



Robust vision-based glove pose estimation for both hands in virtual reality

Fu-Song Hsu¹ · Te-Mei Wang² · Liang-Hsun Chen³

Received: 2 February 2023 / Accepted: 21 August 2023 / Published online: 15 September 2023
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2023

Abstract

In virtual reality (VR) applications, haptic gloves provide feedback and more direct control than bare hands do. Most VR gloves contain flex and inertial measurement sensors for tracking the finger joints of a single hand; however, they lack a mechanism for tracking two-hand interactions. In this paper, a vision-based method is proposed for improved two-handed glove tracking. The proposed method requires only one camera attached to a VR headset. A photorealistic glove data generation framework was established to synthesize large quantities of training data for identifying the left, right, or both gloves in images with complex backgrounds. We also incorporated the “glove pose hypothesis” in the training stage, in which spatial cues regarding relative joint positions were exploited for accurately predict glove positions under severe self-occlusion or motion blur. In our experiments, a system based on the proposed method achieved an accuracy of 94.06% on a validation set and achieved high-speed tracking at 65 fps on a consumer graphics processing unit.

Keywords Glove tracking · Glove dataset · Hand tracking · Vision-based tracking · Hand pose estimation · Haptic glove

1 Introduction

Virtual reality (VR) and augmented reality (AR) are technologies that extend the sensory environment of an individual by enhancing reality with technology. Advances in technology have enabled the achievement of immersive experiences, which may play a key role in future communication. Studies have emphasized the importance of hand tracking within VR environments. Voigt-Antons et al. (2020) compared the user experience of controllers with and without hand tracking ability in VR environments. Participants experienced greater presence when using a controller with hand tracking

ability than when using controllers without this ability in VR environments. They described performing physical tasks in a VR scenario with hand tracking as more realistic than performing physical tasks in a VR scenario without hand tracking. Moreover, two-hand interaction can considerably enhance participant understanding of a VR environment. Hinckley et al. (1998a, b) conducted a series of experiments regarding the advantages of two-handed VR user interfaces and demonstrated that two-handed control is not only faster than one-handed control but also provides more information about the VR scene. Ultimately, manipulating objects with two-hand tracking increases VR user engagement.

For VR or AR users, using two-handed gestures for interacting with other users or objects is more enjoyable and convenient than is using one-handed gestures and enables superior task performance (Kotranza et al. 2006). Both early and recent studies of human–computer interaction have reported that bimanual input improves performance for various tasks by enabling users to perform subtasks with separate hands, such as changing gears while driving, page turning while reading a book (Buxton 1995; Buxton and Myers 1986), and interacting with multiple home devices simultaneously (Vogiatzidakis and Koutsabasis 2022). In visual control tasks, the use of both hands can provide users with information that cannot be obtained from a single hand,

✉ Fu-Song Hsu
fshsu@nycu.edu.tw

Te-Mei Wang
TeMeiWang@itri.org.tw

Liang-Hsun Chen
bill5254.cs05@nycu.edu.tw

¹ Institute of Communication Studies, National Yang Ming Chiao Tung University, Hsinchu, Taiwan

² Electronic and Optoelectronic System Research Laboratories, Industrial Technology Research Institute, Hsinchu, Taiwan

³ Institute of Multimedia Engineering, National Yang Ming Chiao Tung University, Hsinchu, Taiwan

thereby allowing users to shift their attention to secondary tasks (Hinckley et al. 1997).

In this paper, we focus on two-hand manipulation in VR, AR, and mixed reality (MR). A VR glove is a common VR accessory that collects real-time gesture, position, and orientation information about a user's hand. The numerous sensors of a VR glove generate a large quantity of data, and VR gloves are more complex and expensive than are many other input devices. In this paper, a convenient and low-cost method of implementing a VR glove is proposed in which glove motions are captured and tracked using a camera. The proposed method is effective for natural, two-handed motions.

Hand pose tracking is a critical technology in various VR and AR applications, such as human–computer interactions, smart homes, rehabilitation, and gaming. However, hand pose tracking is more difficult to achieve than is full-body tracking because of several factors (Erol et al. 2007). Some of these factors are as follows: (1) free viewpoints: the changing view angle in hand pose tracking increases the challenge of classifying hand gestures; (2) complex articulation: hands have many highly flexible joints that can perform numerous complex hand gestures; (3) severe self-occlusion: fingers of the same hand often completely occlude one another; and (4) finger similarity: fingers are extremely similar and are difficult to distinguish.

Most previous methods for hand pose tracking have focused on single hand pose estimation (Mueller et al. 2017; Mueller et al. 2018; Garcia-Hernando et al. 2018; Moon et al. 2018; Xiong et al. 2019; Fang et al. 2020; Cheng et al. 2021). However, in many real applications, using both hands for interactions is preferable (Buxton and Myers 1986; Buxton 1995; Hinckley et al. 1998a, b; Kotranza et al. 2006; Voigt-Antons et al. 2020; Vogiatzidakis and Koutsabasis 2022). For users of VR or AR applications, interacting with other users or objects by using two-hand gestures is appealing and convenient. Two-hand pose estimation does not simply involve performing single hand pose estimation twice; two-hand pose estimation has unique challenges. First, handling occlusions in two-hand manipulation are challenging because complicated situations, such as the hands crossing over each other, must also be considered. Second, the left and right hand are highly similar; differentiating them is challenging but critical. For example, haptic feedback is a key component of an immersive VR experience; vibration feedback sent to the wrong hand in two-hand pose estimation is immersion-breaking. A user catching a ball with the left hand in a VR sports game but receiving right-hand haptic feedback would be confused.

We focus on two research challenges in VR glove tracking: producing a two-hand glove data set and estimation model training. To produce the data set, a photorealistic glove data generation framework was developed and used

to produce 140,000 training images of two-handed manipulation. Next, we developed a deep learning method to predict glove poses (i.e., the relative locations of the hand joints in terms of distances and angles) to improve performance if severe self-occlusion occurs. The developed system achieved an accuracy of 94.06% with an average error of 7.18 mm on a test set, and it can achieve high-speed motion tracking at 65 frames per second (fps) on a consumer graphics processing unit (GPU) (Fig. 1).

The contributions of this study can be summarized as follows:

- A photorealistic glove data generation framework for synthesizing large quantities of training data of VR gloves was developed.
- A novel training strategy, namely glove pose hypothesis, is proposed for VR glove-tracking models. This method involves the exploitation of spatial cues regarding relative joint positions for accurately predicting glove positions despite occlusion or motion blurring.
- The proposed glove pose estimation method can classify an image as the right glove, left glove, or both gloves. Previously proposed tracking methods cannot differentiate between right and left gloves. Experiments revealed that the proposed method had a high recognition rate and was therefore effective (Fig. 2).

2 Related work

One method of reconstructing hand pose involves using wearable sensors (VR gloves) for data acquisition. VR gloves may contain flex sensors, stretch sensors, inertial measurement units (IMUs), or magnetic sensors. Table 1 lists four commercial VR glove products that contain flex sensors and IMUs. A VR glove uses sensor data to calculate the local joint angles of the hand to reconstruct a hand pose. However, tracking two hands is challenging for VR gloves. Figure 3 presents an example of a straightforward two-hand interaction with VR gloves (Manus VR). The estimated hand positions in the right side of Fig. 3, which are produced by combining the local joint angles of the left and right hands, are clearly unsatisfactory. Current VR gloves are unsuitable for reconstructing hand poses because they can only detect local joint angles and lack a mechanism for handling self-occlusion. Moreover, the accuracy of expensive wearable sensors decreases over time (Chen et al. 2020). By contrast, computer vision methods can obtain holistic data regarding both gloves from only visual input; they may have superior discriminative power for identifying two interacting gloved hands.

