^\omega. \tag{22}$$

The slow variation of the gyroscope bias is modeled as a first-order Markov Process driven by a white Gaussian noise e_{t-1}^b [67], i.e.,

$$b_t = b_{t-1} + e_{t-1}^b. \tag{23}$$

2) Measurement model: The RGB-D camera provides the observation of hand's 3D position. The three IMUs provide the following two types of measurement: 1) absolute orientation w.r.t. global coordinate frame ***G*** and 2) angular

velocity expressed in the sensor coordinate frame. The generalized form of the measurement model is given by

$$z_t = \begin{bmatrix} z_t^p \\ z_t^h \\ z_t^u \\ z_t^f \end{bmatrix} = h(x_t) + v_t = h(x_t) + \begin{bmatrix} v_t^p \\ v_t^h \\ v_t^u \\ v_t^f \end{bmatrix}, \qquad (24)$$

where z_t^p is the measurement of wrist position in camera coordinate frame, z_t^h, z_t^u and z_t^f are measurements for the IMUs attached to the camera, the upper arm and the forearm. v_t is the measurement noise which is assumed to be zero-mean additional Gaussian white noise with covariance matrix Σ.

The RGB-D camera measures the 3D hand position in the camera coordinate frame S^c, therefore, we have a simple model relating this measurement to the state as

$$z_t^p = p_t + v_t^p, \qquad (25)$$

where p_t is the vector for hand position in state vector x, expressed in the camera coordinate frame S^c.

For each IMU, we decompose z_t into z_t^q (measurement of quaternion) and z_t^ω (measurement of angular velocity). We decompose the noise v_t into v_t^q (noise of quaternion) and v_t^ω (noise of angular velocity) respectively. The absolute orientation output of an IMU is the rotation from the global coordinate frame to sensor coordinate frame and the angular velocity output of an IMU is expressed in the sensor coordinate frame. We write the measurements for each IMU as

$$z_t^q = \overline{q}_t + v_t^q, \qquad (26)$$

where $\overline{q}_t$ is the quaternion component for each IMU in the state vector x.

$$z_t^\omega = \omega_t + b_t + v_t^q, \qquad (27)$$

where ω_t and b_t are the angular velocity and the bias components for each IMU in the state vector x.

Our fusion method runs very efficiently and is thus able to fuse the visual and IMU information to estimate the global position and rotation of the hand in real-time.

6 RESULTS

Our multi-sensor device is shown in Fig. 2, and we synchronize all signals from different sensors using the timestamps recorded by the ROS framework. We evaluate our proposed multi-sensor setup in two steps. In the first part, we investigate the performance of the heatmap confidence estimate and evaluate the local hand pose fusion algorithm quantitatively and qualitatively. In a second part, we assess the global hand position tracking quantitatively with multi-sensor fusion.

6.1 Local pose fusion evaluation

We first assess the performance of the proposed heatmap confidence in Sec. 6.1.1 qualitatively, and then evaluate our pose fusion algorithm quantitatively and qualitatively in Sec. 6.1.2 and Sec. 6.1.3 respectively.

HM					
HMC	0.857	0.011	0.025	0.139	0.097

(a)

HM					
HMC	0.710	0.740	0.707	0.656	0.764

(b)

TABLE 1: Cases where our heatmap confidence (HMC) estimation (a) successfully estimates the confidence of different types of heatmaps (HM), and (b) produce unsatisfactory results, where high confidence estimates to otherwise ambiguous heatmaps, when the main cluster has artefacts such as comet-like tails around it.

	Heatmap	f_{bbg}	n_c	f_{cov}	f_{lm}	f_d	α
(a)		10.53	4.00	1028.45	12.31	5.62	0.018
(b)		8.97	1.00	482.367	8.11	2.85	0.100
(c)		8.40	1.00	204.43	1.0	1.0	0.763
(d)		8.19	1.00	145.21	1.0	1.0	0.917

TABLE 2: Breakdown of the heatmap confidence components. The higher the α value, the higher the confidence in the depth-to-pose model.

6.1.1 Heatmap confidence evaluation

We assess the performance of the proposed heatmap confidence estimation algorithm by qualitatively evaluating it using 10 samples that cover the spectrum of possible heatmaps. As shown in Table 1 (a), our proposed algorithm successfully detects both highly confident and highly ambiguous heatmap samples. Furthermore, our algorithm precisely determines the relative confidence of heatmaps. For instance, the last heatmap in Table 1 (a), which has a larger distance between its two clusters as compared to its fourth column counterpart, is correctly predicted to be less confident than the latter. However, as observed in Table 1 (b), our algorithm tends to falter in cases when there are multiple artefacts such as comet-like tails behind, and ring-like features in proximity to the main cluster. Such failure cases could be captured either by using a more comprehensive feature-set or by using a data-driven approach.

We also present a breakdown of the contribution of each component in our heatmap-based confidence estimation algorithm in Table 2. We observe that our algorithm generates coherent values for a wide variety of heatmaps encapsulating

the entire spectrum of heatmaps. Table 2(a) shows that our algorithm estimates a very low confidence value for those ambiguous heatmaps, while it estimates a high confidence value for those distinct heatmaps in Table 2(d). Examining the f_{bbg} values in the first column, we observe that the value is high when the background of the heatmap is predominantly white, which follows our design paradigm in Sec. 4.1. Similarly, f_{cov} also decreases with a decrease in the spread of the clusters, which can be noted by looking at the f_{cov} value of (d), which is nearly an order of magnitude smaller than that of (a). Further, (c) and (d) contain only one local maximum and thus have $f_{lm} = f_d = 1$, while (a), (b) exhibit high f_{lm} and f_d values due to the presence of multiple local maxima.

Additionally, we perform an ablation study to understand the utility of the various components of our heatmap-based confidence estimation function. Table 3 presents the results of this ablation study, where we compare the non-scaled κ and the scaled α values of the positive (confident) and negative (unconfident) heatmaps for the *middle_top* joint. We manually select and label 10 positive heatmaps and 10 negative heatmaps (of the same finger joint) as our ground truth data.

