Back-Hand-Pose: 3D Hand Pose Estimation for a Wrist-worn Camera via Dorsum Deformation Network

Erwin Wu[1,2], **Ye Yuan**[1], **Hui-Shyong Yeo**[3], **Aaron Quigley**[4], **Hideki Koike**[2], **Kris M. Kitani**[1]

[1]Carnegie Mellon University, [2]Tokyo Institute of Technology,
[3]University of St Andrews, [4]University of New South Wales
wu.e.aa@m.titech.ac.jp, yyuan2@cs.cmu.edu, hsy@st-andrews.ac.uk,
a.quigley@unsw.edu.au, koike@c.titech.ac.jp, kkitani@cs.cmu.edu

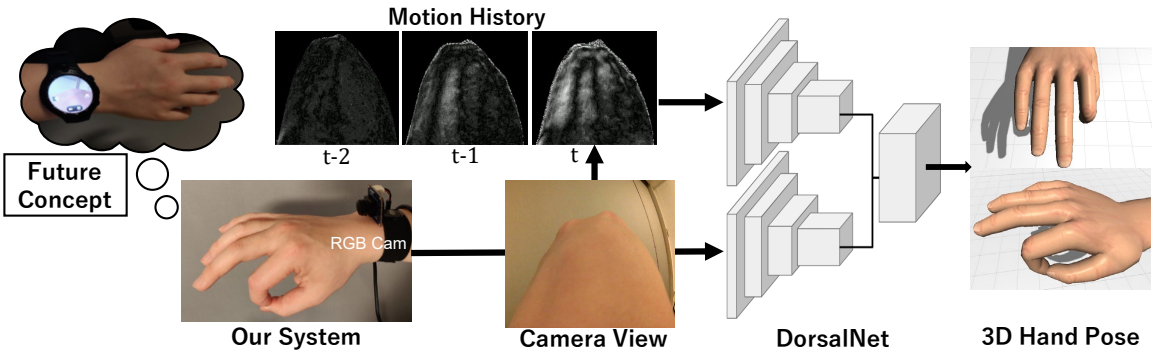

Figure 1. Overall flow of this system, 3D hand pose is estimated in real-time only using a camera looking at the back of the hand.

ABSTRACT

The automatic recognition of how people use their hands and fingers in natural settings – without instrumenting the fingers – can be useful for many mobile computing applications. To achieve such an interface, we propose a vision-based 3D hand pose estimation framework using a wrist-worn camera. The main challenge is the oblique angle of the wrist-worn camera, which makes the fingers scarcely visible. To address this, a special network that observes deformations on the back of the hand is required. We introduce DorsalNet, a two-stream convolutional neural network to regress finger joint angles from spatio-temporal features of the dorsal hand region (the movement of bones, muscle, and tendons). This work is the first vision-based real-time 3D hand pose estimator using visual features from the dorsal hand region. Our system achieves a mean joint-angle error of 8.81° for user-specific models and 9.77° for a general model. Further evaluation shows that our system outperforms previous work with an average of 20% higher accuracy in recognizing dynamic gestures, and achieves a 75% accuracy of detecting 11 different grasp types. We also demonstrate 3 applications which employ our system as a control device, an input device, and a grasped object recognizer.

UIST '20, October 20–23, 2020, Virtual Event, USA

ACM ISBN 978-1-4503-7514-6/20/10 ...$15.00.
https://doi.org/10.1145/3379337.3415897

Author Keywords

Wrist-worn devices; 3D hand pose estimation; Dorsal Hand

CCS Concepts

•Human-centered computing → Gestural input; •Computing methodologies → Artificial intelligence;

INTRODUCTION

Human hands are often used as the primary controller for digital input devices (e.g., mouse, keyboard or controllers) in instrumented settings, but these devices are challenging to use in mobile settings due to their limited portability, hindrance of natural manipulation (e.g., bulky data gloves) or limited range of sensing (e.g., VR controllers). Therefore, it is important to innovate new technology that can allow for capture of the human hand in mobile settings, in which the sensing devices can be naturally worn or held (e.g., wristband, wristwatch, lapel camera, smart glasses or smartphone). To this end, we propose a camera-based wrist-worn 3D hand pose recognition system with a form factor analogous to a smartwatch.

There is a long history of tracking natural hand gestures as input to computers [29, 30, 44], from glove-based to marker-based and then to marker-less vision tracking approaches. Numerous hand tracking approaches have been investigated targeting different environments – not limited to desktop interaction but also virtual reality (VR) and wearable computing. However, existing methods suffer from a diverse range of problems, as we will describe in detail in the related work section. Here we highlight the main problems as follows:

- Not portable - require external cameras in the environment.

- Limited FoV - tracking is lost when the hand is out of view.
- Bulky - need to wear devices such as a glove, extruded camera or to hold large controllers that inhibit finger movement.
- Require a custom sensor, i.e., EMG, NIR, ultrasound, etc.

In contrast to these approaches, this work suggests that a simpler and more natural installation is required. Our goal is to estimate 3D hand pose with a tiny camera that can be embedded on the side of a smartwatch or other wrist-worn device, where the camera only needs to observe the back of the hand (which we refer to as the "dorsal hand" region). However, in this configuration, the oblique angle and placement of the camera make key-point based computer vision techniques (e.g., OpenPose [42]) difficult to adopt because the fingers are not visible in the camera's field of view.

Indeed, research has shown that using the shape or visual cues from the dorsal hand region can be useful for understanding hand pose. For example, by attaching strain gauges [33] or photo-diodes [45] sensors directly on the back of the hand area, it is possible to measure tiny changes in the movement of the skin and tendons, and infer the hand pose. Yeo et al. [53] also build on this idea in the context of gesture recognition. They use an infrared (IR) camera to observe this area from few centimeters away and does not require direct contact measurement. However, all of the aforementioned works only try to recognize a few static hand poses or finger tapping gestures, without exploring the possibility of 3D hand pose estimation.

Inspired by prior work, we believe it is possible to estimate the full 3D hand pose using a similar setup, specifically a miniature fisheye camera on the side of a wrist-worn device. Yet, unlike previous work whose evaluation was limited to discrete hand pose or gestures, we address the task of full 3D hand pose estimation, which provides continuous tracking of finger flexion. We introduce a real-time 3D hand pose estimator using a two-stream convolutional recurrent network. First, we preprocesses the image from the camera into a masked hand image and calculates a motion history image [6]. Then, the two streams (masked hand and motion images) are passed to a residual convolutional network for spatial feature regression, followed by a long short-term memory (LSTM) network-based Kalman filter (KF) which are used to obtain temporal features. Finally, the features are mapped to a specific 3D hand representation and converted to a 3D hand mesh for visualization.

To show the potential of our system, three applications are implemented. The first application is a control system that can be used to interact with smartwatches, smartphones, or even act as a VR controller that supports bare-hand interaction. The second application, is a hand-based input system acting as a virtual mouse or keyboard, to control computing devices without holding a physical controller. Finally, using the results of a grasp recognition method, we build an object detection system to recognize specific items (e.g., pen, cola can or tape roll) that the user is grasping, which is helpful for discreet and tangible interaction or rehabilitation.

Our contributions in this paper are summarized as follows:

1. We propose a novel two-stream residual LSTM-KF network which is the first vision-based architecture to reconstruct angle-based hand pose from images of the back of hand.
2. Evaluations on 3D hand pose estimation with representative baseline which demonstrates the robustness of our approach across users and lighting, and an ablation study is carried out to validate the technical components.
3. Evaluations on gesture and grasp recognition are performed and discussed to show the usability of the system.
4. We demonstrate 3 types of applications to show the potential use-cases of this work.

The video, datasets as well as a trained model will be released at: https://github.com/erwinwu211/Back-Hand-Pose.

RELATED WORK

Our work is related to hand pose estimation, which has been studied in multiple research areas. For example, hand pose estimation is an active research topic within the computer vision community [11], but most of the research is based on external tracking infrastructure in the environment. A full coverage of these topics is beyond the scope of this paper; hence we refer the readers to respective surveys [29, 30] for further details. Here we narrow the focus to approaches that employ wearable devices, which can be briefly categorized into two groups: hand-worn and non-hand-worn approaches.

Non-hand-worn Approaches

Non-hand-worn approaches include wearing device such as head-mounted displays (e.g., HoloLens or Oculus Quest) or shoulder-mounted cameras [43] that can support fully articulated hand pose estimation via vision-based techniques. These approaches, however, require the hands to remain inside the field-of-view (FoV) of the camera to be tracked. It means the hands must be raised to a certain height for a certain period of time, which can cause fatigue. In fact, fatigue is one of the main issues when using this kind of input for a longer period, which is popularly known as the gorilla arm [22] issue.

Hand-worn Approaches

By contrast, approaches based on hand-worn devices aim to work all the time regardless of the hand being raised or lowered. Such hand-worn devices are further categorized into two groups: i) gloves and controller devices or ii) wristbands and armbands with embedded sensors.