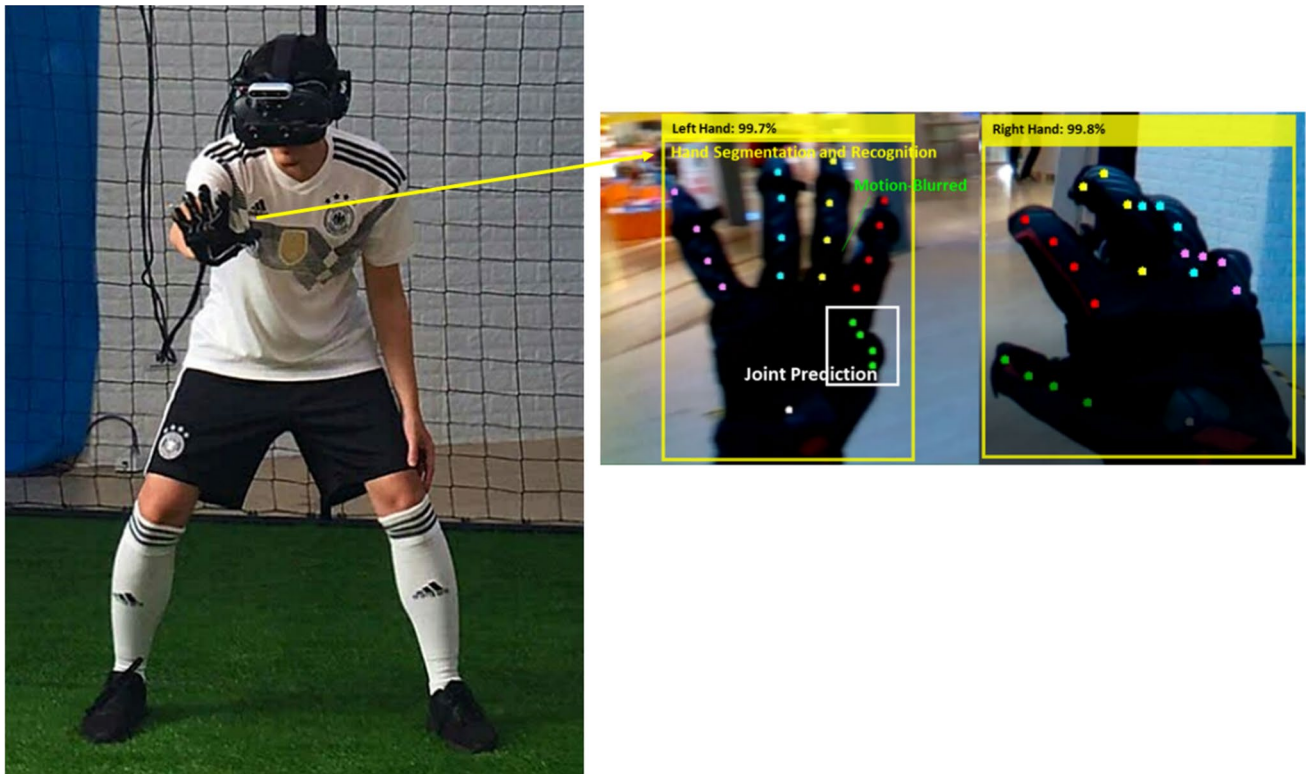


Fig. 1 Three-dimensional pose tracking for two-hand MR gloves. Green, red, yellow, blue, and pink indicate the thumb, index, middle, ring, and little finger joints, respectively (colour figure online)







Fig. 2 Two-hand glove pose estimation. Red and green dots indicate left and right glove joints, respectively (colour figure online)

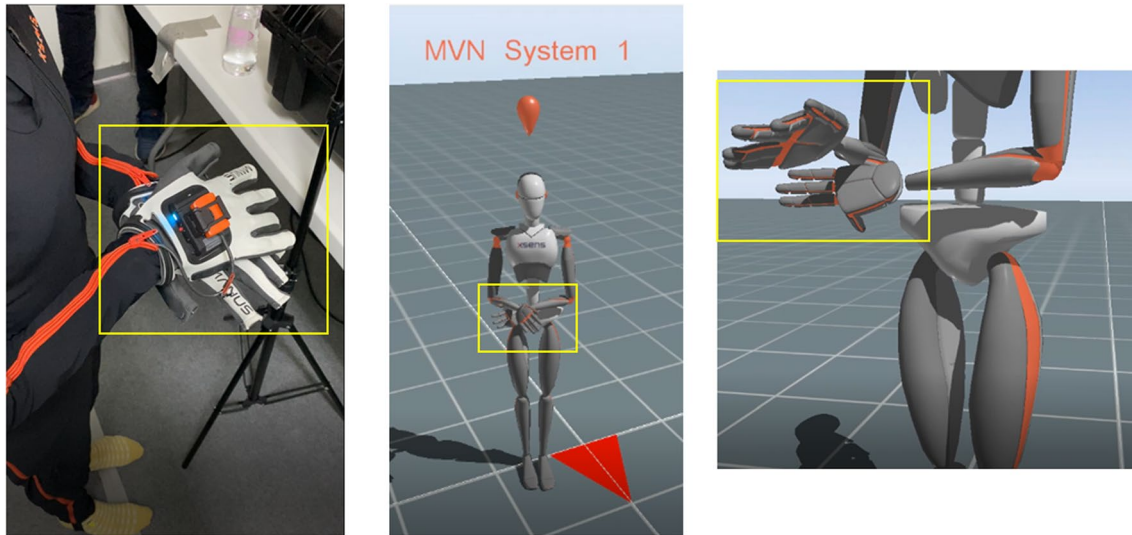
Computer vision methods can be classified as model-based and data-driven methods. Model-based methods (Barron and Kakadiaris 2000; Doosti et al. 2020; Chen et al. 2021) involve assuming a priori constraints for the hand model, such as for the distance distribution between each

pair of hand joints, and then fitting a model to observations by optimizing a developed cost function. Data-driven models use minimum assumptions regarding the hand model and adopt the collected training data to conduct direct mapping from input images to target hand poses. However, most previous methods (Cheng et al. 2021; Fang et al. 2020; Xiong et al. 2019) for hand pose tracking have focused on bare-hand pose estimation. Although some studies have investigated two-hand pose estimation (Lin et al. 2021), these works are not applicable to VR gloves because the appearance of VR gloves differs from that of bare hands and VR gloves have numerous additional functions. Moreover, previous methods cannot differentiate between the left and right gloves, which is challenging but critical for haptic tasks.

Numerous studies have implemented data-driven computer vision methods. Some studies have used monocular red–green–blue (RGB) images (Mueller et al. 2018; Zhao et al. 2021; Spurr et al. 2021; Liu et al. 2021; Yang et al. 2022), whereas others have adopted depth-based methods, in which a single depth map is used for estimation (Tompson et al. 2014; Moon et al. 2018; Xiong et al. 2019; Fang et al. 2020; Ren et al. 2022). Other researchers have used both RGB and depth map information (Rad et al. 2018; Yang et al. 2019), other forms of input data, such as point clouds (Chen et al. 2019; Cheng et al. 2021), or event cameras

Table 1 Information on commercial VR gloves

	Manus VR	CyberGlove	GloveOne	Hi5 VR GLOVE
				
Battery duration	3–6 h of continuous use	No info.	Up to 8 h	> 3 h replaceable battery
Glove size	One size	One size	No info	3 sizes
Weight	68.5 g	70 g + 55 g (6 CyberTouch actuators)	No info	105 g
hand washable	Washable	No info	No info	No info
Communication	2.4 GHz radio frequency (Wireless)	Wire	Blu4.0	2.4 GHz radio frequency (Wireless)
Hand motion tracking	Yes	Yes	Yes	Yes
Sensor sample rate	200 Hz	90 Hz	No info	150 Hz (Output data rate)
Finger sensor accuracy	± 3 degrees	± 3 degrees	No info	± 2 degrees
The amount of sensors	2 IMU 10 Flexible analog sensors	10 Flexible analog sensors	1 IMU 5 Flexible sensors	1 IMU (9-DOF) 7 IMU
Tactile feedback	Yes	Yes	Yes	Yes
The amount of actuators	1 vibration motor inside casing	6 Vibro-tactile actuators	10 Vibro-tactile actuators	1 vibration motor inside casing
Vibrational amplitude	No info	1.2 N peak-to-peak at 125 Hz	No info	No info
Vibrational frequency	No info	0–125 Hz	No info	No info

**Fig. 3** Commercial VR glove tracking two interacting hands

(Rudnev et al. 2021). In general, RGB-only three-dimensional (3D) pose estimation methods have higher average 3D error than do depth-based methods. Table 2 reveals that most current vision-based methods are depth-based methods because these methods provide rich 3D information and are robust to different illumination conditions. These factors

help a system based on computer vision methods to distinguish foreground target hands from unrelated background objects. Therefore, to maximize the accuracy and efficiency of our method, we selected an RGB-Depth (RGB-D)-based method in which an RGB camera is used to track hand joints and a depth camera is used to measure the distance to joints.