We evaluate five different versions and M5 is the full model we use. In model M1, we use only f_{bbg} to estimate the heatmap confidence. We observe that both the κ and α values are very close to each other, which makes differentiating between the confident and unconfident examples very challenging. Upon adding n_c in model M2, we observe that the difference between the confidence estimates increases by nearly 2 times. We account for the cluster-spread using the f_{cov} term in model M3, which results in a significant difference between the confident and unconfident heatmaps, with the κ value of unconfident heatmaps being nearly an order of magnitude larger than the confident heatmaps. At this stage, we also observe a significant difference between the α values, which helps us distinguish between the different kinds of heatmaps. In model M4 and M5 we introduce terms that account for both the number and the spread of local maxima in the heatmaps. These terms further improve the disparity between the positive and negative heatmaps, with the final κ value for negative samples being nearly three orders of magnitude larger than the positive samples. This large disparity allows for the easy differentiation between the positive and negative samples, improving the reliability of the heatmap confidence estimate.

6.1.2 Quantitative evaluation

To evaluate our pose fusion algorithm quantitatively, we compare the accuracy of the phalange angles estimated from the depth, glove and merged pose estimates with those from [21]. Since it is difficult to obtain the ground truth hand poses, we regard the output of [21] as our reference, which is an optimization-based method used to obtain the training data for the glove model and has the state-of-the-art accuracy when the hand is not occluded. Note that [21] needs some frames until convergence to obtain handshape parameters.

The results of three different runs are shown in Fig. 8. Note that the depth and merged pose estimates are able to accurately predict the phalange angles over various hand poses with very limited deviation from the reference.

Model	f_{bbg}	n_c	f_{cov}	f_{lm}	f_d	κ^-_{pred}	α^-_{pred}	κ^+_{pred}	α^+_{pred}
M1	✓	-	-	-	-	9.63	1.0	8.21	1.0
M2	✓	✓	-	-	-	19.75	1.0	8.21	1.0
M3	✓	✓	✓	-	-	15529.23	0.36	1389.78	0.85
M4	✓	✓	✓	✓	-	563446.29	0.08	1389.78	0.852
M5	✓	✓	✓	✓	✓	2536872.00	0.05	1389.78	0.852

TABLE 3: Ablation study highlighting the utility of various components in our heatmap confidence estimation algorithm. κ and α follow from their definitions in Eqs. (6) and (7). The $+$ and $-$ superscripts on κ and α denote positive (confident) and negative (unconfident) heatmaps respectively. α^-_{pred} and α^+_{pred} are the model prediction when the ground-truth should be 0 or 1, respectively Each row shows the estimate of a different model, M5 is the default model we use.

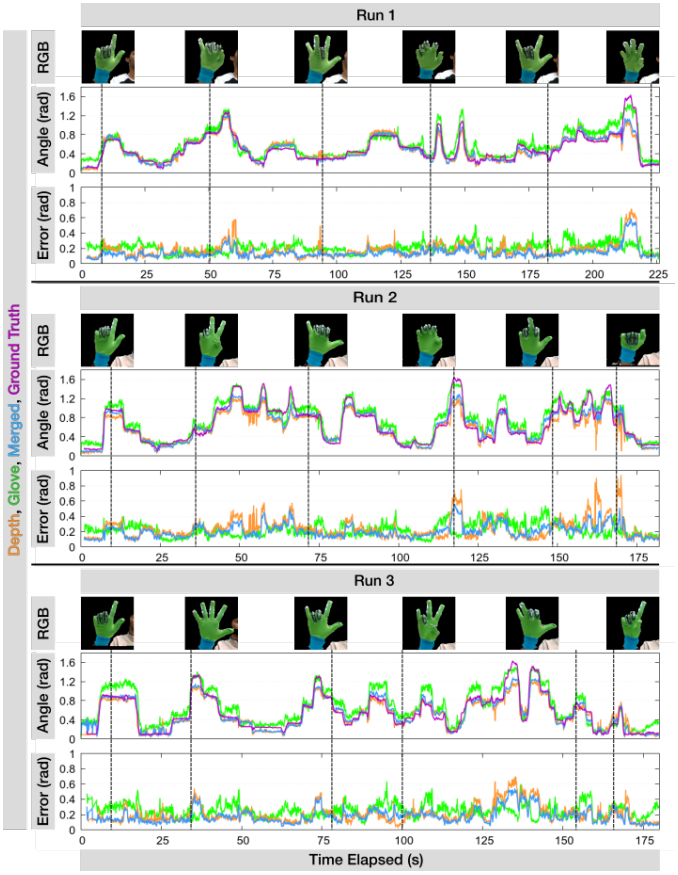


Fig. 8: Comparison of the angles (mean angle of all joints) estimated using the depth (orange), glove (green) and merged poses (blue) with those of the reference (purple) obtained from [21]. The angles from merged pose align with those from depth pose for most of hand poses because the hand is in the FOV for most input frames. However, when the depth pose becomes ambiguous and the heatmap confidence drops, the merged pose automatically gives more weight to the glove pose and continues to provide reliable hand pose estimates.

Run		Depth	Glove	Merged
1	$\boldsymbol{\mu}$ (rad)	0.170	0.209	0.142
	$\widetilde{\boldsymbol{x}}$ (rad)	0.112	0.144	0.098
	$\boldsymbol{\sigma^2}$ (rad^2)	0.045	0.039	0.026
2	$\boldsymbol{\mu}$ (rad)	0.237	0.222	0.206
	$\widetilde{\boldsymbol{x}}$ (rad)	0.165	0.159	0.157
	$\boldsymbol{\sigma^2}$ (rad^2)	0.069	0.040	0.039
3	$\boldsymbol{\mu}$ (rad)	0.200	0.249	0.182
	$\widetilde{\boldsymbol{x}}$ (rad)	0.132	0.186	0.125
	$\boldsymbol{\sigma^2}$ (rad^2)	0.055	0.050	0.037

TABLE 4: Mean (μ), median ($\widetilde{x}$), and variance (σ^2) of the absolute deviation of the mean joint angle predicted by the depth, glove and merged poses as compared to the reference angle from [21] for the three runs depicted in Fig. 8. 4000 unseen frames are used in this experiment. The mean deviation of the merged pose is significantly better than either the depth or the glove pose thus indicating an overall improvement in the accuracy of the pose estimate due to the use of our fusion pipeline.