Glove And Controller Devices

Many gloves have been proposed in the past for translating hand gestures into computer input. A comprehensive survey can be found in [16, 44]. Such glove devices, while able to provide high fidelity hand pose estimation and real-time position and orientation information, suffer from the problems of bulkiness and high cost. Not only do they take considerable time to put on, wearing a glove can result in a reduction in the tactile sensation/sensitivity of the fingertip, and can inhibit finger movements due to friction. Mainstream VR controllers (e.g., Oculus Touch, Valve Knuckle) are also able to infer partial hand poses based on proximity and capacitive sensors [48]. Recently, Arimatus and Mori [3] also proposed similar techniques for hand pose estimation on handheld devices with

proximity sensors. Similarly, holding a controller means the hand's natural movements are inhibited, and they are unable to perform a full grip.

Wristbands And Armbands With Embedded Sensors

There are a wide range of sensors that can be embedded or attached to a wristband or armband. Methods using Electromyography (EMG) [2, 37], force [14, 36], optical [18, 34, 38] and capacitive sensors [31, 40, 47] can measure the surface changes on the wrist or arm to infer hand pose or gestures. Methods using bio-acoustic sensing [1, 15, 28], ultrasound imaging [35, 41] and electrical impedance tomography [56] can measure the internal changes in the flesh to infer hand pose and gestures. One downside is that these approaches require very specific sensors that are less likely to be embedded in wearable devices, compared to an RGB camera. The main limitation, however, is the tracking fidelity of these approaches, which is typically limited to several static hand poses or gestures only, with the exception of Amma et al. [2], but their work requires high density EMG array.

Back of Hand Measurement

Closer to what is proposed here are the approaches introduced in BackHand [33], Behind the Palm [45, 27] and e-Skin Patch [24]. These approaches require the attachment of sensors that have direct contact with the back of the hand to measure tendon movements and skin deformation, such as flexible sticky pads, photo reflective diodes or stretchable e-skin patches. Again, these methods are limited to several static hand poses only.

Hand-worn Device with Camera

The next common type of approach is by wearing a camera (RGB, IR or depth) on the hand. For example, on the inner wrist, WristCam [50], Digits [26] and DigiTap [39] use a camera to track hand poses or finger tapping actions. WristCam [50] only requires a standard camera, but the supported gestures were limited. Digits [26] supports reconstructing the finger poses, but requires an IR laser line projector whereas DigiTap [39] requires a LED flash synced with an accelerometer to detect vibrations occurring during finger taps. CyclopsRing [9] uses a small fisheye lens worn on hand webbings, which enables the detection of seven hand gestures. Attaching a camera on the inner side of arm or hand simplifies the tracking problem because it can partially see the fingers.

On the other side of the arm, approaches that place the camera on the outer wrist, such as those can be embedded into a smartwatch, suffer from heavy occlusion as the fingers are not visible by the camera. To overcome the occlusion problem, Chen et al. [10] employ an elevated camera to track 10 ASL hand poses, which, however, could be bulky. Yeo et al. [53] use a wide angle IR camera to track 11 ASL poses and 5 finger tapping gestures, despite the occluded fingers, by looking at the back of the hand. Recently, Hu et al. [20] use 4 thermal cameras around the wrist to reconstruct full hand pose. Inspired by their work, our approach also focus on visual features at the dorsal hand, but we extend it to support richer, full hand pose estimation.

Deep Learning Approaches to Pose Estimation

Our approach is also inspired by hand and body pose estimation literature from the computer vision community, especially those employ deep learning techniques to deal with heavy occlusion or only rely on indirect features.

For example, using single fisheye camera on the head [52] or the chest [21], it was shown that a well-trained network is able to reconstruct full body pose, despite the majority part of the body is not visible within the camera's field of view. As mentioned before, there are numerous approaches for egocentric hand tracking or gesture recognition using either head-, shoulder- or chest-mount camera [43]. These methods rely on, relatively, a "Third Person View" (TPV) of the hand which can almost see the entire hand. Zhou et al. [59] introduce the state-of-the-art real-time hand pose estimation using a TPV RGB camera. They include feature maps for hand pose regression using ResNet, and design a novel IK-net for training inverse kinematics parameters, which helps to reconstruct the hand joint angles from joint positions.

Nevertheless, different from TPV regression, our goal should be considered as a "First Person View" egocentric hand tracker using indirect features on the back of the hand. Yuan et al. [54, 55] introduce a network that is able to estimate full body pose using only images from an egocentric head-mount camera that is looking at the environment, based on the idea of imitation learning using optical flow and bi-directional LSTM. Their work is a nice illustration on how to extract indirect temporal features and map them to postures.

The closest work to us, Yeo et al. [53] use a two-stream (TS) CNN to learn the temporal deformation of the back of the hand to classify static and dynamic hand poses. Their work uses an IR camera to help segment the hand from the background. But IR cameras are difficult to put into smartwatches since they hardly work when exposed under sunlight. Moreover, their accuracy for recognizing dynamic gestures is very limited ($\sim 50\%$) which is impractical.

METHODOLOGY

The ultimate goal is to extract visual features on the dorsum of a hand, such as the deformations of skin, veins, or tendons, and to build a neural network to learn how to regress these features into the motion of each finger. After all, our goal is not just gesture recognition, but full hand pose estimation.

For 3D hand representation, instead of location-based 3D coordinates, we use the relative joint angle-based representation [25] for the 4 fingers except the thumb, which is independently

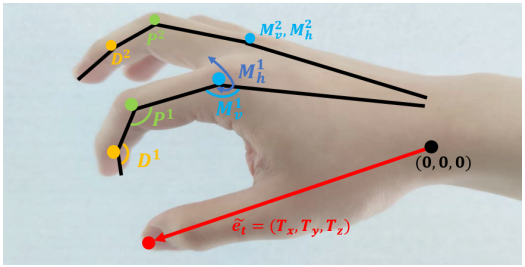

Figure 2. Our 3D hand model representations, the thumb is represented by a single 3D vector and the other 4 fingers are using joint angle.

Figure 3. Comparison with commercial smartwatch, (top) the hardware comparison and (bottom) the cropped images from our camera and the raw images from both Zeblaze.

estimated by end point position (as shown in Figure 2, M, P, D stand for the MCP, DIP, PIP joints of the specific finger, while the *v* and *h* stand for the vertical and horizontal bending of MCP). This is because, in our initial study, we found that the angle of each joint (except thumb) is, to some extent, related to the deformation of a specific area of the dorsal hand, which could make the task of regressing finger joint angles easier. For the thumb, it is more difficult to detect the relevant deformations since they mainly take place on the side of the arm. After a number of trials, we decided to treat the thumb separately and to let the network learn to estimate a 3D vector of the thumb top from the edge information of the dorsal hand, and we then recover the thumb joint angles using inverse kinematics [32]. The joint angle error we use is another commonly used metric for 3D hand pose estimation which is widely employed [49, 58].

For hardware, we use a wide angle RGB camera, that has less environmental restrictions and is more likely to be found in smartwatches than IR cameras (used by previous work [53]), which suffers from stray infrared light from the sun. However, RGB cameras, different from IR cameras, cannot benefit from the easy segmentation of removing the background.

Therefore, we perform a hand segmentation to reduce noise from the background. In addition, it is difficult to observe subtle deformations on the back of the hand from a single image frame (Figure 4). In this paper, we proposed a novel architecture of two-stream LSTM network with Kalman filter (DorsalNet, as shown in Figure 5). The overall idea is to perform temporal deformation extraction from the back of the hand for 3D hand pose estimation. The input images taken by the camera are preprocessed to obtain masked RGB hand images and motion history images. Following this, the networks regress the hand joint angles from the spatial and temporal input, and finally the 3D hand pose is reconstructed with the help of a hand simulator and inverse kinematics (IK) for the thumb. This procedure is done continuously at an interactive rate of approximately 20 frames per second in our simulation, which is acceptable for most of the applications for wearable devices proposed.

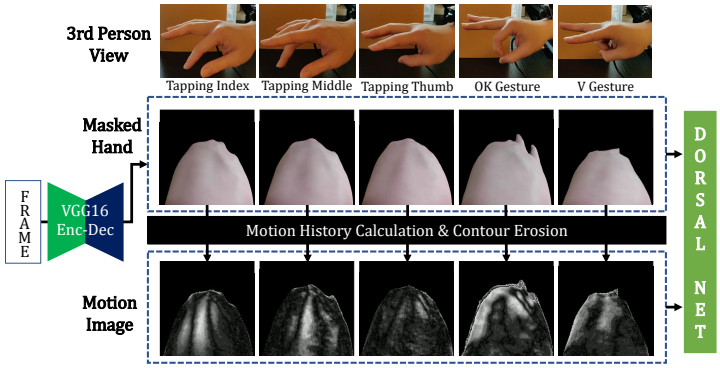

Figure 4. Image Preprocessing: this figure shows how images of 5 different gestures are preprocessed by the auto encoder and motion history.

Hardware and Software

There are commercial off-the-shelf (COTS) smartwatches with built-in side camera for photography purposes[1]. To note that, such side camera typically has a small FoV, is slightly tilted up and is positioned vertically for photography purpose, which leads to only a partial view of the hand, as shown in Figure 3 (right). In this paper, to simplify the implementation as well as to ensure enough FoV and clear imaging of hands, we employ an USB fisheye camera for prototyping. The camera unit (Model: ELP-USBFHD01M-L21) has a 170-degrees FOV (of which we cropped to 120-degrees) and a 2.1mm focal length.

