Development of a Practical Tool in Pick-and-Place Tasks for Human Workers

Yunan He[1], Osamu Fukuda[2], Daisuke Sakaguchi[3]
Nobuhiko Yamaguchi[4], Hiroshi Okumura[5], Kohei Arai[6]
Computing Division, Graduate School of Science and Engineering, Saga University
Saga 840-8502, Japan

***Abstract*—We introduce smart hand, a practical tool for human workers in pick-and-place tasks. It is developed to avoid picking up the wrong thing from one location or place the things in an unexpected location. Smart hand features sensors (e.g., imaging sensors, motion sensors) to sense the world and offers suggestions or aid based on the sensed results when a human worker is performing a pick-and-place task. A smart hand prototype is made in the study. In our design, the smart hand has an RGB-D sensor and an inertial measurement unit (IMU). RGB-D sensor is used to do object detection and distance/position estimation while IMU is used to track the motion of the smart hand. An experiment is conducted to compare the two working conditions that a subject performs the pick-and-place tasks with or without the smart hand. The experiment results proved that the smart hand can avoid human errors in pick-and-place tasks.**

***Keywords*—Pick-and-place task; human-robot collaboration; cognitive system; hand tools**

I. Introduction

Picking up parts in a production line and placing them with some rules is a very common task in a manufacturing plant. For example, the workers in an assembly line need to select the specified bolt to attach two pieces of a design with an appropriate torque. In a factory that makes box lunch, the workers need to pick up a certain amount of food into the lunch box [1]. In a manual sorting line, the workers tried to sort the objects into different categories and place them into corresponding regions. The rules involved in these tasks include selection (which object to pick up), positioning (where to place) and some other task-specified rules like controlling torque or weight. Sometimes due to carelessness or exhaustion after long time working, the workers may make some mistakes in these operations and the rules involved in these tasks cannot be well followed. The improper torque in a machine may lead to an accident or even worse. The less amount of food or wrong kind of food in the box lunch can lead to customer dissatisfaction.

To avoid these circumstances and follow the rules involved in a pick-and-place task, the managers of manufacturing plants take steps to strengthen the production management by introducing the record management system, adding inspection procedures or increasing the break time to avoid exhaustion. Besides the increasing cost, these methods couldn't improve the condition of making errors during the operations fundamentally. The researchers from different fields also come up with many solutions. They try to increase the degree of automation so that the human errors can be avoided. Many automatic screw tightening systems that have been developed since the 90s [2], [3]. But it has a high demand in precise relative position between the parts and the screwdrivers. The parts are usually fixed in a specified pose so that the screwdriver can be programmed to find a feed direction. In this way, the selection and positioning problems won't exist any more and the torque can also be recorded and managed. However, for small parts, it is fine to do so. But for large machines like cars or planes, there are so many spaces that the automatic screw tightening system cannot reach. Similarly, in a robotic grasping and sorting system, the position and orientation of the objects are usually hard coded so that the robot can successfully pick up the objects. Nowadays, computer vision has been introduced to these systems. Selection tasks can be finished using object recognition algorithms while positioning problems are expected to be solved using camera coordination. For example. Chen et al. built a vision-based robotic grasping system using deep learning for garbage sorting [4]. It uses convolutional neural networks (CNN) to identify objects and their locations to grasp objects. A research group from Google took fourteen robotic arms, networked them together to make these robots learn on their own how to pick up small objects [5]. It also uses CNN to learn the pose of the objects. Stage of the art technologies can recognize objects in a high accuracy using CNN, which solves the selection problem, but object pose estimation for grasping using imaging sensor alone is not accurate enough for industries.