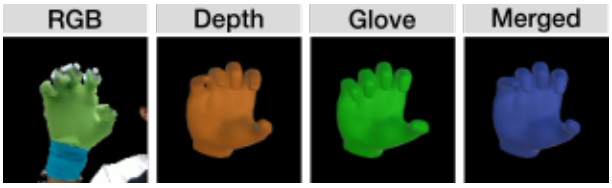


Fig. 9: A failure case. The depth, glove and merged pose estimates for the scenario at 213 s in Run 1 as shown in Fig. 8. The glove pose accurately estimates the pose of the hand, but since the phalange angle is computed incorrectly from the depth pose, which has a high confidence value, the merged pose biases towards it and outputs an incorrect pose estimate.

Furthermore, the phalange angles estimated from the merged pose are coherent with those computed from the depth for an extensive set of hand poses, and such a behaviour can be attributed to the high certainty of the regressed joints in the depth pose which makes the blending parameter α to be closer to 1, thus favouring the depth pose over the glove pose. However, in cases where the depth pose is ambiguous and contains a lot of error, for example when the hand is curled in the form of a fist at around 168 s in Run 2, the confidence in the depth pose decreases, and the merged pose gives more weight to the angle estimated by the glove.

However, although the angle predicted by the glove is more accurate than that of the depth, the merged angle prioritises the latter over the former, and such an inconsistency can be classified as a failure case, for instance at around 213 s in Run 1. Fig. 9 shows the depth, glove and merged poses for this specific case, from which it can be inferred that the angle at the *Pinky Top* joint is estimated incorrectly but confidently ($\alpha_\theta^{\text{pinky\_top}} = 0.769$) from the depth pose. Such a high confidence estimate causes the merged pose to bias towards the incorrect pose from the depth, resulting in the final merged pose being incorrect. This failure can be attributed to an error in the depth pose estimation network that confidently regresses the joint at the wrong location resulting in an incorrect angle estimate. This failure can be addressed by either (1) replacing the depth pose estimation network used in our experiments with the current state-of-the-art to reduce the frequency of such errors, or (2) using temperature-scaling to re-scale the heatmaps in a post-processing step that accounts for the uncertainty in the pose obtained from the depth pose estimation network.

Table 4 compares the mean, median and variance of the absolute deviation of the angles estimated by the depth, glove and merged poses from those of the reference from [21]. Note that the error in the merged pose estimate are significantly smaller than either the depth pose or the glove pose. Furthermore, the variance of the merged pose is relatively smaller than either the depth or glove pose which indicates that the merged pose estimate is not only less noisy but also more consistent with the reference. These observations thus show that the output from our pose fusion algorithm is superior to that from either sensor modality alone.

6.1.3 Qualitative evaluation

We qualitatively assess the performance of our algorithm by both projecting the merged pose estimate onto the input depth image to evaluate the coherence between them, and manipulating a 3D hand mesh to obtain a holistic view of the pose estimate in two real-world scenarios, namely, *unoccluded* and *occluded* scenes, are used for the evaluation. Using the former, we evaluate components of our fusion approach such as finger-straightening, whereas using the latter, we test the transitioning and pose-changing ability of our fusion algorithm when a foreign object actively occludes the hand. Note that to visualize the 3D hand model, we manually build the hand mesh and apply rigging by mapping the hand skeleton $\mathcal{H}_G$ to the one in Unity [12].

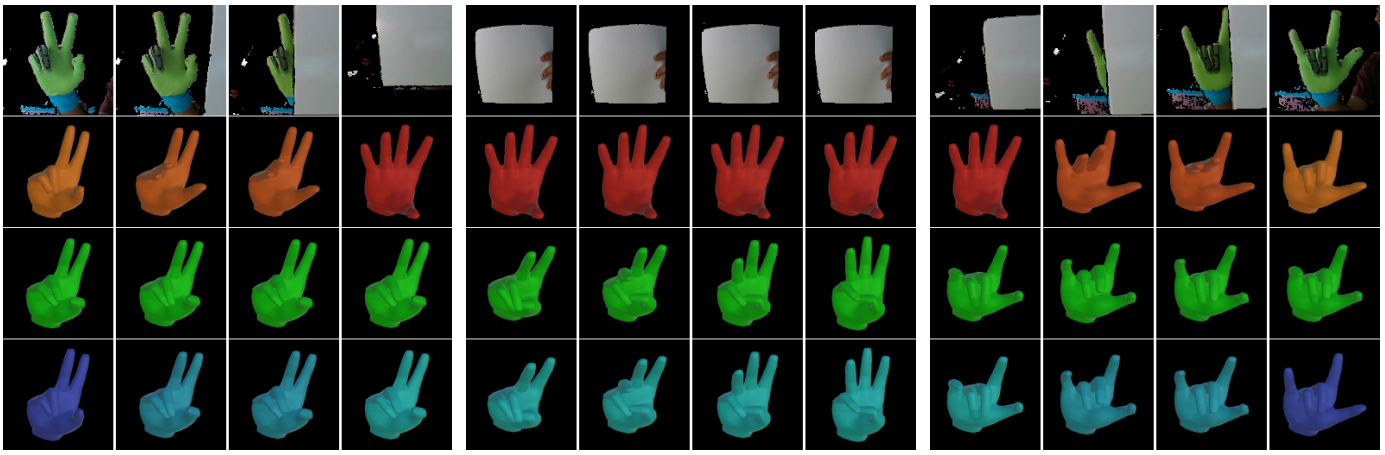
Scenario 1 - No Occlusion

Fig. 10 qualitatively evaluates the depth, glove and merged pose estimates by projecting the 3D hand skeleton on the input depth images using scenarios with self-occlusion, but no foreign body occlusion. It can be noted that the depth network from [31] is able to accurately estimate the pose of the hand even in cases where the fingers self-occlude some joints. Moreover, the finger-joint collinearity constraint is often violated in the depth pose, resulting in the estimated finger pose being unrealistic and improbable. The glove pose, on the other hand, though noisy and relatively inaccurate, gives a high-level estimate of the hand pose at most cases. Furthermore, being an α-weighted fusion of the depth and glove pose estimates, the merged pose adopts the best characteristics from its parent poses, and is able to reliably handle cases where either pose estimate fails. For instance, the merged pose accounts for the failure of the depth pose by using the corresponding information from the glove pose, and also enforces the collinearity constraint in the fingers making the pose realistic and plausible. Lastly, the finger-wise pose merging can also be seen in action in column 1 of Fig. 10 where the thumb is directly adopted from the depth pose whereas the other less-confident fingers are adopted from the glove pose.