Nonetheless, we expect a camera with a wider FoV can be integrated into smartwatches, e.g., by attaching or replacing with a fisheye lens. For example, CyclopsRing [9] enables a 185-degrees FoV using a miniature camera of 14 mm in diameter and 15 mm in height with a fisheye lens, which can fit between the gaps of the fingers. We are also seeing plenty of smartphones with 120-degrees wide angle lens camera nowadays. As shown in Figure 3 (left), our camera module is tilted 8° down with a lens length of 11 mm and a measured height of 9 mm from the skin level. The highest part of our camera module is 24 mm due to the PCB height, which could be further reduced in future manufacturing or by detaching the camera module from the PCB, as done by Hu et al. [20]. For comparison, Chen et al. [10] use an elevated camera with lens height of 55 mm from the skin level, whereas Yeo et al. [53] use the Leap Motion camera with lens height of 15 mm and module height of 32 mm from skin level.

Other hardware includes a Leap Motion camera for collecting ground truth hand pose, a computer with NVIDIA RTX 2080 Ti, i7-9700 CPU is used for training and evaluating the model. For the real-time test, a laptop with GTX 1070 (Max-Q) and i7-8750h is used. For software, the images are processed by OpenCV 4.1, and the camera exposure time is set to 1/50 s. All training are performed using Python 3.7, TensorFlow 1.15 and Keras 2.31. The Unity 3D editor is used to calculate inverse kinematics for the thumb and to visualize the 3D hand mesh.

Data Preprocessing

To obtain robustness and usability, we performed several preprocessing steps including data augmentation, hand segmentation and motion image processing. Because a wrist-worn

[1]Zeblaze Thor SmartWatch: https://amzn.com/B07MC3H6CW/

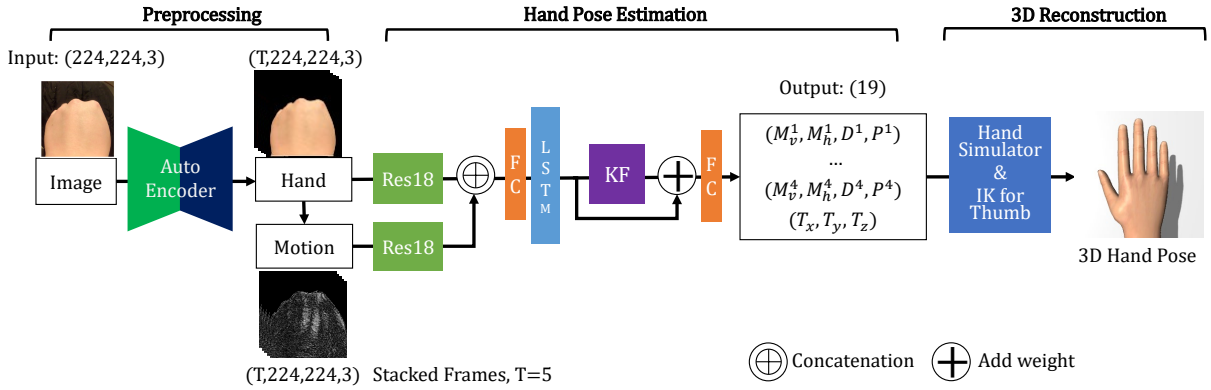

Figure 5. DorsalNet overview, consists of 3 part: the preprocessing stage with the encoder-decoder network for hand masking and the motion image computation; the two-stream LSTM CNN with the KF as an feature extractor; and the hand simulator which reconstruct finger angle to 3D hand pose.

device is not always tightly fixed to the arm, the camera could have some slight rotations around the arm. Therefore, for each input image, we augment the data by rotating the image clockwise with varying angles from -10° to 10° with step size 5°, with the same ground truth. This resulted in 5 times amount of data to enhance robustness across device locations.

Hand segmentation is undertaken by fine-tuning an encoder-decoder network [4] to generate hand masks. In our setup, as the camera is fixed on the arm and looks directly at the dorsum of hand, the bottom half of the image is mostly occupied by the hand (as shown in Figure 4). Thus, it is relatively easy to mask the hand from background. As a first step, data for segmentation was collected from 6 participants (one female, aged between 25-30) across races of East Asian, Mediterranean, and European. All participants are students from the computer science department of two universities from different countries. They were told to wear our device and walk naturally inside a laboratory for about 2 minutes which results in 9,600 images being collected. All images are then masked by color range and contour using OpenCV. Afterwards, the brightness and hand color of these images are changed for data augmentation. For each image, in the HSV color model, the H value is increased/decreased by a random value which generated 10 different color image including the original, and the brightness (v) value is also changed to -20%, -10%, 10%, and 20% for each image. As a result, we train the auto-encoder to create the hand mask of 480,000 images (50 times the amount of the original data). We randomly split the dataset into training (80%) and testing (20%), and the mean precision of the test result of generated hand mask is 98.9% in pixel scale.

As mentioned before, temporal motion images are required for training the two-stream network, which should be generated using pairs of adjacent frames. We explored the common Dense Optical flow (OF) (KV-L1), Lattice OF [51], PWC-Net [46] used in Ego-Pose [55], and motion history images (MHI) [5] used by Opisthenar [53]. Since the deformations need to be captured in a pixel-perfect way in real-time, it turns out that a refined version of the MHI shown in Figure 4 is the best solution, which provides great accuracy with fast computation speed. Different from the Opisthenar [53], our tweaked version use the parameter $\alpha = 0.2$ which means it is observing the weighted sum of 5 past frames. Another problem that might occur is that the network might focus on the hand contour movement instead of the skin deformations, which will harm the network's generalizability. Therefore, an erosion operation is added to the hand masks to filter the outer-edge, and the intensity inside the hand is increased to let the network focus on inner motions on the back of the hand.

Network Architecture

For each training sequence of length T (in this paper, we use T=5), the preprocessed data consists of the masked hand images $I_{1:T}$, the motion history images (MHI) $X_{1:T}$, and the hand pose labels $y_{1:T}$. Each hand pose y_t includes the joint angles $\alpha_t^1, \alpha_t^2, \alpha_t^3, \alpha_t^4$ of the index, middle, ring and little fingers and the 3D position e_t of the thumb top. As shown in Fig. 2, the joint angle α_t^i of each finger has four elements (M_v^i, M_h^i, P^i, D^i) where M_v^i, M_h^i correspond to the vertical and horizontal rotation of the first joint and P^i, D^i correspond to the rotation angles of the second and third joint respectively. Our goal is to learn a neural network based regressor $\tilde{y}_{1:T} = f(I_{1:T}, X_{1:T})$ that maps the input masked images $I_{1:T}$ and MHI $X_{1:T}$ to a sequence of estimated hand poses $\tilde{y}_{1:T} = f(I_{1:T}, X_{1:T})$.

To this end, we propose DorsalNet, a two-stream LSTM based network whose architecture is outlined in Figure 5. For each timestep t, two ResNet18 [19] are used to extract visual features from the masked hand image I_t and the MHI X_t respectively. The two visual features are then concatenated together and passed through a fully-connected layer to form a unified visual feature ϕ_t. Previous research [53] already showed that simple two stream CNN is not sufficient for extracting temporal features of the back of hand. Thus, we use an LSTM layer to process the visual feature sequence $\phi_{1:T}$ into a temporal feature sequence, which is proved to be useful in 3D pose estimation [51, 55]. On the other hand, we noticed that most of our finger motions are simple linear movements, which could be regularized by a Kalman filter (KF). However, KF require a motion model and measurement model to be specified a priori, which are often only crude approximations of reality. In the work of Coskun et al. [12], they introduced a LSTM-based KF to use LSTM to learn the motion and noise model, which shows promising effect on learning human dynamics. Therefore, this architecture is imported to obtain a more stable temporal feature sequence $\psi_{1:T}$. We also add a residual connection to bypass the Kalman filter for more direct feature

learning, so the network will choose whether to use Kalman filter based on the hand motion. For each frame t, the temporal feature ψ_t now includes information from past frames to help make hand pose predictions. Finally, another fully-connected layer is added to map the temporal feature ψ_t to the estimated hand pose $\tilde{y}_t$. We use a single LSTM instead of the three from the previous work [12], because the two stream CNN architecture is heavy in computation, we focus on light-weighting the whole networks to achieve a real-time inference time. That is also the reason why ResNet18 is used but not deeper CNN architecture such as ResNet50 or ResNet101. As a result, the inference time of the whole network using the mid-range notebook PC mentioned in the hardware section is approximately 38ms. To provide supervision for training the DorsalNet, we define the following loss function:

$$L(y_t, \hat{y}_t) = L_{\text{fingers}} + L_{\text{thumb}}\,, \tag{1}$$

$$L_{\text{fingers}} = \frac{1}{16}\sum_{i=1}^{4} \left\| \alpha_t^i - \tilde{\alpha}_t^i \right\|^2, \tag{2}$$

$$L_{\text{thumb}} = \frac{1}{\pi^2}\arccos^2\left(\frac{e_t \cdot \tilde{e}_t}{|e_t||\tilde{e}_t|}\right), \tag{3}$$

where we use symbols with tilde to indicate it is the estimated output of the network and symbols without tilde to indicate ground truth. We also use different losses for the fingers and thumb because their pose representations are different. For the fingers, we use mean squared error (MSE) as the loss for the joint angles as shown in equation (2); for the thumb, we compute the angle between the estimated thumb top vector and the ground truth one as the loss function (3).

