A Thumb Tip Wearable Device Consisting of Multiple Cameras to Measure Thumb Posture

Naoto Ienaga[1(✉)], Wataru Kawai[2], Koji Fujita[3], Natsuki Miyata[4], Yuta Sugiura[1], and Hideo Saito[1]

[1] Keio University, Yokohama, Japan
ienaga@hvrl.ics.keio.ac.jp
[2] The University of Tokyo, Tokyo, Japan
[3] Tokyo Medical and Dental University, Tokyo, Japan
[4] National Institute of Advanced Industrial Science and Technology, Tokyo, Japan

Abstract. Today, cameras have become smaller and cheaper and can be utilized in various scenes. We took advantage of that to develop a thumb tip wearable device to estimate joint angles of a thumb as measuring human finger postures is important in terms of human-computer interface and to analyze human behavior. The device we developed consists of three small cameras attached at different angles so the cameras can capture the four fingers. We assumed that the appearance of the four fingers would change depending on the joint angles of the thumb. We made a convolutional neural network learn a regression relationship between the joint angles of the thumb and the images taken by the cameras. In this paper, we captured the keypoint positions of the thumb with a USB sensor device and calculated the joint angles to construct a dataset. The root mean squared error of the test data was 6.23 and 4.75 degrees.

Keywords: Wearable device · Human computer interaction · Pose estimation.

1 Introduction

Human sensing has become an increasingly indispensable technology in recent years. Contributions to promoting human health by measuring and analyzing human behavior is needed in the world where the number of aging individuals is increasing. In addition, human-computer interfaces are becoming essential as many people use multiple computers. In particular, finger posture measurement is important in designs for people to make devices that are easy to use and virtual reality which has become popular in recent years.

Methods for measuring finger posture can be roughly divided into three categories depending on where a sensor is attached: the environment, specific object, or the hand itself.

Environment A convolutional neural network (CNN) achieves amazing results in various areas of computer vision, including hand keypoint detection.

Methods were proposed to improve the detector by reconstructing three-dimensional (3D) hand keypoints and reprojecting them when generating training data [8], and to estimate the 3D hand keypoints from RGB images [11] and from RGB-D images [6]. There is also a hand pose estimation method using RGB-D images and a machine learning method other than CNN [9]. These methods do not require attaching the sensor to the hand, but the measurement could fail if the hand hides from the sensor.

Specific object Studies have measured hand motion with a sensor attached to a specific object. For instance, a fisheye camera was fixed to the top of a bottle to estimate how the hand grips the bottle [3], and a band sensor was developed to estimate the grasping hand posture [5]. Although the sensor is robust against occlusion when the sensor is attached to a specific object, the sensor can measure only when a human is interacting with the object.

Hand itself Many wearable devices were proposed since they are robust to the occlusion. A typical wearable sensor is a data glove. It is possible to calculate the joint angles of the hand with a sensor embedded in the glove. Data gloves need to cover the whole hand, wrist-worn sensors [4, 2, 7] and a camera [10] were developed to classify the hand pose. A fisheye camera ring was also developed to estimated hand gestures and palm writing by acquiring an image of a hand inside [1].

Our device is also attached to the hand itself. However, unlike previous studies, we utilize multiple cameras in this research. Today, cameras have become smaller and cheaper and can be expected to be utilized in various scenes. We take advantage of that to develop a thumb tip wearable device to measure joint angles of a thumb. The device consists of three small cameras attached at different angles so that the cameras can capture the four fingers. We assume that the appearance of the four fingers will change depending on the joint angles of the thumb.

With the advent of CNNs, it is possible to link input images and their labels automatically as long as there is a large amount of training data. The regression relationship between the joint angles and the images taken by the cameras is learned by the CNN. In this paper, we conduct an initial experiment to assess the effectiveness of the developed device. We captured the keypoint positions of the thumb with the Leap Motion (Leap Motion, `https://www.leapmotion.com`, last accessed: May 5, 2018) and calculated the joint angles of the thumb.

Our contributions are summarized as follows:

- We develop a thumb tip wearable device that use multiple small and inexpensive cameras. It can be easily attached to the thumb and used to estimate the thumb posture.
- We suggest that the finger joint angles can be estimated from the appearance of the fingers with the CNN.

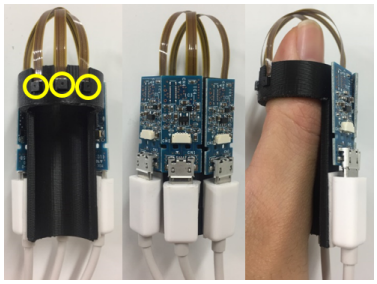

Fig. 1. The thumb tip wearable device we developed. Three cameras (left, circled) and their driving circuits are fixed to the device (center). The device can be attached to the thumb as shown on the right.