Scenario 2 - Occlusion

Fig. 11 (a), Fig. 11 (b) and Fig. 11 (c) qualitatively assess the performance of the pose fusion algorithm when the hand

Fig. 10: Comparison on hand pose estimation. From top to bottom: RGB input, hand pose estimation of image-based [31], wearable-based [7], and our sensor fusion method.



(a) (b) (c)

Fig. 11: Qualitative evaluations of the depth, glove and merged poses when (a) the hand is being occluded by a foreign object, (b) the pose of the hand is changed behind the occlusion, and (c) the foreign object occluding the hand is gradually removed. We see that the pose fusion algorithm can smoothly transition from relying on both the depth and glove poses when the hand is unoccluded to using only the glove when the hand is fully occluded in both scenarios, where the foreign occlusion is gradually introduced and removed. Furthermore, the pose fusion algorithm works reliably even when the depth pose is unavailable for extended periods of time. These abilities of our pose fusion algorithm make the final merged pose superior to using only the depth or glove pose.

is actively being occluded by a foreign object under three scenarios, namely, 1) the hand is initially unoccluded and a foreign object starts occluding it, 2) the hand is already occluded and the pose of the hand is changed behind the occlusion, and 3) the hand is initially occluded and the occluding object is gradually removed.

It can be observed in Fig. 11 (a) that the pose fusion algorithm is able to smoothly transition from relying on using both the depth and glove poses to using only the glove pose when the hand is fully occluded. When there is no foreign object in the segmented depth image in Row 1, the confidence in the depth pose is high, and the merged pose biases towards it. As the foreign object starts occluding the hand, the depth pose becomes more ambiguous till the time no hand is visible, and the depth pose resorts to using the default hand pose. Meanwhile, the merged pose starts giving more weight to the glove pose, which is shown by the change in colour of the merged mesh from orange to green. This example demonstrates 1) our heatmap confidence estimation algorithm is able to estimate the confidence of the depth pose efficiently, and 2) our pose fusion algorithm can smoothly transition from using both the depth and glove poses to using only the glove pose when the hand is fully occluded.

Furthermore, Fig. 11 (b) shows that the change in hand pose when the hand is fully occluded is also effectively captured. Since the hand is fully occluded, the camera is rendered useless, and the depth pose uses the default hand pose and returns confidence of 0. Thus, the merged pose estimate fully relies on the glove pose estimate and directly returns the pose estimated from the glove.

Lastly, Fig. 11 (c) show that the removal of the foreign object occluding the hand is also accurately captured by

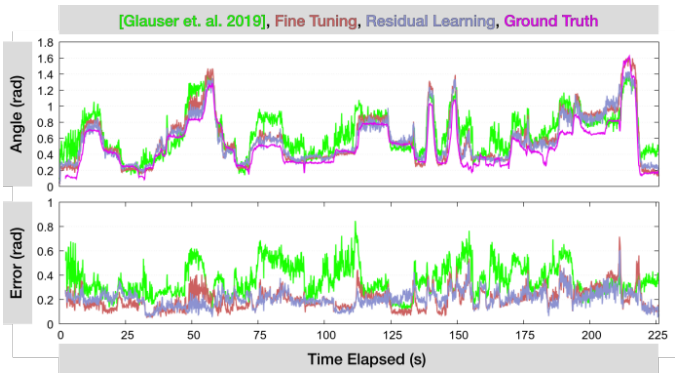


Fig. 12: Comparison of the absolute mean angles as estimated by the generic network (green) from [7], the fine-tuning (red) and side-tuning (greyish-blue) model personalisation approaches, and the the reference (purple) angles from [21]. The generic network has the largest error and performs the worst.

Mapping Model	μ (rad)	$\widetilde{x}$ (rad)	σ^2 (rad^2)
[7]	0.361	0.307	0.071
Fine-Tuning	0.206	0.153	0.035
Side-Tuning	0.209	0.144	0.039

TABLE 5: Mean (μ), median ($\widetilde{x}$), and variance (σ^2) of the absolute deviation of the average joint angle estimated by the generic network from [7] and the calibrated models created using the fine-tuning and side-tuning. Both the calibration approaches have similar inference performance, but are significantly better than the generic network from [7].

our method. As the occlusion is removed, the depth pose estimate becomes more accurate, and the confidence in the depth pose also increases. Consequently, the merged pose starts giving more weight to the depth pose, and the final merged pose output becomes more accurate. This increase in accuracy is explicitly captured by the change in the pose of the pinky finger. The pinky finger in the merged pose is initially bent as a result of the glove overestimating the joint angle, but as the occlusion is removed, the pinky finger in the merged pose starts reflecting the more accurate depth pose. When the occlusion in fully removed, the merged pose gives a high weight to the depth pose and almost exactly reflects it.

The two scenarios - No Occlusion and Occlusion - show that our pose fusion algorithm can handle both unoccluded and complex occlusion scenes, and it can thus be concluded that our merged pose output is superior to using either only the depth pose from the depth camera or only the glove pose from the stretch-sensing glove.