Table 2 Comparison between computer vision-based methods

	Avg.3D error (mm.)	Sensor	Tracking
Lin et al. (2021)	17.42	RGB	Two-hand tracking (bare hand)
Cheng et al. (2021)	8.58	Depth	Single hand tracking (bare hand)
Fang et al. (2020)	8.29	Depth	
Xiong et al. (2019)	8.61	Depth	
Moon et al. (2018)	8.42	Depth	
Garcia-Hernando et al. (2018)	n/a (uses the accuracy 87.45%)	RGB, depth	
Mueller et al. (2018)	50	RGB	
Mueller et al. (2017)	32.6	RGB, depth	

3 Methods

3.1 Two-hand glove data set

Studies have investigated single- or bare-hand tracking methods; however, no data set of two-hand glove poses exists. Therefore, we developed a photorealistic glove data generation framework that produces synthetic hand pose data to train our network (Fig. 4). First, we constructed a prototype of two-hand VR gloves with haptic actuators (Fig. 5). Each glove fingertip has a haptic eccentric rotating mass or linear resonant actuator driven by a DRV2603 haptic driver. Adafruit Feather Nrf52 Bluefruit, which is an all-in-one Bluetooth low-energy development board, was used to interface with the gloves. White wires connecting the controller and haptic actuators were attached on the outside of the glove. 3D hand models of the left and right gloves were constructed using the Unity software, and realistic textures captured from the prototype gloves were applied to the models. These models were used to produce the two-hand glove training images, which comprised images of left or right gloves forming six signs representing the numbers from 0 to 5 from various

angles (Fig. 6). Background augmentation was applied to increase tracking robustness. We downloaded 10,000 images of indoor scenes and pasted the two-hand glove images onto these images to ensure that tracking was stable with diverse background environments. Moreover, the position of the virtual camera was rotated and translated to the egocentric view with respect to the VR headset. The virtual camera parameters were based on those of the VR head-mounted motion camera to ensure that the virtual camera had the same distortion as the real camera.

To accelerate training but ensure accuracy in complex real environments, we first trained an initial model by using the synthetic training images and then used transfer learning to train a final model in the complex real environment. Initial training was performed using a set of 140,000 synthetic data; the final transfer learning model was then trained using 1800 images captured in real environments. Therefore, our two-hand glove data set comprises mostly synthetic hand data (~90%) and a small quantity of real data (~10%).

The adopted hand model is displayed in Fig. 7. Each hand was assigned 21 points: one for the center of the palm, and one for each finger joint and tip (four per finger). Tip joints

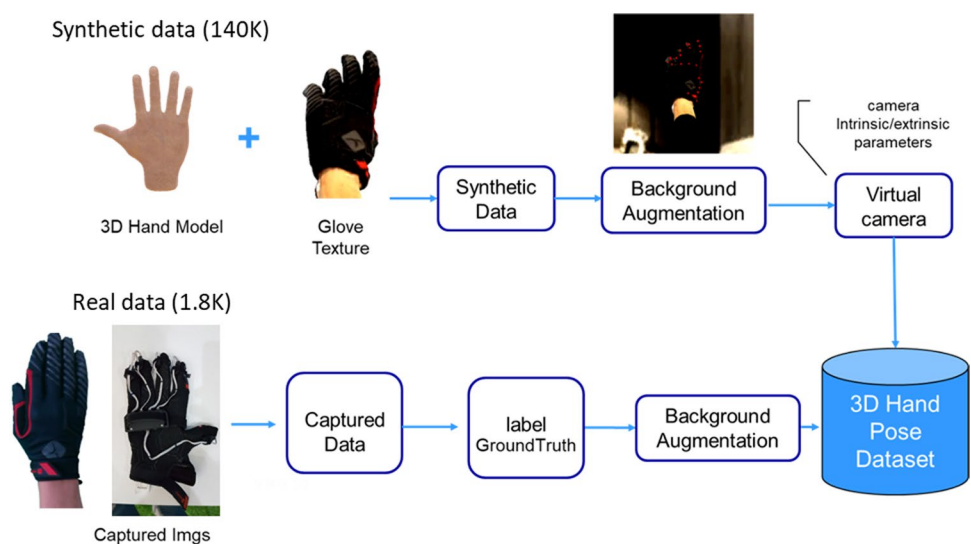
Fig. 4 Two-hand glove data set

Fig. 5 Glove prototypes with haptic actuators. Left, top view; right, bottom view



Fig. 6 Glove prototype and the six sign postures

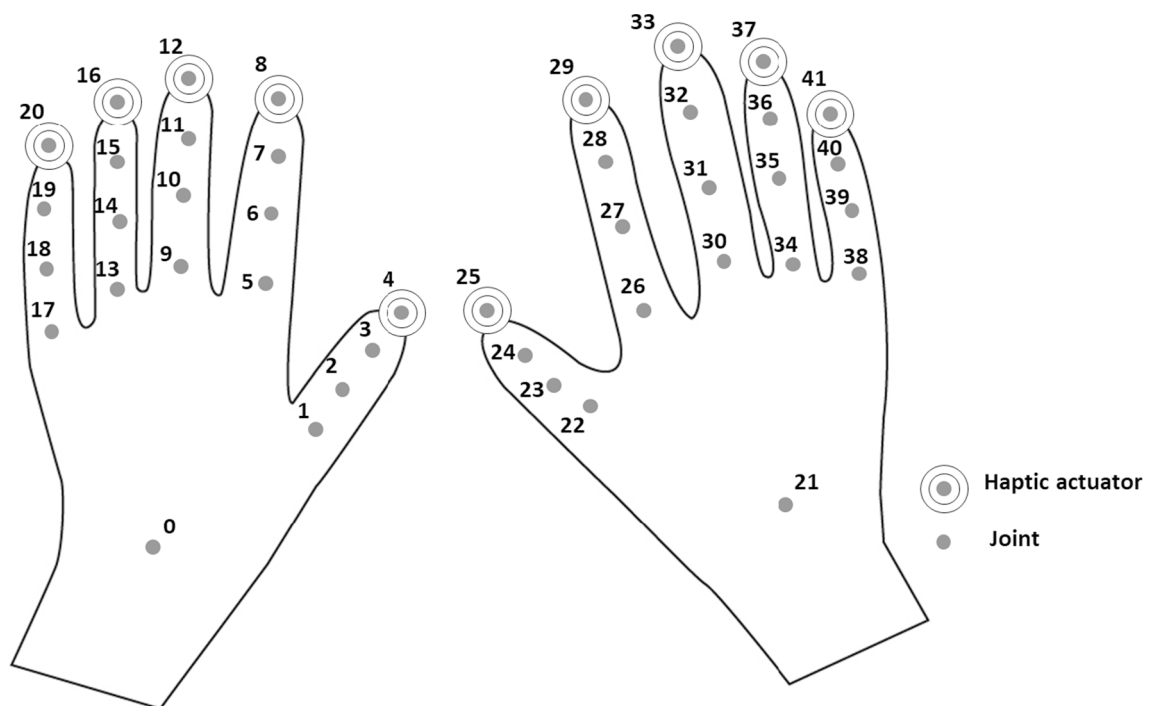
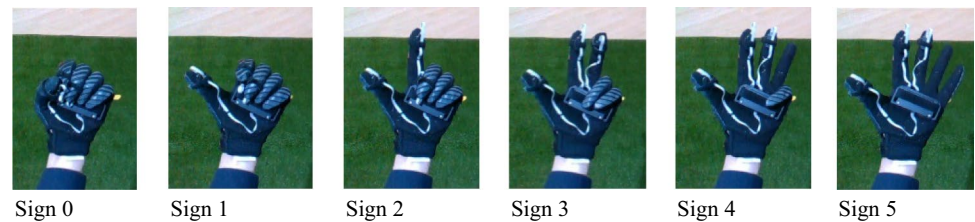


Fig. 7 Two-hand model for our proposed VR glove system

corresponded to the haptic actuators. Across both hands, 42 total joints were tracked.

Our system only requires a lightweight RGB-D sensor. After processing each input frame, the system recognizes and outputs 3D coordinates (x, y, z) for each joint in each hand. However, because of the symmetry of hands, the model is prone to categorize joints as belonging to the incorrect hand if only image texture features are used. This misprediction was avoided by including a joint detection residual neural network (ResNet), which extracts the most discriminative color texture features from the input RGB images to predict the position of an occluded joint on the basis of the positions of unoccluded joints; this prediction is denoted the “glove pose hypothesis.” Relative relationships (such as distances and angles) between joint positions learned from the two-hand glove data set were used to form this hypothesis. Therefore, the positions of joints could be predicted despite occlusion. For example, the proposed method ensures that a joint would be predicted near other joints; a joint would not be predicted in the background, and left-hand joints would be predicted to be located near other left-hand nodes. A flowchart of the proposed system is presented in Fig. 8. The proposed system has an additional, optional binary classifier that uses depth information to increase accuracy in images with complex backgrounds. The details of this function are described in the following sections.