As mentioned above, the rules involved in pick-and-place tasks basically include selection and positioning. Human workers may make mistakes during the operations but for robot workers, either vision-based or programed-based coordination in reach-to-grasp movement has limitations. We are considering building a practical tool for human workers that can help them avoid the mistakes during operations. In the case of a human worker, the supervision of following rules are controlled by the brains but the execution of these rules are performed by the hand, which inspires us to develop a hand-like tool where we can embed some intelligence to help workers supervise the rules and reduce the brain burden. In other words, the hand-like tool can assist people in

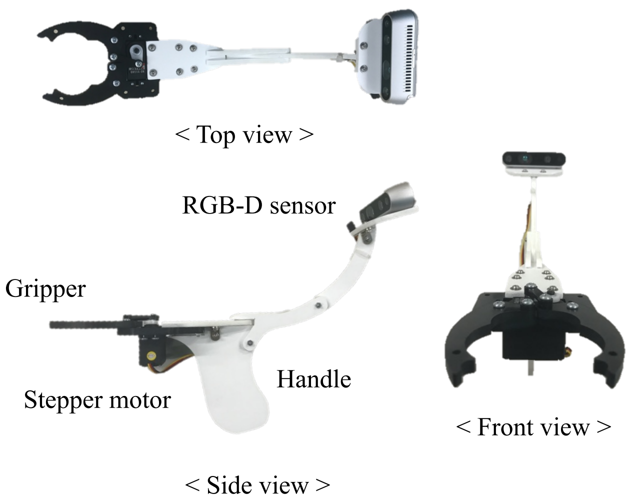


Fig. 1. Smart hand prototype

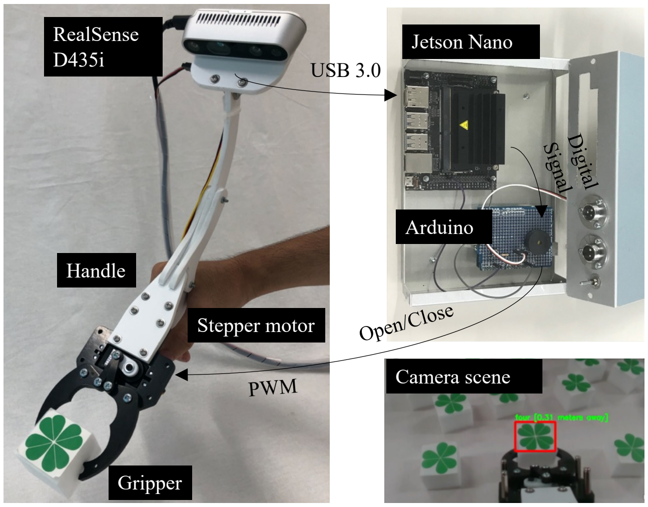


Fig. 2. Smart hand control pipeline

performing the pick-and-place tasks by supervising the rules predefined.

In this paper, we introduce smart hands to assist human workers in performing pick-and-place tasks. We integrate an imaging sensor, a depth sensor as well as an IMU sensor in the control system of the smart hand. Human workers can hold the smart hand to do the pick-and-place movement. The smart hand has imaging sensors to recognize objects and make selections. It also has an IMU sensor to track motions so that the location that an object is placed can be tracked. The smart hand is expected to reduce the workload burden and improve the product quality in a production line.

The remainder of this paper is organized as follows: Section II introduces the prototype of the smart hand. Section III explains three main modules used in the control system of the smart hand. Experiment conducted with the smart hand prototype and its results are presented in Section IV. We then draw some conclusions and outline the future work in Section V.

II. Smart hand

Fig. 1 shows a smart hand prototype which is designed to assist users in performing pick-and-place task. The mechanical structure of the smart hand is designed to make it simple and light. From the side view, we can see that the smart hand prototype consists of a gripper, a stepper motor, a handle and some sensors. Its end effector is a gripper with one degree of freedom, which is used to pick and hold objects. A stepper motor is used to control the gripper to perform close and open movement. A handle is designed to be able to hold and orientate the smart hand easily for users. The sensors are the core parts of the smart hand, which is the reason for calling it “smart”. They sense the surrounding environment, assisting users in picking and placing objects. The sensors are mounted on a quick release plate. The orientation of the sensors can be adjusted by setting angular position of the quick release plate.