6.1.4 Fine-tuning vs. side-tuning

Fig. 12 compares the mean joint angles predicted by the generic model from [7], our fine-tuning-based and side-tuning-based user-calibrated model as compared to the reference angles from [21]. The models calibrated using both fine-tuning and side-tuning perform considerably better than the generic model with a mean absolute error of $0.206\,\mathrm{rad}$ and $0.209\,\mathrm{rad}$ respectively as compared to an error

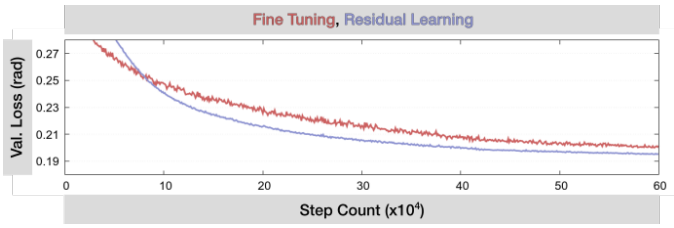


Fig. 13: A plot of the validation loss as a function of the step count for both the fine-tuning and side-tuning approaches. In the plot, fine-tuning is denoted using red, while side-tuning is represented in grey. The side-tuning architecture converges faster and to a smaller validation error than its fine-tuning counterpart, which makes it an ideal for online model calibration.

of $0.361\,\mathrm{rad}$ of the generic model as recorded in Table 5. However, since both model calibration approaches report similar errors and perform equally well, choosing one over the other requires comparison of their training convergence rates. Fig. 13 plots the validation loss during the training of the model calibration approaches as a function of the number of training steps using $\sim$500 training and $\sim$ 8700 validation samples. It can be inferred from the figure that the side-tuning-based approach converges faster with smaller validation error than its fine-tuning counterpart. Since model calibration is performed online, faster convergence of the network results in quicker access to an improved and more accurate hand pose estimate, which enhances the user experience and makes the calibration process less time-consuming.

6.2 Global hand tracking evaluation

When the hand can be detected by the RGB-D camera, the 3D joint position can be estimated accurately with an average error smaller than 10 mm, as reported by denseReg network [31]. Therefore, when the hand is visible in the camera, our fused estimate can be as accurate as the estimate from the RGB-D camera since a small covariance is assigned for it during filtering. It is essential to evaluate the quality of the estimate for challenging situations when the hand is not detected with the camera. Under this circumstance, the system can only count on the estimate using the skeleton model, which should reasonably localize the hand as well.

As shown in Fig. 14, to evaluate the accuracy of the estimate only using the skeleton model, we compute its error compared with the estimate using denseReg network [31] when the hand is detected with the camera. For our system with an auto-calibrated skeleton model, the average error is about 5.77 cm. This means that even when the camera provides no observation as the hand is occluded or moved out of field-of-view, our multi-sensor fusion system can still track the hand position within a reasonable error range.

In Fig. 15, we further visualize the tracking performance in two consecutive frames. For the first frame, we occlude the camera with paper, and only the skeleton model can be used for the global pose estimation. Then we keep the hand static and remove the paper for the second frame when the skeleton model and the camera observation are used. From the figure, we can see that the 3D position estimates between

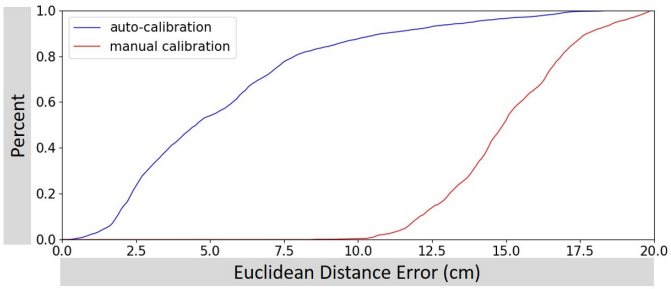


Fig. 14: Cumulative distribution function of the 3D translation error between the estimate using skeleton model as well as IMU readings and the estimate from the camera using denseReg network [31]. Two skeleton models are used in the evaluation, which is auto-calibrated with our proposed method and measured manually.

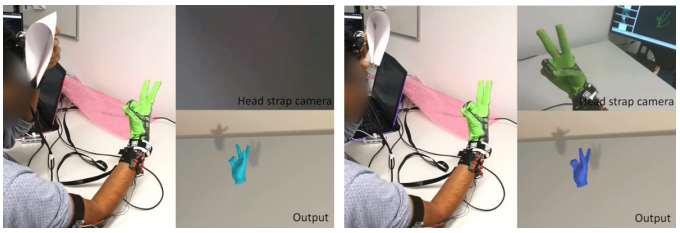


Fig. 15: Qualitative evaluation of the tracking performance in a challenging situation (hand is static). Left: the camera is fully occluded by paper. Right: after removing the occlusion, our tracking system takes the available observation from the camera as an input to compute the fused estimate. The fused results using only the skeleton model (left) and both the the skeleton model as well as the camera observation (right) has small discrepancy.

these frames has small discrepancy in the rendered camera observation. This further verifies the relatively small error between the estimate only using the skeleton model and that from camera. Otherwise there will exist an explicit jitter between two fused positions.

Moreover, from Fig. 14, we also notice that the result with our body model auto-calibration is much better than the one with manual calibration, whose average error is about 15.68 cm. This demonstrates the effectiveness of the proposed auto-calibration method. The main reason is the difficulty in measuring the lengths of some segments accurately, as we have discussed.

In conclusion, for our multi-sensor fusion system, we can not only track the hand position accurately when the hand is detectable in camera but also track the hand position with a reasonable error in the challenging situations when the hand is occluded or moved out of field-of-view.

7 Conclusion and Future Work

In conclusion, we successfully demonstrate the benefits of a multi-sensor hand tracking setup that leverages the powers of combining vision and wearable sensors. To track the hand position in 3D space, we introduce a multi-IMU global tracking algorithm based on a skeleton model and fuse its tracking result with visual 3D tracking information from the RGB-D camera whenever the hand is in view. For the local hand modeling, we combine the hand pose estimates from the stretch sensor and the RGB-D camera using an uncertainty metric computed from the heatmaps generated by the depth-to-pose model. Additionally, we show how the visual observations can be exploited to auto-calibrate both the stretch-sensing glove and the multi-IMU system to further push the accuracy of the overall algorithm. Extensive experimentation shows that our multi-sensor method outperforms existing camera-based and wearable-based methods in terms of accuracy, robustness, and calibration effort.