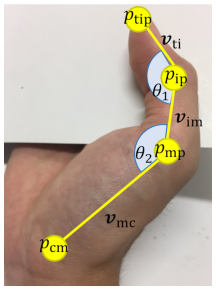

Fig. 2. θ_1 and θ_2 are calculated by finding the inner product of three vectors $\boldsymbol{v}_{\mathrm{ti}}$, $\boldsymbol{v}_{\mathrm{im}}$ and $\boldsymbol{v}_{\mathrm{mc}}$ from the 3D positions of the four thumb keypoints; p_{tip}, p_{ip}, p_{mp} and p_{cm}.

2 Methodology

First, we describe the device we developed. We then explain the constructed dataset. Afterward, we describe the CNN architecture.

2.1 Device

We developed the thumb tip wearable device shown in Fig. 1. We use three small cameras which are arranged at 30° intervals to capture the four fingers. The cameras are IU233N2-Z manufactured by Sony Semiconductor Solutions Corporation (IU233N2-Z/IU233N5-Z, `https://www.sony-semicon.co.jp/products_en/new_pro/december_2016/iu233_e.html`, last accessed: May 5, 2018). The driving circuits are also fixed to the device.

2.2 Dataset

In this research, a dataset is constructed by using the images captured by the cameras as input data and the joint angles of the thumb acquired by the Leap Motion as the outputs. Fig. 3 shows the scenes building the dataset. The acquired image and the 3D positions of the hand keypoints change according to the hand shape. Three captured images are converted to grayscale, resized to 1/8 in height and width, and then connected in the vertical direction to make a single grayscale image of 80 × 180 pixels (examples are in Fig. 3 (b) (b')). The Leap Motion can easily acquire the 3D positions of the hand keypoints, but the accuracy is inferior to motion capture. Therefore, data is acquired only at the frame when the two joint angles θ_1 and θ_2 of the thumb are larger than 0 degrees and smaller than 120 degrees. θ_1 and θ_2 are calculated by finding the inner product of three vectors $\boldsymbol{v}_{\mathrm{ti}}$, $\boldsymbol{v}_{\mathrm{im}}$ and $\boldsymbol{v}_{\mathrm{mc}}$ from the 3D positions of the four thumb keypoints; tip p_{tip}, interphalangeal joint p_{ip}, metacarpophalangeal joint p_{mp} and carpometacarpal joint p_{cm}. These relationships are shown in Fig. 2.

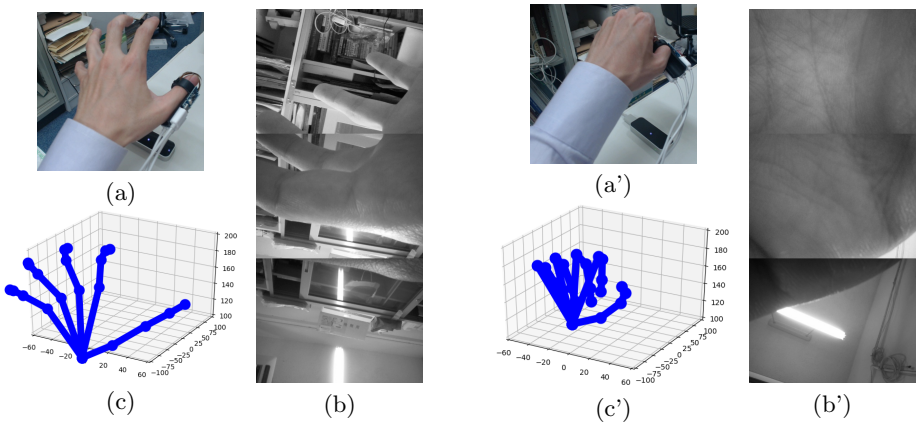

Fig. 3. Examples of the data collection. We used the device consisting of three cameras and the Leap Motion (a) and (a'). Three captured images are converted to grayscale, and then connected in the vertical direction to make a single image (b) and (b'). The Leap Motion provides the 3D positions of the hand keypoints (c) and (c').

Table 1. The CNN architecture. We repeat the first part three times and double the number of filters of the convolutional layer at each time. The activation function, ReLU, is used after all convolutional and fully connected layers. The batch normalization is used before all max pooling layers. After the first two fully connected layers, the dropout is 50%.