Hand Simulation

Once the DorsalNet is trained, it could be used to extract estimated hand pose sequence $\tilde{y}_{1:T}$ from a given video. To visualize the hand pose sequence, we use a hand simulator that can map a hand pose $\tilde{y}_t$ to a 3D mesh, which can be rendered by a graphics pipeline. As the simulator also uses joint angle representation for the thumb pose, we employ inverse kinematics (IK) to solve for the joint angles of the thumb using the estimated thumb top position $\tilde{e}_t$. Finally a smooth function of the simulator is employed for filtering noisy result. The inference time of the simulator is 10 ms, which result in a 41 ms inference time with the network, which could be considered real-time sufficient for our use case.

Till here, we could reconstruct the full 3D hand pose except for the wrist rotation. As a simple implementation, we calculate the centroid of the masked hand using its moments, and map it to the wrist flexion-extension and ulnar-radial deviation. Pronation-supination movement is not considered here because the camera is fixed on the user's arm.

EVALUATION

Participants & Data Collection

We evaluate our system from 3 different perspectives: hand pose estimation, gesture recognition, and grasp recognition. Data was collect from 5 out of the 6 participants who also participated in the previous segmentation study. Similar to prior work [53], we collected data of static gestures of American sign language (ASL) digits (0-9), and dynamic gestures of finger tapping. During the study, the participants were asked to put the right arm on an armrest and to wear our camera with Velcro tape to perform the action.

The entire collection procedure includes 5 sessions for all 15 gestures (both static and dynamic). In each session, the user was told to re-wear the camera and start from a relaxed hand posture to do the specific gesture repeatedly (for static gestures, the user have to return to relaxed posture every time). We asked the users to perform the gesture in a normal speed but the frequency is controlled by themselves, approximately 1 ASL gesture per 3 seconds and 1 tap per second were collected.

An auto-labeling program is written for multi-threading the Leap Motion API and camera image acquiring, where it is calibrated so that the root of the thumb becomes the origin, as depicted in Figure 2. We also fix the camera frame rate to 20 FPS to simplify the synchronization and to align with the inference frame rate. For each session of each gesture, 30 seconds of video at 20 FPS was collected. As a result, video of 600 frames was collected 5 times for all 15 gestures for each participant, which resulted in a total of 225,000 frames. These data were used in the training and evaluation for the hand pose estimation and gesture recognition. For the grasp recognition, we only use the mentioned data for pretraining, but use another dataset for fine-tuning and evaluation (which will be described in later sections). Also, we collect a single-user dataset of different lighting condition which will be described in the *Lighting Condition Study*. In all sections, the ratio of the train and test data split is set to 8:2.

3D Hand Pose Estimation

Procedure

The evaluation of hand pose estimation consists of three separate studies. We first trained our network on an individual user's data to evaluate the personalized model. This aims to study the precision of each specific joint and finger, which could be helpful for future improvement. For comparison, since our work is the first real-time hand pose estimation system using egocentric wrist-worn camera, some similar state-of-the-art networks dealing with direct/indirect pose estimation were used as baselines. Nevertheless, we also carried out a lighting condition study to study the robustness of our network and an ablation study of different network architectures and different inputs on the basis of the proposed method.

Finger and Joint Error Study

We first trained our network on individual subjects to study the precision of each finger and joint. Five personalized models were trained and evaluated on each specific user's data, 20% (9000 frames) of the user's data was randomly kept for this test. Table 1 shows the average result of 5 individual models, all results are recovered to angle unit for a better visibility, where the unit is degree. The first 4 rows show the mean absolute error (MAE) and its standard deviation (SD) of each joint of the 4 fingers, respectively, together with an average result of each joint. The columns stand for each finger joint rotation and the last row is vector angle error of the thumb.

Joint\Finger	Index (1)		Middle (2)		Ring (3)		Pinky(4)		Joint Avg.	Thumb (0)	
	MAE	SD	MAE	SD	MAE	SD	MAE	SD	MAE	MAE	SD
MCPv	7.05°	±0.40	6.32°	±0.54	6.3°	±0.39	6.92°	±1.21	**6.65°**		
MCPh	7.94°	±0.75	7.87°	±0.62	7.17°	±0.64	9.78°	±1.99	**8.14°**		
DIP	6.92°	±0.59	6.78°	±0.76	6.70°	±0.70	9.80°	±1.73	**7.55°**	12.69°	±2.26
PIP	8.47°	±0.92	7.85°	±0.98	7.66°	±0.87	11.11°	±1.33	**8.77°**		
Finger Avg.	**7.60°**		**7.20°**		**6.96°**		**9.40°**				

Table 1. Average result of the individual model of each joint/finger (metrics: MAE(SD) unit: degree).

Method	Individual	General	Leave-1-user
Nearest N.[13]	18.44°	21.78°	20.89°
Direct(ResNet18)	18.39°	22.09°	29.11°
Yuan et al. [55]	12.48°	14.40 °	14.53°
Yeo et al. [53]	16.67°	18.52°	20.24°
Zhou et al. [59]	15.95°	20.06°	21.06°
Ours (w/o KF)	9.28°	10.33°	10.71°
Ours (w/ KF)	**8.81°**	**9.77°**	**9.72°**

Table 2. Comparison with baseline methods, Our methods are divided into with/without Kalman filter (KF).

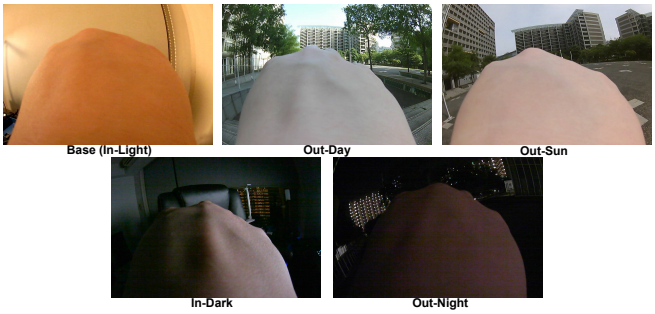

Figure 6. Images of camera under different lighting conditions.

	Base(In-Light)	Out-Day	In-Dark	Out-Night
MAE	7.93°	7.77°	8.46°	8.21°

Table 3. Comparing the accuracy of our method in different lighting condition (Out-Sun removed due to lack of ground truth).

Comparison Study

Next, to show the effect of our network compared with baseline conditions. In this study, we trained both the 5 individual models and a general model for each network. Here, we used a session-split of leaving one specific session (9000 frames for one subject, 45000 frames for general model) for the Individual and General model to study the cross-session generalization of our system. As well as a user-split of leaving one user out, to perform a cross-user validation.

Baselines: As mentioned in the *Procedure* section, since there is no identical work for comparison, we used some typical standards or similar networks as baselines. The Direct (ResNet18) is the condition that directly regresses raw camera images to the 3D representations frame-by-frame, which can be considered as a base condition. Also, we included the Nearest Neighbour Search [13], also known as k-nearest neighbour (k=1), because it is a typical standard for pose estimation. Since the CNN-LSTM architecture we used is similar to the work of Yuan et al. [55], we also include them as baselines, even though they used bi-directional LSTM which means their networks are offline. Another baseline is the work by Yeo et al. [53] which we followed-up. Although their system is not designed for full hand pose regression, we changed the output of their network and fine-tuned with our dataset. Instead of using a Leap Motion camera, we used a monochrome masked hand as the input. Zhou et al. [59] is the state-of-the-art real-time 3D hand capture methods using a single monocular camera. Their network used a location map to extract positional features and regress the 3D joint location of the hand. They also used an IK-Net for learning inverse kinematics to recover the joint location to joint angle and match the output with the MANO hand model. To compare with this work, we fine-tuned their network by changing the input to our raw egocentrics dorsal hand images.

Together with the 5 baselines above, our method with/without Kalman filter are analyzed. All results are using joint angle-based representations while the baseline of Zhou et al. [59] also outputs the full thumb joint rotation since they use the MANO hand model. Therefore, we re-calculate the thumb vector from their output which might cause inaccuracy. However, we believe the overall performances are still comparable.

Lighting Condition Study

Our study is mostly done in an indoor with fluorescent lamp lighting condition. To show the performance of our network in different lighting, we also conduct a comparison of angle MAE in different conditions shown in Figure 6. We asked one of the participants to take data under 4 other lighting conditions besides the base condition (In-Light), which are:

- Outdoor Day: Natural day light on a cloudy day outside.
- Outdoor Sun: Strong sunlight on a fine-weather sunny day.
- Indoor Dark: The lamp is turned off with only stray light from a PC monitor.
- Outdoor Night: Only light from street lamp at night.