Currently, the trunk of the user needs to be static during the experiments. Under our framework, we can extend the system by attaching another IMU on the trunk to track its movement. Then the user can move freely. An obvious next step, would be to extend the proposed setup to allow for capturing both hands, the upper body, or even the full-body motion. As a challenging engineering task, one could try to integrate our setup with the state-of-the-art AR/VR headsets (e.g., HoloLens 2 [70] and Occulus Quest [71]) that equipped with cameras already. Another interesting direction is a possible replacement of the IMUs with a series of stretch sensors ([49] integrated into a textile sleeve.

Furthermore, it would be compelling to also design and add an uncertainty estimator for the sensor-to-pose model and build a universal pose fusion model. To further enhance the tracking and modeling accuracy, it could be beneficial not only to extract a skeletal hand representation but also a dense surface reconstruction from the depth point cloud [72].

References

[1] S. Sridhar, A. Oulasvirta, and C. Theobalt, "Interactive markerless articulated hand motion tracking using rgb and depth data," in *Proc. ICCV*, Dec. 2013.
[2] F. Mueller, D. Mehta, O. Sotnychenko, S. Sridhar, D. Casas, and C. Theobalt, "Real-time hand tracking under occlusion from an egocentric rgb-d sensor," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017, pp. 1284–1293.
[3] F. Mueller, F. Bernard, O. Sotnychenko, D. Mehta, S. Sridhar, D. Casas, and C. Theobalt, "Ganerated hands for real-time 3d hand tracking from monocular rgb," in *Proc. CVPR*, 2018, pp. 49–59.
[4] B. Fang, F. Sun, H. Liu, and D. Guo, "Development of a wearable device for motion capturing based on magnetic and inertial measurement units," *Scientific Programming*, vol. 2017, pp. 1–11, 01 2017.
[5] G. Saggio, F. Riillo, L. Sbernini, and L. R. Quitadamo, "Resistive flex sensors: a survey," *Smart Materials and Structures*, vol. 25, no. 1, p. 013001, dec 2015.
[6] J. T. Muth, D. M. Vogt, R. L. Truby, Y. Mengüç, D. B. Kolesky, R. J. Wood, and J. A. Lewis, "Embedded 3d printing of strain sensors within highly stretchable elastomers," *Advanced Materials*, vol. 26, no. 36, pp. 6307–6312, 2014.
[7] O. Glauser, S. Wu, D. Panozzo, O. Hilliges, and O. Sorkine-Hornung, "Interactive hand pose estimation using a stretch-sensing soft glove," *ACM Transactions on Graphics (TOG)*, 2019.
[8] G. Ponraj and H. Ren, "Sensor fusion of leap motion controller and flex sensors using kalman filter for human finger tracking," *IEEE Sensors Journal*, vol. 18, no. 5, pp. 2042–2049, 2018.
[9] L. Peppoloni, A. Filippeschi, E. Ruffaldi, and C. A. Avizzano, "A novel 7 degrees of freedom model for upper limb kinematic reconstruction based on wearable sensors," in *2013 IEEE 11th International Symposium on Intelligent Systems and Informatics (SISY)*, 2013, pp. 105–110.
[10] H. Zhou and H. Hu, "Reducing drifts in the inertial measurements of wrist and elbow positions," *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 3, pp. 575–585, March 2010.
[11] "ROS," https://www.ros.org/, 2020.
[12] "Unity," https://unity.com/, 2020.

[13] N. K. Iason Oikonomidis and A. Argyros, "Efficient model-based 3d tracking of hand articulations using kinect," in *Proceedings of the British Machine Vision Conference*. BMVA Press, 2011, pp. 101.1–101.11, http://dx.doi.org/10.5244/C.25.101.

[14] M. de La Gorce, D. J. Fleet, and N. Paragios, "Model-based 3d hand pose estimation from monocular video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 9, pp. 1793–1805, 2011.

[15] A. Tkach, M. Pauly, and A. Tagliasacchi, "Sphere-meshes for real-time hand modeling and tracking," *ACM Trans. Graph.*, vol. 35, no. 6, pp. 222:1–222:11, 2016.

[16] I. Oikonomidis, N. Kyriazis, and A. A. Argyros, "Full DOF tracking of a hand interacting with an object by modeling occlusions and physical constraints," in *Proc. ICCV*, 2011.

[17] H. Hamer, K. Schindler, E. Koller-Meier, and L. V. Gool, "Tracking a hand manipulating an object," in *2009 IEEE 12th International Conference on Computer Vision*, 2009, pp. 1475–1482.

[18] J. Taylor, R. Stebbing, V. Ramakrishna, C. Keskin, J. Shotton, S. Izadi, A. Hertzmann, and A. Fitzgibbon, "User-specific hand modeling from monocular depth sequences," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 644–651.

[19] D. J. Tan, T. Cashman, A. Fitzgibbon, D. Tarlow, S. Khamis, S. Izadi, and J. Shotton, "Fits like a glove: Rapid and reliable hand shape personalization," 06 2016, pp. 5610–5619.

[20] J. Taylor, L. Bordeaux, T. Cashman, B. Corish, C. Keskin, T. Sharp, E. Soto, D. Sweeney, J. Valentin, B. Luff, A. Topalian, E. Wood, S. Khamis, P. Kohli, S. Izadi, R. Banks, A. Fitzgibbon, and J. Shotton, "Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences," *ACM Trans. Graph.*, vol. 35, no. 4, pp. 143:1–143:12, 2016.

[21] A. Tkach, A. Tagliasacchi, E. Remelli, M. Pauly, and A. Fitzgibbon, "Online generative model personalization for hand tracking," *ACM Trans. Graph.*, vol. 36, no. 6, 2017.

[22] B. Doosti, "Hand pose estimation: A survey," *CoRR*, vol. abs/1903.01013, 2019.

[23] J. Tompson, M. Stein, Y. Lecun, and K. Perlin, "Real-time continuous pose recovery of human hands using convolutional networks," *ACM Trans. Graph.*, vol. 33, no. 5, Sep. 2014.