In the prototype design, the sensors include an RGB-D camera and an IMU. RGB-D sensor streams RGB image together with the corresponding depth. It can be used for recognizing and localizing the target objects. Since RGB-D sensor is limited by a minimum detectable distance to function properly, it is connected to the gripper with a long arm to make sure that the target object is always inside the detectable distance. IMU is used for the detection of movements and rotations in 6 degrees of freedom. It is used to supervise the placing position in a pick-and-place movement. Intel RealSense depth camera D435i is selected for the designed prototype. It combines the depth sensing capabilities with the addition of an IMU [6].

The sensor data from the depth camera are sent to Jetson Nano through USB 3.0 and processed there. See Fig. 2. Jetson Nano is a small-size computer integrating a 128-core NVIDIA GPU where you can run multiple neural networks in parallel for applications like image classification, object detection, segmentation and speech processing. It is selected because of its compact size and high computation performance. Remote server with state-of-the-art GPU has also been developed to process the sensor data only if the computational power of the Jetson Nano is not enough for practical applications. The commands generated based on the processing results are sent to the Arduino through digital GPIO pins. Arduino then triggers the corresponding events based on the received commands. It controls the stepper motor to open and close using pulse width modulation (PWM) signals. It also controls a buzzer and LEDs to warn the system state for simple interface with users. Jetson Nano and Arduino are powered with a mobile battery. They are all installed in a compact control box.

III. Control System

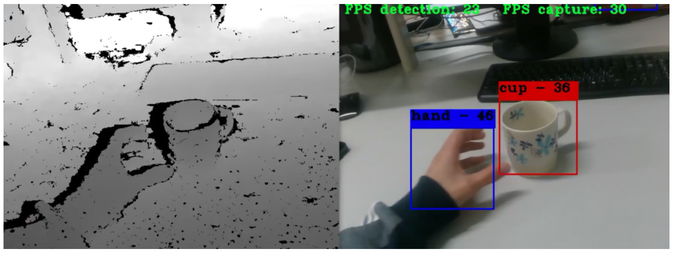
The smart hand features three function modules including object recognition, distance estimation and motion tracking to assist users in pick-and-place tasks. These three functions are realized using the hardware introduced above. The three function modules are used to supervise the whole movement that if it follows the predefined rules. For example, the object detection model is used to detect the objects and help the user analyze whether the object is the expected target or not. Distance estimation can be used to measure the distance between the target object and the hand. The distance then controls the timing to trigger the open/close movement. Motion tracking is to track the motion of the smart hand. It can help the user to determine if the object is properly placed on the assigned place. These three function modules are discussed in the followings.

A. Object Detection

RealSense depth camera D435i streams RGB images on which the control system detects the target potential objects. The detection results are either given by visual clues that are shown with a monitor or prompted by buzzing sounds. The object detection module solves the problem of selection in pick-and-place movement. If users use the smart hand to pick up some object that is not expected, the control system will show the warning information using visual cues or specific buzzing sound.

Object detection algorithm improves a lot these years due to the deep learning evolution in computer vision. The novel object detection method can even comparable with cognition capability of human. YOLO is one of the most popular object detection algorithms due to its high processing speed and reliable detection result. It is a convolutional neural network that accepts images as input and outputs the object class together with object location in an image. YOLO processes images at 30 FPS on a Pascal Titan X and has a mAP of 57.9% on COCO dataset [7]. YOLO has several variants. The main differences of these variants are the number of convolutional layers. More convolutional layers means higher accuracy and lower processing speed. For example, YOLOv3-tiny is the compact version of YOLOv3 network. YOLOv3 has 53 convolutional layers for feature extraction while YOLOv3-tiny has only 10 layers.