In all condition, we take the same quantity of data as the former studies from the participant, which results in 45000 frames for each lighting. However, in the Out-Sun condition, we cannot use Leap Motion to capture the ground truth due to high intensity infared light so only the other 4 conditions are evaluated. Qualitative performance of Out-Sun is shown in our video.

Ablation Study

Starting from the very basic two CNN networks (VGG16 and ResNet18), we analyze the effect of the network by gradually adding other model parts. This ablation study is mainly

Architecture (Input)	Angle Error		Inference
	Individ.	General	Time (ms)
VGG16 (RGB)	16.07	18.19	54
ResN18 (RGB)	16.11	18.70	17
ResN18+LSTM (RGB)	11.95	14.01	35
ResN18+LSTM (Motion)	9.29	10.69	33
ResN18+LSTM (TS)	9.28	10.13	40
Ours (w/o Data Aug.)	9.35	11.11	40
Ours (TS)	**8.81**	**9.77**	41

Table 4. Results of ablation study of different network architecture and input data. The metrics of Angle Error is MAE (degree), TS stands for two-stream input, 'Ours' stands for ResN18+LSTM+KF (TS).

comparing how different input and different temporal feature extraction will affect the precision of the hand tracking, and the same data were used as the comparison study. Three different types of input together with a with/without data augmentation condition were compared, while the network is changed by with/without LSTM or Kalman filters. In total, 7 conditions are compared as shown in Table 4, the inference time (ms) using the laptop is also recorded for comparison. To notice, the *ResNet18 (RGB)* method here is different from the *Direct (ResNet)* in the former study for it uses the masked hand images preprocessed by our system instead of raw images.

Results of 3D Hand Pose

Finger and Joint Error Study: The result (Table 1) shows that the index, middle, and ring fingers achieve higher precision (MAE around 7) since the deformations occur in the middle of image, while the pinky finger performs worst (MAE=9.40). For the joints, it is a bit surprising that the MCPhs also do not perform well (MAE=8.14), worse than the DIPs (MAE=7.55), while the PIPs are the worst (MAE=8.77).

Comparison Study: When compared with other baseline methods (Table 2), the proposed method with KF outperforms the baseline with a large advantage. (MAE: Individual =8.81), General=9.77, Leave-1-user=9.72). Even the proposed method w/o KF leads the baseline with an average of approximately 4-degree error. In the baseline methods, the work from Yuan et al. performs the best (MAE: Individual=12.48, General=14.40, Leave-1-user=14.53). Also, different from other methods, the proposed method does not show a great difference between the general, leavel-1-user, and individual model, which could be a proof of the generality of our network.

Lighting Condition Study: From Table 3, we can tell that the performance becomes worse when the lighting gets darker. However, the difference is relatively small between the best (Out-Day, MAE=7.77) and the worst (In-Dark, MAE=8.46).

Ablation Study: Observing the results (Table 4), in the first and second row, the VGG16 and the ResNet18 show similar results, yet the ResNet18 is much faster in inference time. Comparing different inputs of row 3-5, the motion input (MAE: Individual=9.29, General=10.69) obtains a much higher accuracy with less inference time than the RGB input (MAE: Individual=11.95, General=14.01), while two-stream input obtains a higher accuracy in the general model (Motion: General=10.69; TS: General MAE=10.33) with a slightly greater inference time. For the network architecture, comparing row 2 with row 3, we can notice there is a 4-degree difference with/without LSTM. Also, comparing row 5 and 7, it is evident that the LSTM-based KF outperforms normal LSTM with the highest accuracy. Overall, it is clear that with two streams of input and more complex networks, the accuracy becomes higher. Lastly, the difference from row 6 and row 7 indicates that, using data augmentation will greatly increase the general accuracy (from 11.11 to 9.77, 12% increase).

Gesture Recognition

Procedure

To demonstrate the usability of this work, we also compared with the state-of-the-art work, the Opisthenar [53], using back hand visual features for gesture classification. This is done by adding a multilayer perceptron (MLP) to the end of the DorsalNet and uses softmax activation function to do categorization. In reference to the study in their paper [53], besides the leave-1-session-out split done in the individual- and general-model evaluations, we also add a leave-1-user-out split evaluation for the general model to show the cross-user accuracy.

Baselines

Since the target of this evaluation is to compare with Opisthenar [53], we only include the Nearest Neighbour and Direct Regression methods using ResNet18 as another two baselines. It is not consequential to compare with other vision-based state-of-the-art gesture recognition methods using RGB inputs, because most of them are specialized in Third-Person-View and optimized for dealing with occlusion problems, which is not suitable for this study that primarily extracting the deformation of the image.

Results of Gesture Recognition

The results are divided into static gestures (ASL0-9) and dynamic gestures (Tapping 0-4), as shown in the Table 5 and Table 6. The metric in the table is the mean percentage while the individual and general column is the same as before. The Leave-1-user result is the averaged result of leaving one specific user for testing, which is repeated for all users (5 times).

Methods	Static ASL Gesture Accuracy		
	Individ.	General	Leave-1-user
Nearest N. [13]	44.6%	37.4%	33.4%
Direct Regression	71.2%	69.0%	57.2%
Yeo et al. [53]	88.6%	82.5%	70.4%
Ours (w/o KF)	**91.4%**	**88.8%**	**84.8%**
Ours (w/ KF)	90.2%	**88.8%**	83.0%

Table 5. Accuracy of static gestures (ASL0-9) classification.

Methods	Dynamic Tapping Accuracy		
	Individ.	General	Leave-1-user
Nearest N. [13]	47.8%	40.0%	28.4%
Direct Regression	51.0%	50.4%	39.8%
Yeo et al. [53]	59.6%	62.2%	55.0%
Ours (w/o KF)	85.2%	85.0%	77.6%
Ours (w/ KF)	**89.4%**	**86.8%**	**79.8%**

Table 6. Accuracy of dynamic gestures (5 finger-tappings) classification.

In the static results, both of our methods outperform the Opisthenar work by Yeo et al. [53] and what surprised us is that the proposed method without Kalman filter performs the best with an individual model with the accuracy of 91.4% and in Leave-1-user model with 84.8%. In the general model, both our methods achieve the same result of 88.8%. One assumption is that because in the static gestures there are less periodic motions which might cause an error in the prediction of the KF. However, both results show a high precision of detecting hand gestures with no obvious overtraining. By contrast, we also noticed that the Opisthenar obtains high accuracy in individual (88.6%) but a great decrease in general and leave-1-user model (82.5%, 70.4%, respectively), which might indicate an overfitting to the user of their methods.

Unlike the small improvements in the static results, our method outperforms Yeo et al.'s method with an overall 25% higher accuracy. The methods with KF perform the best in all three conditions (89.4% for individual models, 86.8% for general models, and 79.8% for leave-1-user model). These results show the ability of our network to detect dynamic hand postures which could extend the use of the system. Also, the dynamic results of our methods are close to the static ones, which indicates the back hand deformation is suitable for both dynamic and static classification.

Grasp Recognition

Procedure

During the development of this system, we identified the potential of using the dorsal hand features to recognize a range of different grasp types. Therefore, in this study, we collect another dataset of different grasp types as shown in Figure 7. The new collected data are used for fine-tuning the proposed network, of which the last layer is changed to a dense layer with softmax activation and output of 11, for a categorical estimation. Since the input size is fixed to T=5, for an input of grasping with size T_g ($T_g \geq 5$), a window-shift with stride 1 is performed, which result in $T_g - T + 1$ prediction outputs. All these outputs are thresholded and averaged for the top-1 result. Because there are also some approaching frames and holding frames where the hand is not grasping, we manually labeled these frames as "other" class to reduce noise. During the test, we first filtered the outputs contains "other" class, only those input without "other" frames were evaluated.

Grasping Dataset

The grasps can be classified either by its purpose or its shape. In this paper, 11 different classes of grasp (as shown in Figure 7) were chosen by referring to the work of Cai et al. [7, 8, 17]. The original Thumb-n-Finger grasp was substituted to Thumb-Index (T1F), Thumb-Middle (T2F), and Thumb-Ring (T3F) grasp, in order to obtain a more precise effect of different finger, (Thumb-pinky was not considered because it is a limited gesture). For this study, we gathered data from another 5 participants (3 of them are also subjects of the hand pose dataset). To ensure the participants make the grasp motion correctly, specific items were prepared for them to hold according to the grasp types. Again, 5 sessions of data collection were performed for each grasp per person, in which 10 grasps

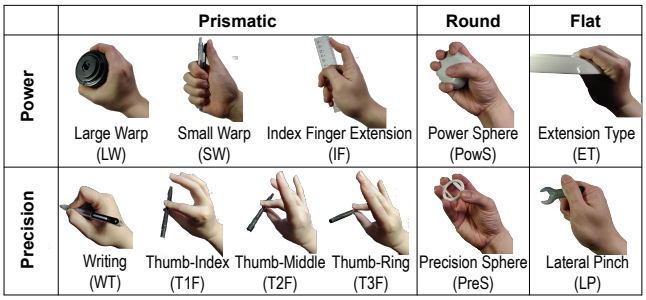

Figure 7. The 11 grasp types used in the grasp recognition evaluation.