Layer	Filter size, strides or number of units
Input	$80 \times 180 \times 1$
Convolutional	$3 \times 3 \times$ (1st:32, 2nd:64, 3rd:128), 1
Convolutional	$3 \times 3 \times$ (1st:32, 2nd:64, 3rd:128), 1
Max pooling	2×2, 2
Fully connected	1024
Fully connected	1024
Fully connected	2

2.3 Network

The regression relationship for predicting the output value θ_1 and θ_2 estimated from the input image is learned by the CNN. The architecture of the CNN is shown in Table 1.

3 Experiment

The input image was scaled so that each pixel value falls within the range of [0, 1]. Then the average of all input data was subtracted from the input data and divided by the standard deviation of all input data. The training data was shuffled randomly. The input data was augmented by random flipping (left to

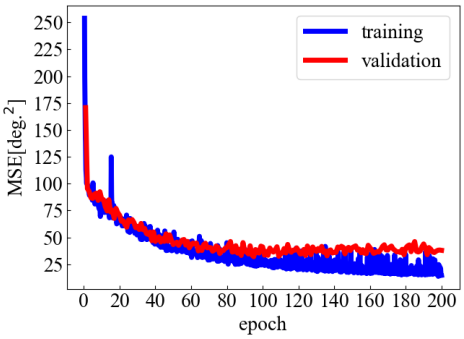

Fig. 4. Training and validation loss.

Table 2. MSE and RMSE of the testing.

Thumb angle	θ_1	θ_2
MSE[deg.^{2}]	38.83	22.52
RMSE[deg.]	6.23	4.75

Table 3. Correct and predicted joint angles of the thumb, and the MSE and RMSE of Fig. 5's *scene 1* and *scene 2*.

	Correct θ_1, θ_2		Predicted θ_1, θ_2		MSE[deg.^{2}]		RMSE[deg.]	
scene 1	2.58	4.27	15.69	8.68	171.70	19.42	13.10	4.41
scene 2	30.12	21.14	20.19	16.00	98.57	26.47	9.93	5.14

right) and random rotation by a random angle (max angle is 25). We used the Adam optimizer ($\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 1.0$). The learning rate was 0.03, and the batch size was 256. Loss function was the mean squared error (MSE). One male subject gathered a dataset while moving his hands randomly and keeping the palm level with the Leap Motion; 17931 frames were collected at around 30 fps (8997 frames with the right hand and 8934 frames with the left hand). Of those, 12553 frames were used for training, and 2689 frames were used for validation and testing. The training curve is illustrated in Fig. 4.

As the result of the validation because the loss of 95 epochs was the smallest (MSE: 31.93), we tested using the CNN learned with 95 epochs. The results of the MSE and the root mean squared error (RMSE) of the testing are shown in Table 2. Two testing examples and the results are shown in Table 3 and Fig. 5.

4 Discussion

Fig. 6 shows a breakdown of the angle of the dataset. It is understood that most data is less than 40 degrees. There are some possible reasons for this.

The first is that it is difficult to move one's hand completely randomly. If we moved our hands randomly, we would not cover the inner range of the motion. To solve this problem, it is necessary to first generate movements that uniformly cover the inner motion range and to have the subjects imitate the movements as much as possible to create a dataset.

The next reason is the accuracy limit of the Leap Motion. This time, we used the Leap Motion because it is easy to use. However, especially when the thumb was bent deeply (θ_1 and θ_2 are large), we could not measure it properly.

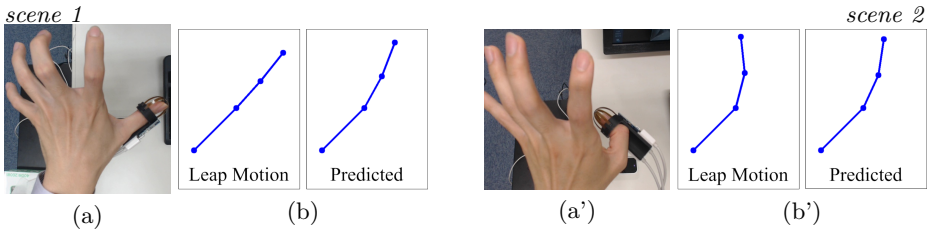

Fig. 5. Two specific examples of testing, *scene 1* and *scene 2*. the hand states (a) and (a'), and the states of the thumb are reconstructed by θ_1 and θ_2 (b) and (b'). The left of (b) and (b') was created based on θ_1 and θ_2 acquired by the Leap Motion, and the right was created based on the predicted θ_1 and θ_2. Note that other parameters (e.g., finger length) are appropriate values because only angles were estimated.