Among the three function modules of the smart hand, object detection takes up the most computing resources. To find the best platform for running object detection module, we tested the processing speed of YOLOv3 and YOLOv3-tiny on Jetson Nano, Jetson TX2 and remote server, respectively. The remote server owns a Nvidia GeForce GTX Titan X GPU. These platforms are selected because we want the smart hand to be portable. The results are shown in Table I. It can be seen from the table that running YOLOv3 on Jetson Nano and Jetson TX2 is not suitable since the processing speed (FPS) is too slow for practical applications. YOLOv3-tiny works fine on both Jetson Nano and Jetson TX2. As for the remote server, besides the processing speed of GPU, the FPS also relies on the transmission speed of the network. Since the size of the remote server is not limited, modern desktop GPUs can be used so that the computational power is not a problem.



(a) Depth map (b) RGB image

Fig. 3. Object detection

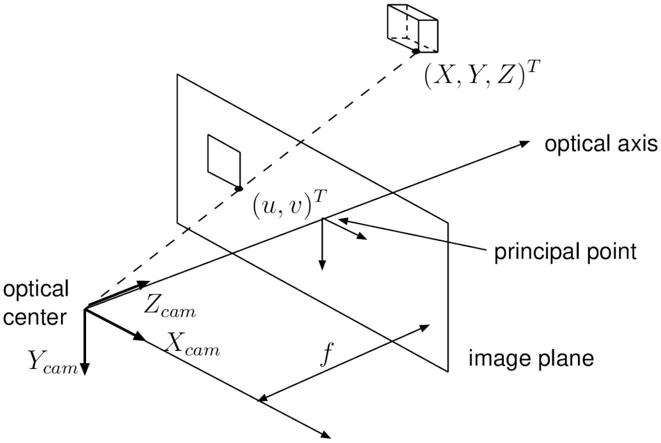


Fig. 4. Converting a 2D image point to a 3D world point

TABLE I. FPS Comparison of YOLO Model

	Jetson TX2	Jetson Nano	Remote Server
YOLOv3	5 FPS	3 FPS	16 FPS
YOLOv3-tiny	16 FPS	14 FPS	Not tested

Considering the processing speed in different platforms shown in Table I, we use either Jetson Nano to run YOLOv3-tiny or the remote server to run YOLOv3. An object detection example of running YOLOv3 with remote server is shown in Fig. 3(b). It detects the cup and the hands in a reach-to-grasp movement. Fig. 3(a) shows the corresponding depth map of the same camera scene.

B. Position Estimation

RGB-D sensor can be used to estimate the position of the target object in real world. By estimating the position, the smart hand can be aware of its spatial relationship with the target object. When the smart hand is within the distance range that can successfully grasp an object, the control system

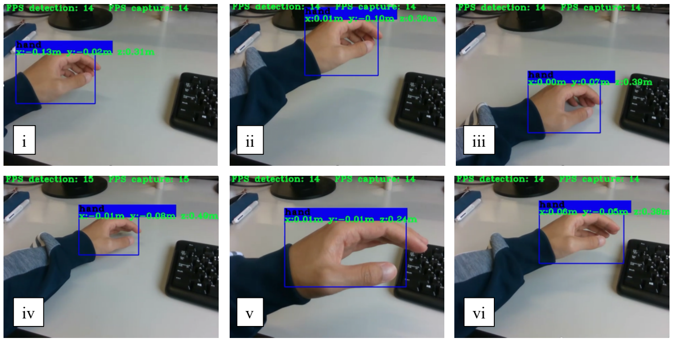

FPS detection: 14 FPS capture: 14
hand
x:-0.13m y:-0.02m z:0.31m
i
FPS detection: 14 FPS capture: 14
hand
x:0.01m y:-0.10m z:0.36m
ii
FPS detection: 14 FPS capture: 14
hand
x:0.00m y:0.07m z:0.39m
iii
FPS detection: 15 FPS capture: 15
hand
x:-0.01m y:-0.08m z:0.49m
iv
FPS detection: 14 FPS capture: 14
hand
x:0.01m y:-0.01m z:0.24m
v
FPS detection: 14 FPS capture: 14
hand
x:0.06m y:-0.05m z:0.38m
vi