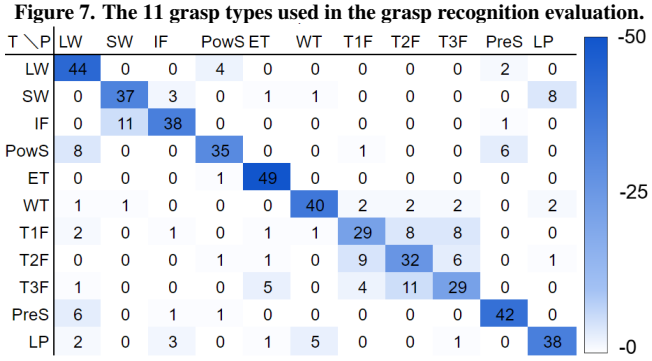

Figure 8. Confusion matrix for the grasp recognition result.

(about 15 frames/grasp) were collected for each grasp. As a result, we have approximately 40000 images for fine-tuning. In the evaluation, we still left one session out (10 grasps for each user result in 50 grasps) for testing and the rest for training.

Results of Grasp Recognition

The Figure 8 shows the confusion matrix with heatmap of the 50 tests for each grasp type. The overall accuracy is 75.1%, while the best one appears to be the extension type (holding a plate) with 49 correct classifications (98%). However, as we assumed, the T1F, T2F, and T3F gesture only achieved 58%, 64%, and 58% of accuracy, respectively. The matrix could indicate that these three grasps look similar which will confuse the network. But we still could see the potential of distinguishing these tiny differences using dorsal hand features. Besides these 3 motions, the Index Finger Extension (IF) also seems to be mis-classified as Small Wrap with a high frequency (22%).

APPLICATION

Based on the evaluation result, we also implement several applications to show the usability of the system (Figure 9).

Smart Devices Control: As a typical usage of an interaction system, we developed a controller for a smartwatch (Figure 9 (left)). Previous work [53] also mentioned a similar application using gesture recognition, which is, however, limited by the available gestures. Since our system is able to estimate full hand pose, linear control such as changing the time with finger angle becomes possible. Similar usage could also be extended to a VR/MR controller as a wearable barrier-free hand tracker. Furthermore, full hand pose allows 3D manipulation of virtual objects or expressive communication in remote collaboration.

Virtual Mouse & Keyboard: Similar to the first application, this hand tracker could also act as an one-handed input device such as mouse or keyboard. Even without holding a real

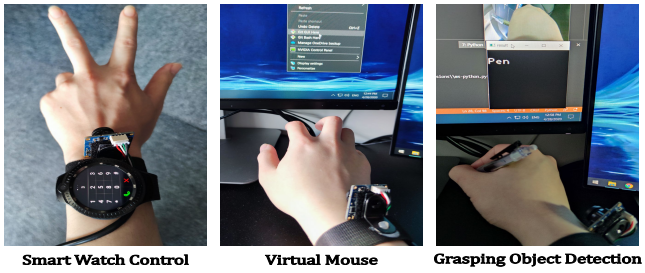

Figure 9. Three types of different demo applications.

mouse, users could use their wrist rotation to control the position of pointer and perform clicking as if using a real mouse. For typing input, instead of normal QWERTY keyboard, we developed a simple 8-key keyboard which works similar to the 10-key pad on a smartphone (please refer to video figure).

Grasping Object Recognizer: Another application is based on the grasp recognition (Figure 9 (right)). Since the evaluation results validated that it is possible to classify different grasp types from the back of the hand images, it should also work for object grasping recogniton with a limited object set. For example, Gripmarks [57] and Metaphoric Hand [23] allow summoning of different tools just by changing hand pose. This does not require explicit mode or tool switching, thus could have a fast and natural affordance.

DISCUSSION

Performing six studies on three types of use cases provided us with a number of interesting insights, and the persuasive quantitative results demonstrate the usability of this approach. The main novelty of this work lies in developing a new network architecture to estimate hand pose from an egocentric dorsal hand view. While some prior work already identified the features on the back of the hand and applied it to gesture recognition, here the real-time 3D hand pose tracking is more challenging and provides a wider range of use.

From the results of the four studies in hand pose estimation, we could imagine a clear picture about the performance, with an almost half the angle-error than the other baseline. To notice, here we only compared with vision-based techniques because the main focus of this study is system that could be naturally embedded in wearable devices. The result of each joint shows that the index, middle, and ring finger gain a relatively high accuracy estimated by back of the hand features. And, the result of an average angle error of 8.81° for individual and 9.77° in general even outperforms some methods using TPV camera [58] where the fingers can be clearly seen. Also, the result of lighting condition study and the ablation study could provide information which might be helpful in developing robust networks for extracting temporal deformations.

By applying our method to gesture recognition, we show that our system outperforms the previous work [53] with almost 30% higher accuracy (86.8% in general) in detecting dynamic finger tapping. This accuracy enriches the range of applications and provides possibility to recognize high frequency gestures (such as keyboard typing and piano playing). On the other hand, the static result just slightly exceeds the previous work, and it seems that the Kalman filter will cause a decrease in accuracy for non-periodical gestures.

With an average result of 75.1% for 11 classes for individual model, the grasp recognition shows some potential of detecting the grasp type from back hand features, while the pick-motion with the different finger can be confused with each other. However, it could be useful in a situation with limited range of items (such as VR or remote cooperation). We also noticed that grasp recognition might be helpful to rehabilitation usage such as analysis of the grasp of people with/without hypodactyly.

For the hand segmentation, we augmented the data by changing the color or brightness of the dorsal hand and achieved a high accuracy, but it is not sufficient to claim our network is robust to different types of hand, without testing on diverse users. There are multiple factors that might affect the result, such as skin color, skin thickness, hair volume, hand shape, etc. However, to note that, from the ablation study, we can observe that the motion-only input performs very close to the two-stream input, and surpasses the RGB-only input. From which we can tell the network is more looking at the overall deformations than the color information of the hand. Also, one of our participants had hairy skin and the features are still successfully extracted (which is not enough to claim this generalizability). Nevertheless, in the use case of wearing a personal smartwatch, the system is not necessary to be generalized but can be initially calibrated to the user by collecting a small amount of data and fine-tuning the model, this will result in a personalized model with higher accuracy and might also work as a security identification using the dorsal hand.

Lastly, the system faces difficulties with the wrist articulation. When the wrist is tilted up/down at extremes, the system fails to track hand pose temporarily for not seeing the dorsum of hand. This problem only happens in specific degree of vertical wrist rotation and does not suffer from horizontal rotation because of the limited horizontal range of human wrist. Also, in most applications, we assume that the user's dorsal hand would appear in the camera view when the hand is interacting with other objects or when the hand is in a natural, flat position.

Limitations & Future Work

From the results and discussions, we gained a detailed understanding of this approach. However, the work still has some limitations, which we discuss here and outline future plans.

The first limitation is the number of users in this study (N=5). Due to the current pandemic it was difficult for us to collect more diverse participants. We aim to conduct a study with a larger population, with a wider range of skin, skin age, blemishes or scars and hair characteristics along with more detailed tasks in the future.

Secondly, besides the angle of wrist which we discussed before, there is also a limitation that fast wrist motion causes camera motion blur, and it impacts our preprocessing pipeline that takes 5 frames for calculating motion history. While we envision that a camera with a higher frame rate and lower exposure time could mitigate this issue, and in common use cases, there are few situations which include fast wrist movement.

Nonetheless, it is also possible to use a hierarchy network detecting stable hand motion as a trigger of our system.

We used a simple method to estimate the wrist rotation based on image moments which was satisfactory in our test case. Future work should compare and evaluate deep neural network based methods. Also, our system currently does not estimate the pronation-supination (roll) movement. This problem could be solved by fusing the gyroscope data from the smartwatch or calculating the camera rotation from background movement.

In the lighting condition study, the hand detector works well even in a moderately dark environment where the hand could be barely captured as well as in a strong sunlight condition (as shown in Figure 6 and video figure). However, it could not work in a completely dark condition. An IR camera may overcome this problem but is susceptible to sunlight during outdoor use. A future solution could switch between RGB and an IR camera for day/night use or add a small lighting device.

Lastly, this approach needs further improvement for real-use. We did test with the images from a real smartwatch with a side camera (Figure 3) by padding the images to fit into our network input, and we are able to track the index, middle and ring finger to some extent, albeit the model was trained with data from a different camera. In future work, we would collect a dataset directly from a smartwatch and evaluate its efficacy.

CONCLUSION

In this paper, we have introduced DorsalNet, a network for 3D hand pose estimation by detecting the deformation of the back of the hand. We employ a camera on the outer side of the wrist, which could be further embedded in wrist-worn devices.

Despite the difficulties of feature extraction, such as the finger occlusion due to the camera placement, we succeeded in designing the network to regress indirect features and map it to finger angle. We performed 6 studies which show that our system obtains high accuracy in tracking 3D hand pose (MAE=8.81°/9.77°/9.72°for individual/General/Leave-1-user), recognizing static (88.8% for General) and dynamic (86.8% for General) gesture, and detecting grasp type (75.1% Overall) across different users and lighting conditions.

Finally, we introduced some example of applications to demonstrate the potential of our system. Although there is room for improvement in the approach and system that we would like to explore in the future, we believe that the current result of study and the data as well as the network architecture can provide insights of how to design a hand pose estimator from an egocentric back hand view.