[24] M. Oberweger, P. Wohlhart, and V. Lepetit, "Hands deep in deep learning for hand pose estimation," *arXiv preprint arXiv:1502.06807*, 2015.

[25] M. Oberweger and V. Lepetit, "Deepprior++: Improving fast and accurate 3d hand pose estimation," in *International Conference on Computer Vision Workshops*, 2017.

[26] L. Ge, H. Liang, J. Yuan, and D. Thalmann, "3d convolutional neural networks for efficient and robust hand pose estimation from single depth images," in *Proc. CVPR*, 2017.

[27] G. Moon, J. Y. Chang, and K. M. Lee, "V2v-posenet: Voxel-to-voxel prediction network for accurate 3d hand and human pose estimation from a single depth map," *CoRR*, vol. abs/1711.07399, 2017. [Online]. Available: http://arxiv.org/abs/1711.07399

[28] S. Baek, K. I. Kim, and T. Kim, "Augmented skeleton space transfer for depth-based hand pose estimation," *CoRR*, vol. abs/1805.04497, 2018.

[29] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis, "Coarse-to-fine volumetric prediction for single-image 3d human pose," *CoRR*, vol. abs/1611.07828, 2016. [Online]. Available: http://arxiv.org/abs/1611.07828

[30] G. Moon, J. Y. Chang, Y. Suh, and K. M. Lee, "Holistic planimetric prediction to local volumetric prediction for 3d human pose estimation," *CoRR*, vol. abs/1706.04758, 2017. [Online]. Available: http://arxiv.org/abs/1706.04758

[31] C. Wan, T. Probst, L. Van Gool, and A. Yao, "Dense 3d regression for hand pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5147–5156.

[32] F. Mueller, M. Davis, F. Bernard, O. Sotnychenko, M. Verschoor, M. A. Otaduy, D. Casas, and C. Theobalt, "Real-time pose and shape reconstruction of two interacting hands with a single depth camera," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, pp. 1–13, 2019.

[33] Y. Zhou, M. Habermann, W. Xu, I. Habibie, C. Theobalt, and F. Xu, "Monocular real-time hand shape and motion capture using multi-modal data," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5346–5355.

[34] F. L. Hammond, Y. Mengüç, and R. J. Wood, "Toward a modular soft sensor-embedded glove for human hand motion and tactile pressure measurement," in *Proc. IROS*. IEEE, 2014, pp. 4000–4007.

[35] J.-B. Chossat, Y. Tao, V. Duchaine, and Y.-L. Park, "Wearable soft artificial skin for hand motion detection with embedded microfluidic strain sensing," in *Proc. ICRA*. IEEE, 2015, pp. 2568–2573.

[36] A. Rashid and O. Hasan, "Wearable technologies for hand joints monitoring for rehabilitation: A survey," *Microelectronics Journal*, 02 2018.

[37] J. Kim, N. D. Thang, and T. Kim, "3-d hand motion tracking and gesture recognition using a data glove," in *2009 IEEE International Symposium on Industrial Electronics*, 2009, pp. 1013–1018.

[38] R. Xu, S. Zhou, and W. J. Li, "Mems accelerometer based nonspecific-user hand gesture recognition," *IEEE Sensors Journal*, vol. 12, no. 5, pp. 1166–1173, 2012.

[39] D. Roetenberg, H. Luinge, and P. Slycke, "Xsens mvn: Full 6dof human motion tracking using miniature inertial sensors," *Xsens Motion Technologies BV, Tech. Rep*, vol. 1, 2009.

[40] Y. Huang, M. Kaufmann, E. Aksan, M. J. Black, O. Hilliges, and G. Pons-Moll, "Deep inertial poser: learning to reconstruct human pose from sparse inertial measurements in real time," *ACM Trans. Graph.*, vol. 37, no. 6, 2018.

[41] "CyberGlove," http://www.cyberglovesystems.com/, 2020.

[42] "ManusVR glove," https://manus-vr.odoo.com/hardware, 2020.

[43] "5DT Glove," http://www.5dt.com/data-gloves/, 2019.

[44] B. O'Brien, T. Gisby, and I. A. Anderson, "Stretch sensors for human body motion," in *Proc. SPIE*, vol. 9056, 2014, p. 905618.

[45] Z. Shen, J. Yi, X. Li, M. H. P. Lo, M. Z. Chen, Y. Hu, and Z. Wang, "A soft stretchable bending sensor and data glove applications," *IEEE Int. Conf. on Real-time Computing and Robotics (RCAR)*, vol. 3, no. 1, p. 22, 2016.

[46] T. F. O'Connor, M. E. Fach, R. Miller, S. E. Root, P. P. Mercier, and D. J. Lipomi, "The language of glove: Wireless gesture decoder with low-power and stretchable hybrid electronics," *PloS one*, vol. 12, no. 7, p. e0179766, 2017.

[47] W. Park, K. Ro, S. Kim, and J. Bae, "A soft sensor-based three-dimensional (3-d) finger motion measurement system," *Sensors*, vol. 17, p. 420, 02 2017.

[48] "StretchSense," https://stretchsense.com/, 2020.

[49] O. Glauser, D. Panozzo, O. Hilliges, and O. Sorkine-Hornung, "Deformation capture via self-sensing capacitive arrays," *ACM Trans. Graph.*, vol. 38, no. 2, pp. 16:1–16:16, 2019.

[50] C. Malleson, J. Collomosse, and A. Hilton, "Real-time multi-person motion capture from multi-view video and imus," *International Journal of Computer Vision*, pp. 1–18, 2019.

[51] X. Xiao and S. Zarar, "A wearable system for articulated human pose tracking under uncertainty of sensor placement," in *2018 7th IEEE International Conference on Biomedical Robotics and Biomechatronics (Biorob)*. IEEE, 2018, pp. 1144–1150.

[52] A. Gilbert, M. Trumble, C. Malleson, A. Hilton, and J. Collomosse, "Fusing visual and inertial sensors with semantics for 3d human pose estimation," *International Journal of Computer Vision*, vol. 127, no. 4, pp. 381–397, Apr 2019.