Fig. 5. Hand position estimation

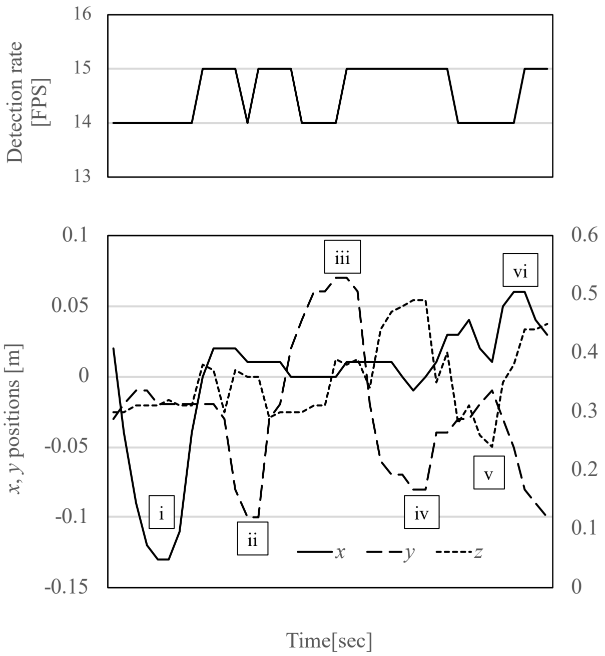


Fig. 6. Tracking the position of the hand

can automatically trigger the command to close the hand and grasp the object. With the object detection module, the control system can detect objects from RGB images streamed from the RGB-D sensor in real-time (Fig. 3(b)). The RGB-D sensor also captures the corresponding depth map of the scene right after an RGB image is captured (Fig. 3(a)).

From the depth map we can estimate the object distance using the detection results from the object detection module. The object detection module offers us the center point of an object , which indicates the location of the object in the image. Using the same center point, the object can be located in the corresponding depth map. If we crop an image patch with size of 20×20 from the center point of an object in the depth map, the object distance can be defined as the average pixel value of this image patch.

After we get the distance of the object, the problem of estimating the object position becomes converting a 2D image point to a 3D world point. With the principles from the perspective projection [8] as shown in Fig. 4, we can easily estimate the object position using Eq. 1,

$$\begin{bmatrix} \mu \\ \nu \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{1}$$

where (X, Y, Z) is the 3D world point and (μ, ν) is the corresponding 2D image pixel point. f is the focal length of the camera. Since RGB sensor and depth sensor have different viewpoints, the depth map needs to be aligned to RGB sensor to have the same viewpoints before estimating the object position. If there are multiple objects in the image scene, the control system needs to identify the target object. We define the object that is nearest to the center of the image is the target object. The object position is only calculated and tracked on target object. An example is shown in Fig. 5. The position of the hand is estimated in every frame and the result is shown with a label near the bounding box. Fig. 6 shows the hand tracking path of the example in Fig. 5.

C. Motion Tracking

An IMU is used track the motion of the hand so as to track the position and orientation of the smart hand at any time in a pick-and-place movement. Tracking the position and orientation of the smart hand can help us to determine if an object is placed in the expected position. Inside an IMU unit there are 3 accelerometers and 3 gyroscopes. Accelerometers measure all forces working on an object while gyroscope