3.2 Joint detector

The input data for the joint detector were RGB color images with a resolution of 640×360 that were captured in real-time by an RGB-D camera; the outputs are the predicted two-dimensional (2D) locations of the 42 joints in the input image. We adopted a network architecture based on ResNet that was proposed by Pishchulin et al. (2016) and Insafutdinov et al. (2016) to train our joint detector by

using stochastic gradient descent on our two-hand glove data set. Figure 9 depicts the architecture of our joint detector network.

The joint detector has two functions: (1) extracting useful color features and computing heat maps (ResNet101) and (2) using the extracted features to predict the 42 joint locations of the two gloves. To improve the prediction accuracy, four transposed convolution layers were used: the joint prediction, location refinement, hand pose hypothesis, and intermediate supervision layers.

3.2.1 Joint prediction

The joint prediction module estimates the two-hand glove pose by predicting the maximum possible 2D positions of the 42 joints as heat maps. Let (x, y) denote the predicted position of the i^{th} joint. The output is as follows:

$$\{(x_i, y_i) \mid i \in \{1, 2, \dots, 42\}\}.$$

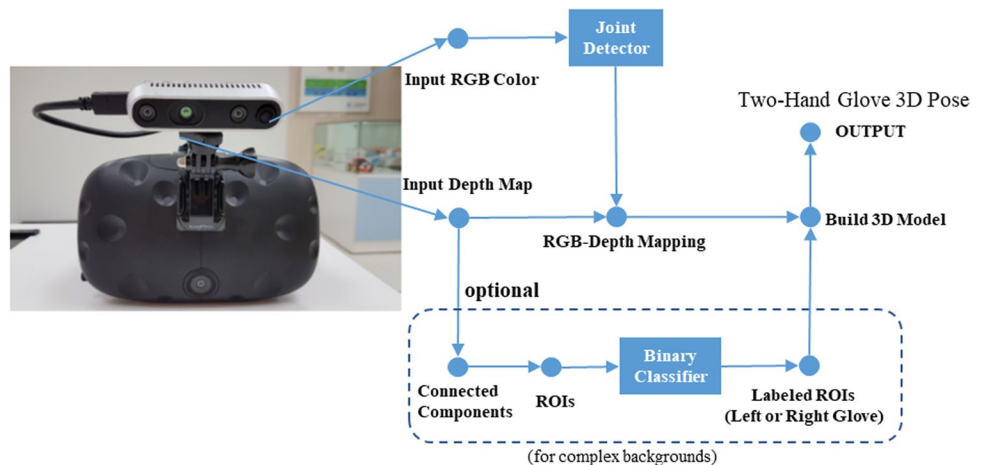
The ground truth (x^t, y^t) is then used to produce the heat maps of the hand joints $\{h_i^t \mid i \in \{1, 2, \dots, 42\}\}$ for training as follows:

$$h_i^t(x, y) = \begin{cases} 1, & \text{if } d((x, y), (x^t, y^t)) \leq c \\ 0, & \text{otherwise} \end{cases}$$

where d denotes the Euclidean distance. Therefore, h_i^t encodes the correct locations (assigned a value of 1) of the i^{th} joint to be predicted. The variable c is a neighborhood radius of the target joint, which is related to the image resolution.

To minimize the distance between the predicted and ground-truth positions, a binary cross-entropy loss function is used to measure the error between a predicted heat map $h_i(x, y)$ and training heat map $h_i^t(x, y)$. Specifically, the loss for the location of the i^{th} joint location is expressed as follows:

Fig. 8 Flowchart of the proposed glove-tracking system



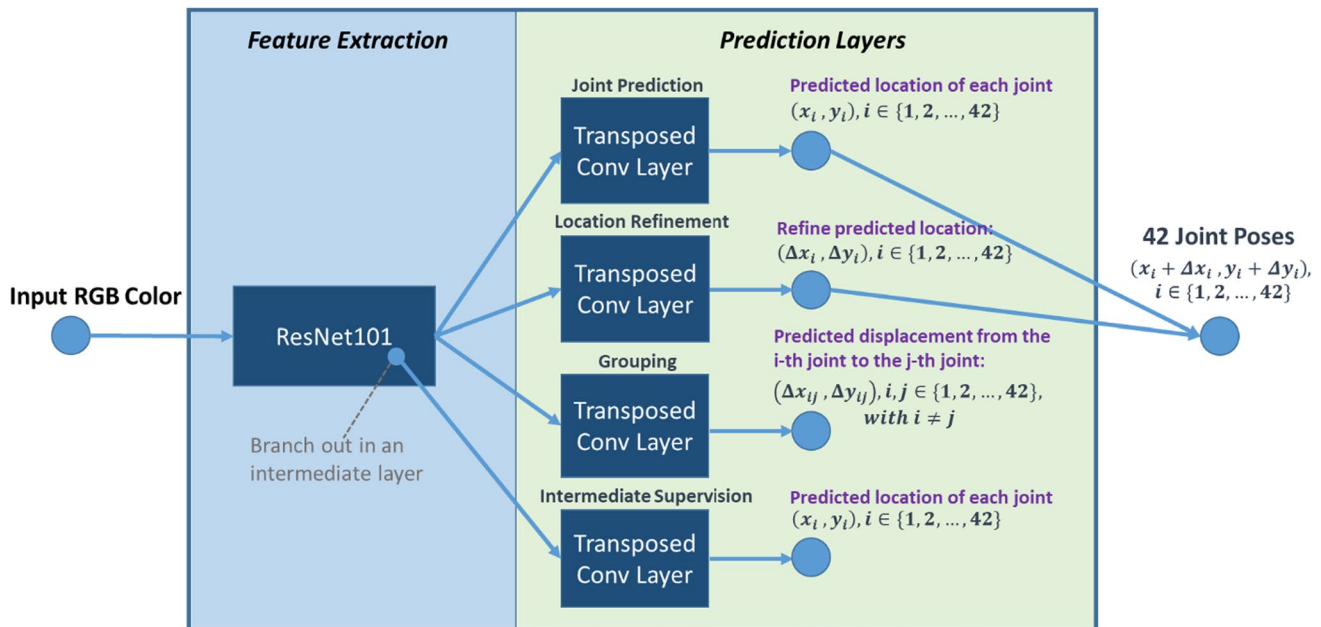


Fig. 9 Network architecture of our joint detector

$$L_i = \sum_{(x,y) \in \mathbb{Z}^2} -h_i^t(x,y) \log h_i(x,y) - (1 - h_i^t(x,y)) \log(1 - h_i(x,y))$$

Thus, the total loss of the joint prediction is expressed as follows:

$$\text{Total loss} = \sum_i L_i.$$

3.2.2 Location refinement

The location refinement module obtains the possible deviations of the 42 joints from their heat map location. Let $(\Delta x_i, \Delta y_i)$ denote the predicted deviation of the i^{th} joint compared with its heat map location. The output of this joint prediction is expressed as follows:

$$\{(\Delta x_i, \Delta y_i) \mid i \in \{1, 2, \dots, 42\}\}.$$

To find the precise predicted location, the following problem is solved:

$$\{(x_i + \Delta x_i, y_i + \Delta y_i) \mid i \in \{1, 2, \dots, 42\}\}.$$

This joint prediction produces 42 pairs of prediction outputs denoted as $\{f_i^t \mid i \in \{1, 2, \dots, 42\}\}$. As in the joint prediction module, the ground truth $\{(x_i^t, y_i^t) \mid i \in \{1, 2, \dots, 42\}\}$ is used to produce feature maps for training. The i^{th} pair of training feature maps is then set as follows:

$$f_i^t(x, y) = (x_i^t - x, y_i^t - y)$$

The deviation value is assigned to $f_i^t(x, y)$, which indicates the difference $(\Delta x, \Delta y)$ between the predicted location and the ground-truth location. In this joint prediction, the Huber loss (Huber 1992) is the loss function that measures the average error between the predicted output feature maps $f_i(x, y)$ and the training feature maps $f_i^t(x, y)$. The Huber loss is quadratic for small errors and linear for large errors. For a target value t and a predicted value s , the Huber loss is defined as follows:

$$L(t, s) = \begin{cases} (t - s)^2 & \text{for } |t - s| \leq \delta \\ 2\delta |t - s| - \delta^2 & \text{otherwise} \end{cases}$$

where δ is a user-defined constant. If $|t - s| \leq \delta$, the Huber loss is equivalent to measuring the squared error; otherwise, this loss is linear with a slope of δ .