[53] C. Chen, R. Jafari, and N. Kehtarnavaz, "A real-time human action recognition system using depth and inertial sensor fusion," *IEEE Sensors Journal*, vol. 16, no. 3, pp. 773–781, 2016.

[54] G. Park, A. Argyros, J. Lee, and W. Woo, "3d hand tracking in the presence of excessive motion blur," *IEEE Transactions on Visualization and Computer Graphics*, vol. PP, pp. 1–1, 02 2020.

[55] S. Zhou, F. Fei, G. Zhang, J. D. Mai, Y. Liu, J. Y. J. Liou, and W. J. Li, "2d human gesture tracking and recognition by the fusion of mems inertial and vision sensors," *IEEE Sensors Journal*, vol. 14, no. 4, pp. 1160–1170, 2014.

[56] J. Wu and R. Jafari, "Zero-effort camera-assisted calibration techniques for wearable motion sensors," in *Proceedings of the Wireless Health 2014 on National Institutes of Health*, ser. WH '14. New York, NY, USA: Association for Computing Machinery, 2014, p. 1–8. [Online]. Available: https://doi.org/10.1145/2668883.2668888

[57] C. He, P. Kazanzides, H. T. Sen, S. Kim, and Y. Liu, "An inertial and optical sensor fusion approach for six degree-of-freedom pose estimation," *Sensors*, vol. 15, pp. 16 448–16 465, 07 2015.

[58] T. Helten, M. Muller, H.-P. Seidel, and C. Theobalt, "Real-time body tracking with one depth camera and inertial sensors," in *The IEEE International Conference on Computer Vision (ICCV)*, December 2013.

[59] T. von Marcard, R. Henschel, M. J. Black, B. Rosenhahn, and G. Pons-Moll, "Recovering accurate 3d human pose in the wild using imus and a moving camera," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 601–617.

[60] Z. Zheng, T. Yu, H. Li, K. Guo, Q. Dai, L. Fang, and Y. Liu, "Hybridfusion: Real-time performance capture using a single depth sensor and sparse imus," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 384–400.
[61] UltraLeap. (2020) Ultraleap. [Online]. Available: https://www.ultraleap.com/
[62] G. Park, T.-K. Kim, and W. Woo, "3d hand pose estimation with a single infrared camera via domain transfer learning," in *2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 2020, pp. 588–599.
[63] J. Hughes, A. Spielberg, M. Chounlakone, G. Chang, W. Matusik, and D. Rus, "A simple, inexpensive, wearable glove with hybrid resistive-pressure sensors for computational sensing, proprioception, and task identification," *Advanced Intelligent Systems*, p. 2000002, 2020.
[64] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *Kdd*, vol. 96, no. 34, 1996, pp. 226–231.
[65] J. O. Zhang, A. Sax, A. Zamir, L. Guibas, and J. Malik, "Side-tuning: Network adaptation via additive side networks," *arXiv preprint arXiv:1912.13503*, 2019.
[66] J. Tompson, M. Stein, Y. Lecun, and K. Perlin, "Real-time continuous pose recovery of human hands using convolutional networks," *ACM Trans. Graph.*, vol. 33, no. 5, p. 169, 2014.
[67] Z.-Q. Zhang and J.-K. Wu, "A novel hierarchical information fusion method for three-dimensional upper limb motion estimation," *IEEE transactions on instrumentation and measurement*, vol. 60, no. 11, pp. 3709–3719, 2011.
[68] E. Olson, "Apriltag: A robust and flexible visual fiducial system," in *2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 3400–3407.
[69] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-d point sets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, no. 5, pp. 698–700, Sep. 1987.
[70] "Hololens 2," https://www.microsoft.com/en-us/hololens, 2020.
[71] "Oculus Quest," https://www.oculus.com/quest/, 2020.
[72] J. Malik, A. Elhayek, and D. Stricker, "Whsp-net: A weakly-supervised approach for 3d hand shape and pose recovery from a single depth image," *Sensors*, vol. 19, no. 17, p. 3784, 2019.

Zhaopeng Cui is a research professor at the College of Computer Science and the State Key Laboratory of CAD & CG at Zhejiang University. He obtained his Bachelor's degree and Master's degree at Xidian University in 2009 and 2012 respectively, and received his PhD degree in computer science under the supervision of Prof. Ping Tan at Simon Fraser University. From 2017 to 2020, he worked as a Senior Researcher in the Computer Vision and Geometry Group led by Marc Pollefeys at ETH Zurich.

Hanxue Liang is a master student in the Robotics, System and Control program at ETH Zurich. He received his BE in Energy and Automation Engineering from Xi'an Jiaotong University, China. His research interests include 3D perception, object detection and pose estimation.

Oliver Glauser is a Researcher at Capskin Sensors. He did his PhD on self-sensing devices for motion and deformation capture at the Interactive Geometry Lab at ETH Zurich under the supervision of Olga Sorkine-Hornung.

Nikhil Gosala is a PhD student at the Robot Learning Lab in the University of Freiburg, Germany. He received his MSc in Computer Science from ETH Zurich, Switzerland, and BE in Computer Science from BITS Pilani, India. His research interests include perception, planning, localisation and sensor fusion.

Shihao Wu is a postdoc at the Interactive Geometry Lab in ETH. He received his PhD degree in the Computer Graphics Group in the University of Bern, MS degree at South China University of Technology, and BS degree at the South China Normal University. His research interests include computer graphics, deep geometric modeling, and point set processing.

Fangjinhua Wang is a first-year PhD student in Computer Science at the Computer Vision and Geometry group in ETH Zurich. He received his MSc in Robotics, Systems and Control from ETH Zurich, Switzerland, and BE in Mechatronics Engineering from Zhejiang University, China. His research interests include 3D Reconstruction, SLAM and Robotics.

Olga Sorkine-Hornung is a Professor of Computer Science at ETH Zurich. She leads the Interactive Geometry Lab at the Institute of Visual Computing. Prior to joining ETH, she was an Assistant Professor at the Courant Institute of Mathematical Sciences, New York University (2008-2011). She earned her BSc in Mathematics and Computer Science and PhD in Computer Science from Tel Aviv University (2000, 2006). Her research interests include geometry processing and interactive geometric modeling.