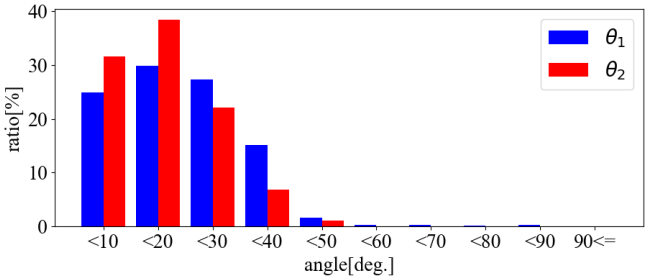

Fig. 6. A breakdown of the angle of the dataset. For example, about 25% of the dataset is data with θ_1 less than 10 degrees.

In *scene 1* of Fig. 5, the thumb is straight (a), and the illustration of the thumb created based on the thumb angles acquired by the Leap Motion (the left of (b)) is almost straight. However, in *scene 2*, the thumb actually (a') bends much more than the illustration (the left of (b')). To build a more accurate dataset, we have to use motion capture.

The limited accuracy may be caused by the device. The part that integrates the driving circuits with the device was large and hard, and cables between the cameras and the driving circuits were at the tip of the thumb. This design hindered the thumb movement. Again, cameras have become smaller and cheaper recently. We believe that this trend will continue. Camera images could be available via Wi-Fi or Bluetooth. Thus, these camera problems will be solved someday. Although we used three cameras this time, we are considering using more cameras to acquire more information in the future.

5 Conclusions

Taking advantage of the fact that cameras have become smaller and cheaper recently, we developed a thumb tip wearable device that can be easily attached to a thumb tip to measure the joint angles of the thumb as measuring finger

movements has become an important technique especially in recent years. We proposed a method using a CNN to estimate the joint angles. We experimentally suggested that the finger joint angles wearing the device could be estimated by the state of the fingers.

We ignored the background of the input image; its influence was unknown (e.g., when the background changes). It may be necessary to incorporate some foreground (e.g., the hand) segmentation process into our deep learning framework.

In the future, we would like to propose a device and a method that can estimate joint angles other than the finger to which the device is attached by increasing the number of cameras and devising their arrangement. Furthermore, the device and the construction method of the dataset will be also improved.

Acknowledgements. This work was supported by JST AIP-PRISM Grant Number JPMJCR18Y2, Grant-in-Aid for JSPS Research Fellow Grant Number JP17J05489.

References

1. Chan, L., Chen, Y.L., Hsieh, C.H., Liang, R.H., Chen, B.Y.: Cyclopsring: Enabling whole-hand and context-aware interactions through a fisheye ring. In: Proc. UIST. pp. 549–556 (2015)
2. Fukui, R., Watanabe, M., Shimosaka, M., Sato, T.: Hand shape classification with a wrist contour sensor. In: Proc. Experimental Robotics. pp. 939–949 (2013)
3. Kashiwagi, N., Sugiura, Y., Miyata, N., Tada, M., Sugimoto, M., Saito, H.: Measuring grasp posture using an embedded camera. In: Proc. WACVW. pp. 42–47 (2017)
4. Kim, D., Hilliges, O., Izadi, S., Butler, A.D., Chen, J., Oikonomidis, I., Olivier, P.: Digits: Freehand 3d interactions anywhere using a wrist-worn gloveless sensor. In: Proc. UIST. pp. 167–176 (2012)
5. Miyata, N., Honoki, T., Maeda, Y., Endo, Y., Tada, M., Sugiura, Y.: Wrap & sense: Grasp capture by a band sensor. In: Proc. UIST. pp. 87–89 (2016)
6. Mueller, F., Mehta, D., Sotnychenko, O., Sridhar, S., Casas, D., Theobalt, C.: Real-time hand tracking under occlusion from an egocentric rgb-d sensor. In: Proc. ICCVW. pp. 1284–1293 (2017)
7. Rekimoto, J.: Gesturewrist and gesturepad: Unobtrusive wearable interaction devices. In: Proc. ISWC. pp. 21–27 (2001)
8. Simon, T., Joo, H., Matthews, I., Sheikh, Y.: Hand keypoint detection in single images using multiview bootstrapping. In: Proc. CVPR. pp. 1145–1153 (2017)
9. Sridhar, S., Mueller, F., Oulasvirta, A., Theobalt, C.: Fast and robust hand tracking using detection-guided optimization. In: Proc. CVPR. pp. 3213–3221 (2015)
10. Vardy, A., Robinson, J., Cheng, L.T.: The wristcam as input device. In: Proc. ISWC (1999)
11. Zimmermann, C., Brox, T.: Learning to estimate 3d hand pose from single rgb images. In: Proc. ICCV. pp. 4903–4911 (2017)