3.2.3 Grouping

The grouping module predicts the positions of occluded joints on the basis of the positions of unoccluded joints by applying the glove pose hypothesis, as discussed in Sect. 3.1. The module uses the relative relationship (such as a distance and angle) between known joint positions learned from the two-hand glove database to ensure that predicted joints appear near joints on the correct hand and not in background locations. In particular, the left–right symmetry of hands often causes nodes to be predicted on the incorrect hand if only image texture features are used; by grouping nodes as part of the left or right hand, this problem can be avoided.

Let $(\Delta x_{ij}, \Delta y_{ij})$ denote the predicted position of an occluded joint j based on the position of a visible joint i with $\{(x_i, y_i) \mid i \in \{1, 2, \dots, 42\}\}$.

The predicted relative relationship between joint i and joint j is then used to predict the position of joint j . For example, the predicted position of joint 1 from the joint prediction module can be input into this joint prediction to output all the predicted locations of the 42 joints as follows:

$$\{(x_1, y_1)\} \cup \{(x_1 + \Delta x_{1j}, y_1 + \Delta y_{1j}) \mid j \in \{2, \dots, 42\}\}.$$

This joint prediction involves the use of 42×41 pairs of relationship maps learned from the database; these pairs are collectively denoted as g indexed by i and j .

$$\{g_{ij}^t \mid i \in \{1, 2, \dots, 42\}, j \in \{1, 2, \dots, 42\}, \text{with } i \neq j\}.$$

First, g_{ij} , which is a translation vector learned from the known position of joint i , is applied to the predicted position of joint j . The ground-truth positions of the 42 joints $\{(x_i^t, y_i^t) \mid i \in \{1, 2, \dots, 42\}\}$ to produce $g_{ij}^t(x, y)$, which is the pair of ground-truth relationship maps indexed by i and j , are given as follows:

$$g_{ij}^t(x, y) = (x_j^t - x, y_j^t - y)$$

The aforementioned positions do not depend on the index i but only on (x, y) and the index j . Given a predicted position (x_i, y_i) of the i th joint, the positions of the remaining 41 joints indexed by j are expressed as follows:

$$(x_j, y_j) = (x_i, y_i) + g_{ij}(x_i, y_i), \quad j \in \{1, 2, \dots, 42\} \text{ with } j \neq i.$$

To minimize the distance between learned relationship maps (g_{ij}) and ground-truth relationship maps (g_{ij}^t), the loss function of the grouping module is the Huber loss, which measures the average error between g_{ij} and g_{ij}^t . This loss function is included in model training; if the best identified solution violates the relationship, the solution is penalized to ensure that the ultimate best solution respects the joint relationship.

3.2.4 Intermediate supervision

The intermediate supervision module uses the features extracted in the intermediate layer to predict the locations of the 42 hand joints as an alternative prediction to that of the output layer. This training objective is included to address the gradient vanishing problem for deep networks. This method was experimentally demonstrated to improve

network training by Wei et al. (2016). The output of this joint prediction is the predicted positions of the 42 joints and has the same form as the output of the joint prediction branch.

$$\{(x_i, y_i) \mid i \in \{1, 2, \dots, 42\}\}.$$

The ground truth and loss function used in the intermediate supervision module are identical to those used in the joint prediction module.

After training, the outputs from the joint prediction and location refinement modules can be combined to produce the final prediction for a given RGB input image. The coarsely predicted locations of the 42 joints are obtained by taking the argument max for each of the heat maps output from the joint prediction module.

$$\left\{ (x_i, y_i) \mid (x_i, y_i) = \underset{(x,y) \in \mathbb{Z}^2}{\operatorname{argmax}} h_i(x, y), i \in \{1, 2, \dots, 42\} \right\}.$$

By incorporating a 2D translation vector $f_i(x_i, y_i) = (\Delta x_i, \Delta y_i)$ produced by the location refinement module into the corresponding coarsely predicted location (x_i, y_i) produced by the joint prediction module for each of the 42 joints indexed by i , the final predicted locations of the 42 joints can be obtained as follows:

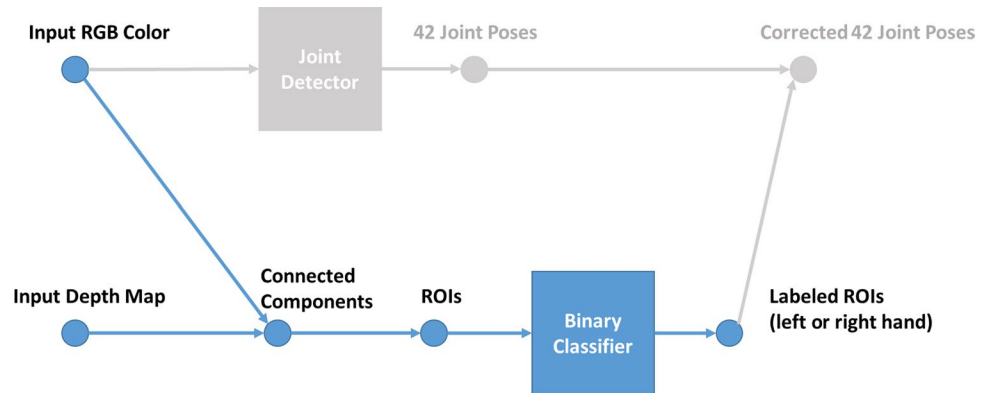
$$\{(x_i + \Delta x_i, y_i + \Delta y_i) \mid i \in \{1, 2, \dots, 42\}\}.$$

The grouping and intermediate supervision modules are only used during training for regularization; they do not participate in the predictions of the trained model.

3.3 Binary classifier

Perfect extraction of a target from a complex environments in 2D camera space is challenging, especially in real-time. To increase the robustness of the aforementioned joint detection method, an auxiliary binary classifier was developed. This classifier uses depth information to classify pixels as background or glove pixels. The hand gloves are assumed to be closer to the camera than are other objects; thus, the binary classifier can apply a threshold to the depth information to extract region of interests (ROIs) that contain the gloves, thereby removing the background. The framework of the developed binary classifier is displayed in Fig. 10. This classifier uses the following steps to obtain ROIs.

Fig. 10 Flowchart of the developed binary classifier



3.3.1 Obtaining the input depth map

The depth map of the current frame is obtained from the RGB-D camera.

3.3.2 Finding maximally connected components

Pixels outside a predefined depth range (200–650 mm) are excluded from the image, and maximally connected components (pixel areas) are computed from the remaining pixels.

3.3.3 Obtaining ROIs

Connected components with an area smaller than a predefined threshold are excluded, and the remaining connected components are the hand ROIs.

3.3.4 ROI classifier training

A data set for training the binary classifier was produced by recording multiple videos of various left- and right-hand gestures with real backgrounds. Approximately 30,000 images containing left- and right-hand regions were extracted from the videos. The classifier was then trained using these images.

3.3.5 Postprocessing

The performance of the developed binary classifier was improved by applying two reasonable assumptions. (1) Spatial distribution: A user has only one left or right hand; unreasonable classification results should be rejected, such as labeling both ROIs as the left hand. (2) Temporal distribution: After processing a frame, the location and predicted label of each ROI are stored. The ROIs obtained for the next frame are matched to the ROIs of the previous frame. ROIs with similar locations and sizes are selected as matched pairs. The classification result of the previous ROI is inherited by the ROI in the current frame. This method ensures

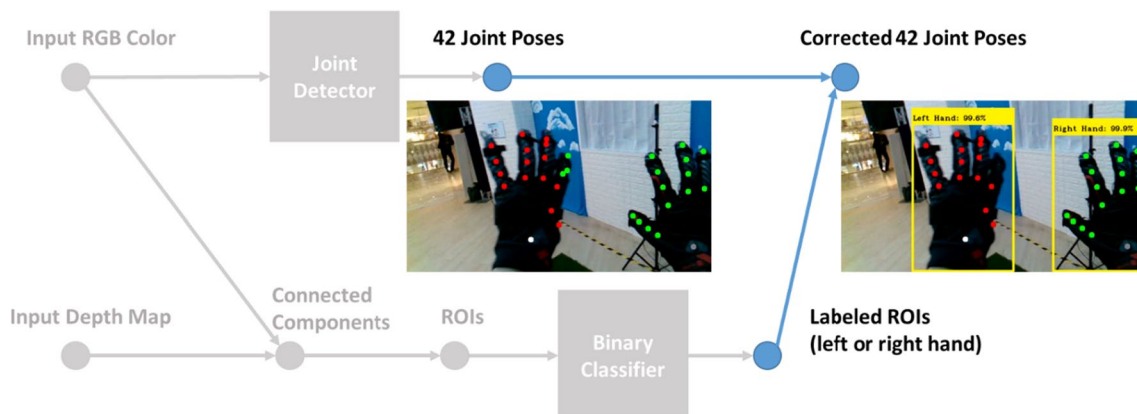


Fig. 11 Joint detector correction by using binary classification

Table 3 Specifications of the hardware used in the experiments

Motion sensor	Intel RealSense D435 camera
Head-mounted display	HTC VIVE
VIVE Ready computer	CPU: Intel i9-9900 K GPU: GeForce RTX 2080Ti

temporal consistency of the ROIs. Only nonmatching ROIs must be input into the network for classification.