ACKNOWLEDGEMENT

We thank the ACs and reviewers for their constructive feedback which helped improved this manuscript. This work is funded by JST CREST, Grant Number JPMJCR17A3 and JST AIP Acceleration, Grant Number JPMJCR20U1, Japan.

REFERENCES

[1] Brian Amento, Will Hill, and Loren Terveen. 2002. The Sound of One Hand: A Wrist-Mounted Bio-Acoustic Fingertip Gesture Interface. In *CHI '02 Extended Abstracts on Human Factors in Computing Systems (CHI EA '02)*. Association for Computing Machinery, New York, NY, USA, 724–725. DOI: http://dx.doi.org/10.1145/506443.506566

[2] Christoph Amma, Thomas Krings, Jonas Böer, and Tanja Schultz. 2015. Advancing Muscle-Computer Interfaces with High-Density Electromyography. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. Association for Computing Machinery, New York, NY, USA, 929–938. DOI: http://dx.doi.org/10.1145/2702123.2702501

[3] Kazuyuki Arimatsu and Hideki Mori. 2020. Evaluation of Machine Learning Techniques for Hand Pose Estimation on Handheld Device with Proximity Sensor. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–13. DOI:http://dx.doi.org/10.1145/3313831.3376712

[4] V. Badrinarayanan, A. Kendall, and R. Cipolla. 2017. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, 12 (Dec 2017), 2481–2495. DOI: http://dx.doi.org/10.1109/TPAMI.2016.2644615

[5] A. F. Bobick and J. W. Davis. 2001. The recognition of human movement using temporal templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23, 3 (March 2001), 257–267. DOI: http://dx.doi.org/10.1109/34.910878

[6] G. R. Bradski and J. Davis. 2000. Motion segmentation and pose recognition with motion history gradients. In *Proceedings Fifth IEEE Workshop on Applications of Computer Vision*. 238–244. DOI: http://dx.doi.org/10.1109/WACV.2000.895428

[7] M. Cai, K. M. Kitani, and Y. Sato. 2017. An Ego-Vision System for Hand Grasp Analysis. *IEEE Transactions on Human-Machine Systems* 47, 4 (Aug 2017), 524–535. DOI:http://dx.doi.org/10.1109/THMS.2017.2681423

[8] Minjie Cai, Kris M. Kitani, and Yoichi Sato. 2018. Understanding hand-object manipulation by modeling the contextual relationship between actions, grasp types and object attributes. *CoRR* abs/1807.08254 (2018). http://arxiv.org/abs/1807.08254

[9] Liwei Chan, Yi-Ling Chen, Chi-Hao Hsieh, Rong-Hao Liang, and Bing-Yu Chen. 2015. CyclopsRing: Enabling Whole-Hand and Context-Aware Interactions Through a Fisheye Ring. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15)*. ACM, New York, NY, USA, 549–556. DOI: http://dx.doi.org/10.1145/2807442.2807450

[10] Feiyu Chen, Jia Deng, Zhibo Pang, Majid Baghaei Nejad, Huayong Yang, and Geng Yang. 2018. Finger Angle-Based Hand Gesture Recognition for

Smart Infrastructure Using Wearable Wrist-Worn Camera. *Applied Sciences* 8, 3 (2018). DOI: http://dx.doi.org/10.3390/app8030369

[11] Xinghao Chen. 2020. Awesome Hand Pose Estimation, A curated list of related resources for hand pose estimation. (2020). https://github.com/xinghaochen/awesome-hand-pose-estimation.

[12] H. Coskun, F. Achilles, R. DiPietro, N. Navab, and F. Tombari. 2017. Long Short-Term Memory Kalman Filters: Recurrent Neural Estimators for Pose Regularization. In *2017 IEEE International Conference on Computer Vision (ICCV)*. 5525–5533. DOI: http://dx.doi.org/10.1109/ICCV.2017.589

[13] Pádraig Cunningham and Sarah Jane Delany. 2007. k-Nearest Neighbour Classifiers. (2007).

[14] Artem Dementyev and Joseph A. Paradiso. 2014. WristFlex: Low-power Gesture Input with Wrist-worn Pressure Sensors. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. ACM, New York, NY, USA, 161–166. DOI: http://dx.doi.org/10.1145/2642918.2647396

[15] T. Deyle, S. Palinko, E. S. Poole, and T. Starner. 2007. Hambone: A Bio-Acoustic Gesture Interface. In *2007 11th IEEE International Symposium on Wearable Computers*. 3–10. DOI: http://dx.doi.org/10.1109/ISWC.2007.4373768

[16] L. Dipietro, A. M. Sabatini, and P. Dario. 2008. A Survey of Glove-Based Systems and Their Applications. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 38, 4 (July 2008), 461–482. DOI: http://dx.doi.org/10.1109/TSMCC.2008.923862

[17] T. Feix, R. Pawlik, H. Schmiedmayer, J. Romero, and D. Kragic. 2009. A Comprehensive Grasp Taxonomy. In *Robotics, Science and Systems: Workshop on Understanding the Human Hand for Advancing Robotic Manipulation*.

[18] Rui Fukui, Masahiko Watanabe, Tomoaki Gyota, Masamichi Shimosaka, and Tomomasa Sato. 2011. Hand Shape Classification with a Wrist Contour Sensor: Development of a Prototype Device. In *Proceedings of the 13th International Conference on Ubiquitous Computing (UbiComp '11)*. Association for Computing Machinery, New York, NY, USA, 311–314. DOI: http://dx.doi.org/10.1145/2030112.2030154

[19] K. He, X. Zhang, S. Ren, and J. Sun. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 770–778. DOI: http://dx.doi.org/10.1109/CVPR.2016.90

[20] Fang Hu, Peng He, Songlin Xu, Yin Li, and Cheng Zhang. 2020. FingerTrak: Continuous 3D Hand Pose Tracking by Deep Learning Hand Silhouettes Captured by Miniature Thermal Cameras on Wrist. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 4, 2, Article 71 (June 2020), 24 pages. DOI: http://dx.doi.org/10.1145/3397306

[21] D. Hwang, K. Aso, and H. Koike. 2019. MonoEye: Monocular Fisheye Camera-based 3D Human Pose Estimation. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. 988–989. DOI: http://dx.doi.org/10.1109/VR.2019.8798267

[22] Sujin Jang, Wolfgang Stuerzlinger, Satyajit Ambike, and Karthik Ramani. 2017. Modeling Cumulative Arm Fatigue in Mid-Air Interaction Based on Perceived Exertion and Kinetics of Arm Motion. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. Association for Computing Machinery, New York, NY, USA, 3328–3339. DOI: http://dx.doi.org/10.1145/3025453.3025523

[23] Y. Jang, I. Jeon, T. Kim, and W. Woo. 2017. Metaphoric Hand Gestures for Orientation-Aware VR Object Manipulation With an Egocentric Viewpoint. *IEEE Transactions on Human-Machine Systems* 47, 1 (Feb 2017), 113–127. DOI: http://dx.doi.org/10.1109/THMS.2016.2611824

[24] S. Jiang, L. Li, H. Xu, J. Xu, G. Gu, and P. B. Shull. 2020. Stretchable e-Skin Patch for Gesture Recognition on the Back of the Hand. *IEEE Transactions on Industrial Electronics* 67, 1 (Jan 2020), 647–657. DOI: http://dx.doi.org/10.1109/TIE.2019.2914621

[25] Jintae Lee and T. L. Kunii. 1995. Model-based analysis of hand posture. *IEEE Computer Graphics and Applications* 15, 5 (Sep. 1995), 77–86. DOI: http://dx.doi.org/10.1109/38.403831

[26] David Kim, Otmar Hilliges, Shahram Izadi, Alex D. Butler, Jiawen Chen, Iason Oikonomidis, and Patrick Olivier. 2012. Digits: Freehand 3D Interactions Anywhere Using a Wrist-worn Gloveless Sensor. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology (UIST '12)*. ACM, New York, NY, USA, 167–176. DOI: http://dx.doi.org/10.1145/2380116.2380139

[27] Wakaba Kuno, Maki Sugimoto, and Yuta Sugiura. 2019. Finger Posture Estimation by Measuring Skin Deformation on Back of Hand. *The Journal of The Institute of Image Information and Television Engineers* 73, 3 (2019), 595–601. DOI: http://dx.doi.org/10.3169/itej.73.595

[28] Gierad Laput, Robert Xiao, and Chris Harrison. 2016. ViBand: High-Fidelity Bio-Acoustic Sensing Using Commodity Smartwatch Accelerometers. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST '16)*. ACM, New York, NY, USA, 321–333. DOI: http://dx.doi.org/10.1145/2984511.2984582

[29] Joseph J. LaViola. 1999. *A Survey of Hand Posture and Gesture Recognition Techniques and Technology*. Technical Report. USA.