3.4 Joint detector with a binary classifier

After obtaining the left- and right-hand regions from the binary classifier, the locations of the 42 joints predicted by the joint detector can be verified and updated (Fig. 11). The left- and right-hand bounding boxes only permit joints for the corresponding hand, thereby enabling the predicted right-hand points (green) on the left hand to be correctly reassigned.

4 Experiments

A series of experiments were conducted to evaluate the proposed method. Experiments were first conducted on the validation set of our training data. Both 2D and 3D metrics were used for quantitatively measuring the glove pose estimation accuracy of the proposed method. In particular, we investigated the effects of the glove pose hypothesis method and various convolutional neural network architectures. Experiments were performed using the HTC VIVE VR headset and a VIVE Ready computer. The hardware specifications are listed in detail in Table 3. Moreover, experiments were performed on another publicly data set to demonstrate the generalizability of the proposed method.

4.1 Evaluation metrics and results

The positions of the VR gloves in the validation set were labeled manually. The validation set comprised 636 images of real scenes that were not part of the training set. The 2D and 3D Euclidean distances were used to quantify the accuracy of the joint detector for predicting hand joints in the validation set. For the i th joint, the 2D distance from the predicted location (x_i, y_i) to the ground truth $(x_i^{(t)}, y_i^{(t)})$ in the image is expressed as follows:

$$d_i = \sqrt{\left(x_i^{(t)} - x_i\right)^2 + \left(y_i^{(t)} - y_i\right)^2}$$

where d_i is the distance in pixels. Similarly, the 3D distance from a predicted coordinate (x_i, y_i, z_i) to its ground truth $(x_i^{(t)}, y_i^{(t)}, z_i^{(t)})$ is expressed as follows:

$$d_i = \sqrt{\left(x_i^{(t)} - x_i\right)^2 + \left(y_i^{(t)} - y_i\right)^2 + \left(z_i^{(t)} - z_i\right)^2}$$

where d_i is the distance in 3D space in millimeters and z_i is the depth value obtained from the RGB-D camera. We adopted the percentage of correct keypoints (PCK), which is a commonly used metric, for evaluating pose estimation accuracy (Sapp and Taskar 2013; Tompson et al. 2014; Rhodin et al. 2016). Accuracy is defined as the ratio of the number of correct predictions to the total number of predictions. The effectiveness of the glove pose hypothesis method was evaluated by comparing two joint detectors with the same network architecture (ResNet101) but with or without the grouping loss function; the results are presented in Table 4. The inclusion of the grouping loss function as a training objective clearly improved the accuracy and error in the 2D and 3D metrics. Figure 12 indicates that the glove pose hypothesis enables the incorrectly assigned left-hand points (green) to be correctly assigned to the left hand. We also investigated the performance of the proposed method for a user making a fist. Figure 13a–c depicts gloved fists at various angles, Fig. 13c and d displays fists at the same angle but for users with different hand sizes, and Fig. 13e and f shows fists with motion blur. The proposed method is effective even for the challenging fist posture and with blur. This result is encouraging and indicates that the hand model joint prediction based on glove pose hypothesis was correctly determined from the 42×41 pairs of relationship maps learned from the two-hand glove database.

The computational performance of the proposed method is suitable for real-time applications. Table 5 presents the performance of two joint detectors with ResNet50 and ResNet101, respectively. ResNet50 has considerably lower complexity than does ResNet101 while maintaining high accuracy; thus, ResNet50 was selected in the developed system. The average endpoint error (EPE) and the area under the curve (AUC) values of each method are presented in Table 6; using the grouping module improves these values. Moreover, the grouping approach is effective regardless of the setting or backbone; performance is improved for both ResNet50 or ResNet101.

Table 4 Results obtained without and with the grouping loss function

	Non-grouping	Grouping
2D (acc./pixel)	93.00%/10.02	93.49%/8.32
3D (acc./mm)	93.12%/8.36	94.06%/7.18

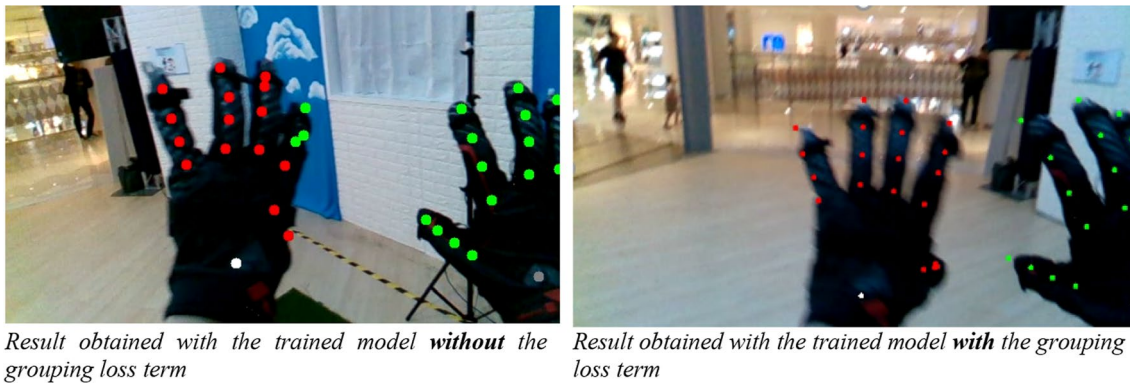


Fig. 12 Results obtained with the training models with and without the grouping loss term

4.2 Binary classifier

Figure 14 presents the qualitative results obtained with the developed binary classifier. The classifier was effective

Table 5 Results obtained with ResNet50 and ResNet101

	ResNet50	ResNet101
2D (acc./pixel)	93.44%/9.38	93.49%/8.32
3D (acc./mm)	93.34%/7.93	94.06%/7.18
Computation performance	65 fps	50 fps

for glove images in challenging real-world scenes, namely scenes in which both hands are directly in front of the camera [the most common case; Fig. 14a]; self-occluding gestures exist [Fig. 14b]; both hands are almost out of the frame [Fig. 14c]; both hands are crossed, and severe self-occlusion occurs [Fig. 14d]; both arms are crossed, and reversed hand positions occur [Fig. 14e]; and only a single hand exists [Fig. 14f]. The results reveal that the developed model is robust even for challenging situations, such as for those depicted in Fig. 14c and e.

Proposed ROIs are only accepted by the binary classifier if its confidence (between 0 and 1) is > 0.99 . The performance

Fig. 13 Glove pose estimation for fists. The red and green dots indicate left and right glove joints, respectively



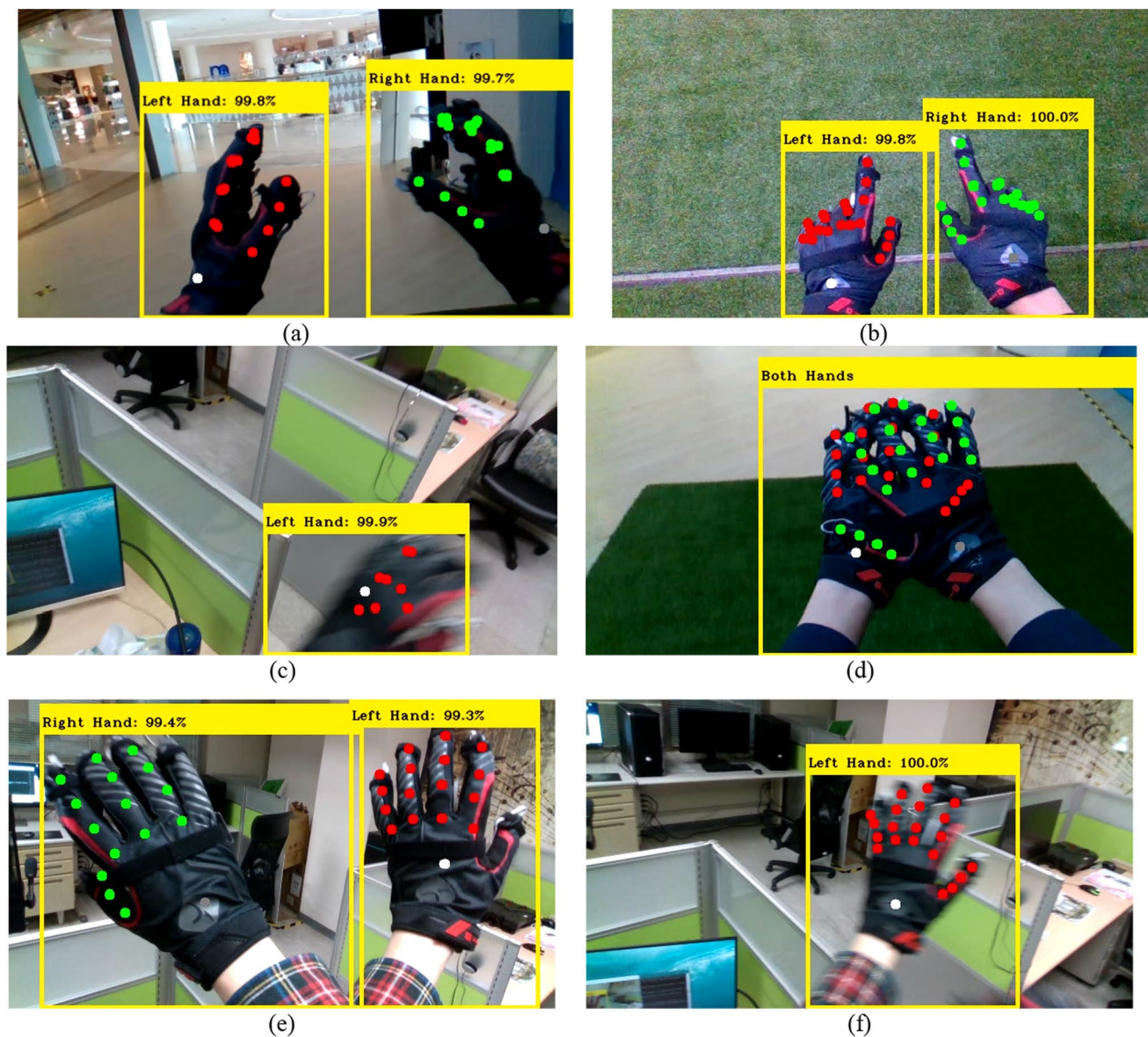


Fig. 14 Qualitative results for VR glove pose estimation with real backgrounds

of the developed system was stable for a variety of backgrounds. Table 7 presents the quantitative results obtained on the validation data set. The results indicate that the deep

Table 6 Comparison of EPE and AUC for ResNet50 and ResNet101 with and without grouping

Method	2D		3D	
	EPE↓	AUC↑	EPE↓	AUC↑
ResNet50	9.92	0.825	8.31	0.845
ResNet50 + grouping	9.38	0.827	7.93	0.847
ResNet101	10.02	0.827	8.36	0.846
ResNet101 + grouping	8.32	0.831	7.18	0.852

The bold represents the value of "+grouping"

learning model (ResNet) achieves an error rate of 14.47%; ResNet with the proposed glove pose hypothesis and binary classifier modules achieves a substantially higher prediction accuracy, with the mean error rate being only 3.6%.

Table 7 Quantitative results for left- and right-hand joint classification

Method	Error rate (%)
ResNet	14.47
ResNet + Glove pose hypothesis	12.42
ResNet + Glove pose hypothesis + Binary classifier	3.6

Table 8 Computation performance of the proposed approach with and without the developed binary classifier

Method	Average FPS
Without binary classifier	65
With binary classifier	63

Table 9 Results obtained with the proposed model for the Ego3D data set

	RGB	RGB-D
2D(acc./pixel)	92.82%/9.45	96.34%/6.87
3D (acc./mm)	–	91.20%/10.27

4.3 Computational performance

The computational performance of the proposed method with and without the binary classifier was compared in terms of the fps value achieved on a system with the hardware specifications listed in Table 3. Two minutes of testing in the environment was performed for each method; the first 20 frames during system startup were not included. The results are presented in Table 8. The binary classifier requires little additional computing resources because binary classification is only performed if temporally consistent matches cannot be identified.

4.4 Results on the Ego3D data set

For further validating the proposed approach, experiments were performed on the Ego3D data set of bare-hand images (Lin et al. 2021). Previous relevant studies have only performed bare-hand pose estimation. To the best of our knowledge, no public data set exists for VR glove pose estimation. Ego3D is an egocentric view, synthetic bare-hand data set that contains 50,000 training and 5000 testing images in the RGB and RGB-D formats with joint annotations. The proposed model was tested for the RGB-only and RGB-D data sets to determine the effect of including depth information.

ResNet accepts RGB color image by default; therefore, we inserted an additional 7×7 convolution layer to transform four-channel RGB-D images into three-channel feature maps before the images were input into the original ResNet. Moreover, min–max normalization was performed; red, green, and blue color values of 0–255 and depth values ranging from 0 to > 100 were all mapped to $[0, 1]$.

Table 9 reveals that, as in the experiments on our data set, the proposed approach also achieved high accuracy on Ego3D. We adopted the 2D and 3D Euclidean distance metrics (Sect. 4.1) to evaluate the model performance for 2D and 3D images, respectively. The results for the RGB-D images were

considerably better than those for the RGB images. RGB-D images are superior to RGB images for extracting hand poses in complex backgrounds. The developed binary classifier applies this fact to obtain ROIs effectively from depth maps.

5 Conclusion

Two-hand motion tracking is a critical technique for VR interactions. Tracking hand movements is more challenging than is tracking body movements because of the free, egocentric viewpoint, complex hand articulation, self-occlusion, and finger similarity involved when tracking hand movements; tracking movements of two hands simultaneously is even more difficult. Most VR gloves use expensive flex and IMU sensors for hand tracking; however, these sensors only provide local joint angles. In this study, cameras were used for simultaneously tracking two-hand VR glove motions. The vision-based method can obtain more holistic information regarding hand movements and has better discriminative power for two-hand manipulation than do other relevant methods.

The main contributions of this study are as follows. First, a residual network based on a computer vision method was developed to achieve effective hand pose estimation and ensure hand coherence. Second, a two-hand data set containing 140,000 synthetic images and 1800 real images of hands wearing VR gloves was produced for model training. This data set can be used for other VR and AR applications. Third, a training objective called “grouping” was introduced into the developed model for effectively handling self-occlusions. The efficacy of the adopted grouping loss function was demonstrated experimentally. The developed system achieved an accuracy of 94.06% on our validation set and high-speed motion tracking at 65 fps on a consumer GPU. Experiments on another publicly available data set verified the accuracy, efficacy, and generalizability of the proposed approach.

In the future, we intend to combine the vision-based method with local joint angle data from commercial VR gloves to improve the accuracy of the proposed method. Moreover, we intend to test the proposed method in additional real environments to further verify its performance.

Supplementary Information The online version contains supplementary material available at <https://doi.org/10.1007/s10055-023-00860-6>.

Acknowledgements This study was supported by the Industrial Technology Research Institute, the National Science and Technology Council, Taiwan (Grant Numbers: NSTC 111-2222-E-A49-008 and NSTC 112-2221-E-A49-129).

Data availability The authors confirm that the data supporting the findings of this study are available within the article.

Declarations

Conflict of interest The authors have no relevant financial or nonfinancial interests to disclose.