[30] Rui Li, Zhenyu Liu, and Jianrong Tan. 2019. A survey on 3D hand pose estimation: Cameras, methods, and datasets. *Pattern Recognition* 93 (2019), 251 – 272. DOI: http://dx.doi.org/https://doi.org/10.1016/j.patcog.2019.04.026

[31] X. Liang, H. Heidari, and R. Dahiya. 2017. Wearable Capacitive-Based Wrist-Worn Gesture Sensing System. In *2017 New Generation of CAS (NGCAS)*. 181–184. DOI:http://dx.doi.org/10.1109/NGCAS.2017.80

[32] Cheng-Chang Lien and Chung-Lin Huang. 1998. Model-based articulated hand motion tracking for gesture recognition. *Image and Vision Computing* 16, 2 (1998), 121 – 134. DOI:http://dx.doi.org/https://doi.org/10.1016/S0262-8856(97)00041-3

[33] Jhe-Wei Lin, Chiuan Wang, Yi Yao Huang, Kuan-Ting Chou, Hsuan-Yu Chen, Wei-Luan Tseng, and Mike Y. Chen. 2015. BackHand: Sensing Hand Gestures via Back of the Hand. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15)*. ACM, New York, NY, USA, 557–564. DOI: http://dx.doi.org/10.1145/2807442.2807462

[34] Jess McIntosh, Asier Marzo, and Mike Fraser. 2017a. SensIR: Detecting Hand Gestures with a Wearable Bracelet Using Infrared Transmission and Reflection. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology (UIST '17)*. ACM, New York, NY, USA, 593–597. DOI: http://dx.doi.org/10.1145/3126594.3126604

[35] Jess McIntosh, Asier Marzo, Mike Fraser, and Carol Phillips. 2017b. EchoFlex: Hand Gesture Recognition Using Ultrasound Imaging. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 1923–1934. DOI: http://dx.doi.org/10.1145/3025453.3025807

[36] Jess McIntosh, Charlie McNeill, Mike Fraser, Frederic Kerber, Markus Löchtefeld, and Antonio Krüger. 2016. EMPress: Practical Hand Gesture Classification with Wrist-Mounted EMG and Pressure Sensing. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 2332–2342. DOI: http://dx.doi.org/10.1145/2858036.2858093

[37] Leap Motion. 2019. LeapUVC Documentation. (2019). https://github.com/leapmotion/leapuvc/blob/master/LeapUVC-Manual.pdf.

[38] Santiago Ortega-Avila, Bogdana Rakova, Sajid Sadi, and Pranav Mistry. 2015. Non-Invasive Optical Detection of Hand Gestures. In *Proceedings of the 6th Augmented Human International Conference (AH '15)*. Association for Computing Machinery, New York, NY, USA, 179–180. DOI: http://dx.doi.org/10.1145/2735711.2735801

[39] Manuel Prätorius, Dimitar Valkov, Ulrich Burgbacher, and Klaus Hinrichs. 2014. DigiTap: An Eyes-free VR/AR Symbolic Input Device. In *Proceedings of the 20th ACM Symposium on Virtual Reality Software and Technology (VRST '14)*. ACM, New York, NY, USA, 9–18. DOI:http://dx.doi.org/10.1145/2671015.2671029

[40] Jun Rekimoto. 2001. GestureWrist and GesturePad: Unobtrusive Wearable Interaction Devices. In *Proceedings of the 5th IEEE International Symposium on Wearable Computers (ISWC '01)*. IEEE Computer Society, Washington, DC, USA, 21–. http://dl.acm.org/citation.cfm?id=580581.856565

[41] S. Sikdar, H. Rangwala, E. B. Eastlake, I. A. Hunt, A. J. Nelson, J. Devanathan, A. Shin, and J. J. Pancrazio. 2014. Novel Method for Predicting Dexterous Individual Finger Movements by Imaging Muscle Activity Using a Wearable Ultrasonic System. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 22, 1 (Jan 2014), 69–76. DOI: http://dx.doi.org/10.1109/TNSRE.2013.2274657

[42] T. Simon, H. Joo, I. Matthews, and Y. Sheikh. 2017. Hand Keypoint Detection in Single Images Using Multiview Bootstrapping. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 4645–4653. DOI: http://dx.doi.org/10.1109/CVPR.2017.494

[43] Mohamed Soliman, Franziska Mueller, Lena Hegemann, Joan Sol Roo, Christian Theobalt, and Jürgen Steimle. 2018. FingerInput: Capturing Expressive Single-Hand Thumb-to-Finger Microgestures. In *Proceedings of the 2018 ACM International Conference on Interactive Surfaces and Spaces (ISS '18)*. Association for Computing Machinery, New York, NY, USA, 177–187. DOI:http://dx.doi.org/10.1145/3279778.3279799

[44] D. J. Sturman and D. Zeltzer. 1994. A survey of glove-based input. *IEEE Computer Graphics and Applications* 14, 1 (Jan 1994), 30–39. DOI: http://dx.doi.org/10.1109/38.250916

[45] Y. Sugiura, F. Nakamura, W. Kawai, T. Kikuchi, and M. Sugimoto. 2017. Behind the palm: Hand gesture recognition through measuring skin deformation on back of hand by using optical sensors. In *2017 56th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*. 1082–1087. DOI: http://dx.doi.org/10.23919/SICE.2017.8105457

[46] D. Sun, X. Yang, M. Liu, and J. Kautz. 2018. PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8934–8943. DOI:http://dx.doi.org/10.1109/CVPR.2018.00931

[47] Hoang Truong, Shuo Zhang, Ufuk Muncuk, Phuc Nguyen, Nam Bui, Anh Nguyen, Qin Lv, Kaushik Chowdhury, Thang Dinh, and Tam Vu. 2018. CapBand: Battery-Free Successive Capacitance Sensing Wristband for Hand Gesture Recognition. In *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems (SenSys '18)*. Association for Computing Machinery, New York, NY, USA, 54–67. DOI: http://dx.doi.org/10.1145/3274783.3274854

[48] Valve. 2020. Valve Index Controllers. (2020). https://www.valvesoftware.com/en/index/controllers.

[49] Josien C. van den Noort, Henk G. Kortier, Nathalie van Beek, DirkJan H. E. J. Veeger, and Peter H. Veltink. 2016. Measuring 3D Hand and Finger Kinematics—A Comparison between Inertial Sensing and an Opto-Electronic Marker System. *PLOS ONE* 11, 11 (11 2016), 1–16. DOI: http://dx.doi.org/10.1371/journal.pone.0164889

[50] Andrew Vardy, John Robinson, and Li-Te Cheng. 1999. The WristCam As Input Device. In *Proceedings of the 3rd IEEE International Symposium on Wearable Computers (ISWC '99)*. IEEE Computer Society, Washington, DC, USA, 199–. http://dl.acm.org/citation.cfm?id=519309.856464

[51] E. Wu and H. Koike. 2019. FuturePose - Mixed Reality Martial Arts Training Using Real-Time 3D Human Pose Forecasting With a RGB Camera. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. 1384–1392. DOI: http://dx.doi.org/10.1109/WACV.2019.00152

[52] W. Xu, A. Chatterjee, M. Zollhöfer, H. Rhodin, P. Fua, H. Seidel, and C. Theobalt. 2019. Mo2Cap2: Real-time Mobile 3D Motion Capture with a Cap-mounted Fisheye Camera. *IEEE Transactions on Visualization and Computer Graphics* 25, 5 (May 2019), 2093–2101. DOI: http://dx.doi.org/10.1109/TVCG.2019.2898650

[53] Hui-Shyong Yeo, Erwin Wu, Juyoung Lee, Aaron Quigley, and Hideki Koike. 2019. Opisthenar: Hand Poses and Finger Tapping Recognition by Observing Back of Hand Using Embedded Wrist Camera. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology (UIST '19)*. Association for Computing Machinery, New York, NY, USA, 963–971. DOI: http://dx.doi.org/10.1145/3332165.3347867

[54] Ye Yuan and Kris Kitani. 2018. 3D Ego-Pose Estimation via Imitation Learning. In *Computer Vision – ECCV 2018*, Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss (Eds.). Springer International Publishing, Cham, 763–778.

[55] Y. Yuan and K. Kitani. 2019. Ego-Pose Estimation and Forecasting As Real-Time PD Control. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 10081–10091. DOI: http://dx.doi.org/10.1109/ICCV.2019.01018

[56] Yang Zhang and Chris Harrison. 2015. Tomo: Wearable, Low-Cost Electrical Impedance Tomography for Hand Gesture Recognition. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15)*. ACM, New York, NY, USA, 167–173. DOI: http://dx.doi.org/10.1145/2807442.2807480

[57] Qian Zhou, Sarah Sykes, Sidney Fels, and Kenrick Kin. 2020b. Gripmarks: Using Hand Grips to Transform In-Hand Objects into Mixed Reality Input. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–11. DOI:http://dx.doi.org/10.1145/3313831.3376313

[58] Xingyi Zhou, Qingfu Wan, Wei Zhang, Xiangyang Xue, and Yichen Wei. 2016. Model-Based Deep Hand Pose Estimation. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI'16)*. AAAI Press, 2421–2427.

[59] Yuxiao Zhou, Marc Habermann, Weipeng Xu, Ikhsanul Habibie, Christian Theobalt, and Feng Xu. 2020a. Monocular Real-time Hand Shape and Motion Capture using Multi-modal Data. (2020).