References

- Barron C, Kakadiaris IA (2000) Estimating anthropometry and pose from a single image. *Proc IEEE Conf Comput vis Pattern Recognit* 1:669–676. <https://doi.org/10.1109/CVPR.2000.855884>
- Buxton W, Myers B (1986) A study in two-handed input. In: *Proceedings of the SIGCHI conference on human factors in computing systems*, Boston, Massachusetts, USA., 321–326. <https://doi.org/10.1145/22627.22390>
- Buxton W (1995) Chunking and phrasing and the design of human-computer dialogues. In: Baecker RM, Grudin J, Buxton WAS, Greenberg S. (Eds), *Readings in human-computer interaction*, 494–499. <https://doi.org/10.1016/B978-0-08-051574-8.50051-0>
- Chen W, Yu C, Tu C, Lyu Z, Tang J, Ou S, Fu Y, Xue Z (2020) A Survey on hand pose estimation with wearable sensors and computer-vision-based methods. *Sensors* 20(4):1074. <https://doi.org/10.3390/s20041074>
- Chen Y, Tu Z, Ge L, Zhang D, Chen R, Yuan J (2019) SO-HandNet: self-organizing network for 3D hand pose estimation with semi-supervised learning. In: *Proceedings of the IEEE/CVF international conference on computer vision*, 6961–6970
- Chen Y, Tu Z, Kang D, Bao L, Zhang Y, Zhe X, Chen R, Yuan J (2021) Model-based 3D Hand Reconstruction via Self-Supervised Learning. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10451–10460. <https://doi.org/10.48550/arXiv.2103.11703>
- Cheng W, Park JH, Ko JH (2021) HandFoldingNet: A 3D hand pose estimation network using multiscale-feature guided folding of a 2D hand skeleton. In: *Proceedings of the IEEE/CVF international conference on computer vision*, 11260–11269. <https://doi.org/10.48550/arXiv.2108.05545>
- Doosti B, Naha S, Mirbagheri M, Crandall DJ (2020) Hope-net: a graph-based model for hand-object pose estimation. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp 6608–6617. <https://doi.org/10.48550/arXiv.2004.00060>
- Erol A, Bebis G, Nicolescu M, Boyle RD, Twombly X (2007) Vision-based hand pose estimation: a review. *Comput vis Image Underst* 108(1–2):52–73. <https://doi.org/10.1016/j.cviu.2006.10.012>
- Fang L, Liu X, Liu L, Xu H, Kang W (2020) JGR-P2O: Joint graph reasoning based pixel-to-offset prediction network for 3D hand pose estimation from a single depth image. In: *European Conference Computer Vision*, pp 120–137. <https://doi.org/10.48550/arXiv.2007.04646>
- Garcia-Hernando G, Yuan S, Baek S, Kim TK (2018) First-person hand action benchmark with RGB-D videos and 3D hand pose annotations. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 409–419. <https://doi.org/10.48550/arXiv.1704.0246>
- Hinckley K, Pausch R, Proffitt D, Kassell NF (1998a) Two-handed virtual manipulation. *ACM Trans Comput Hum Interact* 5(3):260–302. <https://doi.org/10.1145/292834.292849>
- Hinckley K, Pausch R, Proffitt D, Kassell NF (1998b) Two-handed virtual manipulation. *ACM Trans Comput Hum Interact (TOCHI)* 5(3):260–302. <https://doi.org/10.1145/292834.292849>
- Hinckley K, Pausch R, Proffitt D (1997) Attention and visual feedback: the bimanual frame of reference. In: *Proceedings of the 1997 symposium on interactive 3D graphics*, Providence, Rhode Island, USA. 121–ff. <https://doi.org/10.1145/253284.253318>
- Huber PJ (1992) Robust estimation of a location parameter. In: *Breakthroughs in statistics*, pp 492–518. https://doi.org/10.1007/978-1-4612-4380-9_35
- Insafutdinov E, Pishchulin L, Andres B, Andriluka M, Schiele B (2016) DeeperCut: a deeper, stronger, and faster multi-person pose estimation model. In: *European conference on computer vision*, pp 34–50. <https://doi.org/10.48550/arXiv.1605.03170>
- Kotranza A, Quarles J, Lok B (2006) Mixed reality: are two hands better than one?. In: *Proceedings of the ACM symposium on virtual reality software and technology*, Limassol, Cyprus. pp 31–34. <https://doi.org/10.1145/1180495.1180503>
- Lin F, Wilhelm C, Martinez T (2021) Two-hand global 3D pose estimation using monocular RGB. In: *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp 2373–2381. <https://doi.org/10.48550/arXiv.2006.01320>
- Liu S, Jiang H, Xu J, Liu S, Wang X (2021) Semi-supervised 3D hand-object poses estimation with interactions in time. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp 14687–14697. <https://doi.org/10.48550/arXiv.2106.05266>
- Moon G, Chang JY, Lee KM (2018) V2V-PoseNet: voxel-to-voxel prediction network for accurate 3D hand and human pose estimation from a single depth map. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 5079–5088. <https://doi.org/10.48550/arXiv.1711.07399>
- Mueller F, Mehta D, Sotnychenko O, Sridhar S, Casas D, Theobalt C (2017) Real-time hand tracking under occlusion from an egocentric RGB-D sensor. In: *Proceedings of the IEEE international conference on computer vision*, pp 1154–1163. <https://doi.org/10.48550/arXiv.1704.02201>
- Mueller F, Bernard F, Sotnychenko O, Mehta D, Sridhar S, Casas D, Theobalt C (2018) GANerated hands for real-time 3D hand tracking from monocular RGB. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 49–59. <https://doi.org/10.48550/arXiv.1712.01057>
- Pishchulin L, Insafutdinov E, Tang S, Andres B, Andriluka M, Gehler PV, Schiele B (2016) DeepCut: joint subset partition and labeling for multi person pose estimation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 4929–4937. <https://doi.org/10.48550/arXiv.1511.06645>
- Rad M, Oberweger M, Lepetit V (2018) Feature mapping for learning fast and accurate 3D pose inference from synthetic images. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 4663–4672. <https://doi.org/10.48550/arXiv.1712.03904>
- Ren P, Sun H, Hao J, Wang J, Qi Q, Liao J (2022) Mining multi-view information: a strong self-supervised framework for depth-based 3D hand pose and mesh estimation. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 20555–20565. <https://doi.org/10.1109/CVPR52688.2022.01990>
- Rhodin H, Richardt C, Casas D, Insafutdinov E, Shafiei M, Seidel H-P, Schiele B, Theobalt C (2016) EgoCap: egocentric markerless motion capture with two fisheye cameras. *ACM Trans Grap* 35(6):1–11. <https://doi.org/10.48550/arXiv.1609.07306>
- Rudnev V, Golyanik V, Wang J, Seidel HP, Mueller F, Elgharib M, Theobalt C (2021) Real-time neural 3D hand pose estimation from an event stream. In: *Proceedings of the IEEE/CVF international conference on computer vision*, pp 2385–12395. <https://doi.org/10.48550/arXiv.2012.06475>
- Sapp B, Taskar B (2013) MODEC: multimodal decomposable models for human pose estimation. *IEEE Conf Comput vis Pattern Recognit* 2013:23–28. <https://doi.org/10.1109/CVPR.2013.471>

- Spurr A, Dahiya A, Wang X, Zhang X, Hilliges O (2021) Self-supervised 3D hand pose estimation from monocular RGB via contrastive learning. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 11230–11239. <https://doi.org/10.48550/arXiv.2106.05953>
- Tompson J, Stein M, Lecun Y, Perlin K (2014) Real-time continuous pose recovery of human hands using convolutional networks. *ACM Trans Grap* 33(5):1–10. <https://doi.org/10.1145/2629500>
- Vogiatzidakis P, Koutsabasis P (2022) ‘Address and command’: two-handed mid-air interactions with multiple home devices. *Int J Hum Comput Stud* 159:102755. <https://doi.org/10.1016/j.ijhcs.2021.102755>
- Voigt-Antons J N, Kojic T, Ali D, Möller S (2020) Influence of hand tracking as a way of interaction in virtual reality on user experience. In: 2020 Twelfth international conference on quality of multimedia experience (QoMEX), Athlone, Ireland, pp 1–4. <https://doi.org/10.1109/QoMEX48832.2020.9123085>
- Wei SE, Ramakrishna V, Kanade T, Sheikh Y (2016) Convolutional pose machines. In: Proceedings of the IEEE conference on computer vision and pattern recognition, 4724–4732. <https://doi.org/10.48550/arXiv.1602.00134>
- Xiong F, Zhang B, Xiao Y, Cao Z, Yu T, Zhou JT, Yuan J (2019) A2J: anchor-to-joint regression network for 3D articulated pose estimation from a single depth image. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 793–802. <https://doi.org/10.48550/arXiv.1908.09999>
- Yang L, Li S, Lee D, Yao A (2019) Aligning latent spaces for 3D hand pose estimation. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 2335–2343. <https://doi.org/10.1109/ICCV.2019.00242>
- Yang L, Li K, Zhan X, Lv J, Xu W, Li J, Lu C (2022) ArtiBoost: boosting articulated 3D hand-object pose estimation via online exploration and synthesis. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 2750–2760. <https://doi.org/10.48550/arXiv.2109.05488>
- Zhao Z, Zhao X, Wang Y (2021) TravelNet: self-supervised physically plausible hand motion learning from monocular color images. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 11666–11676. <https://doi.org/10.1109/ICCV48922.2021.01146>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.