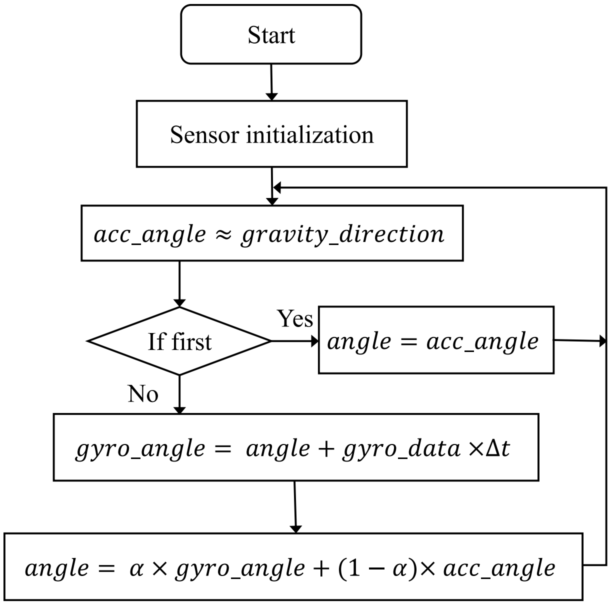

Start
Sensor initialization
acc_angle ≈ gravity_direction
If first
Yes
angle = acc_angle
No
gyro_angle = angle + gyro_data ×Δt
angle = α × gyro_angle + (1 − α)× acc_angle

Fig. 7. Complementary filter flow chart

measures the angular velocity.

For orientation, both the accelerometers and gyroscopes are able to estimate the angular position of the smart hand. Gyroscope can achieve this by integrating the angular velocity over time. Accelerometer can do this by determining the direction of the gravity vector. However, both of the methods have problems. It is not able to measure the precise angular position of the smart hand by using only gyroscope or accelerometer. In the case of accelerometer, every small forces working on an object create disturbance in measurement, long term measurement is reliable. So low pass filter is needed for correction. In the case of gyroscopic sensor, the integration is done over period of time the value starts to drift in the long term, so high pass filter is needed for gyroscopic data correction [9]. Therefore, the control system selects a complementary filter that consists of both low and high pass.

By using the complementary filter, both the accelerometer and gyroscope make contributions in estimation the angular position of the smart hand. Gyroscope data is used on the short term since it is very precise and not susceptible to external forces. Accelerometer data is used on the long term as it does not drift. The complementary filter flow chart is shown in Fig. 7. The initial state of angular position is determined using the accelerometer only. The gyroscope data is integrated every timestep with the current angle value. Then it is combined with the low-pass data from the accelerometer. The constant α is the coefficient value of the complementary filter. In our design, $\alpha = 0.98$.

IV. Experiments

We designed an experiment to validate the functions of the smart hand and prove that smart hands actually do the help when the user try to perform a pick-and-place task. In the experiment, we prepared 100 sponge cubes and of which, 70 sponge cubes were attached with stickers of three-leaf clovers and the rest 30 sponge cubes were attached with four-leaf stickers. The clover cubes are scattered on the table. As it can be seen from Fig. 8. We trained an object detection neural network described in Section 3.3 to distinguish the two kinds of clover cubes. The network runs at 12.6 FPS averaged on Jetson Nano. Three subjects aged from 23 to 27 are asked to pick out all the four-leaf clover and put them in a box one by one. If a subject cannot find a four-leaf clover within 5 seconds, the experiment is halted. The number of clovers that have not been found and the number of three-leaf clover that has not been picked out are reported.

The experiment results are shown in Table II and Table III. It is relatively easy for human to pick out the four-leaf clover if they have enough time. But under the time pressure, they may make some mistakes. Three cases of failure in finding the four-leaf clovers are identified when the subjects used their own hands. Compared to the human hands, experiment with smart hands has only one case failure in finding the four-leaf clover. No mistaken reports in both cases.

TABLE II. Pick-and-place with human hands

	Subject A	Subject B	Subject C
Failed in finding four-leaf clovers	0	2	1
Mistakenly pick the three-leaf clover	0	0	0

TABLE III. Pick-and-place with the smart hand

	Subject A	Subject B	Subject C
Failed in finding four-leaf clovers	0	1	0
Mistakenly pick the three-leaf clover	0	0	0

Generally, the capability of the smart hand to find a four-leaf clover highly relies on the performance of the neural network model. One failure case identified means that the object detection network may not work very well from some specific camera view angle. Increasing the size of the dataset may improve its performance. No matter the subjects used their own hands or the smart hand, they never picked the three-leaf clover mistakenly. It may because that the experiment is a short-term task. With time restrictions, the subjects are highly focused, and they hardly made a mistake by picking a three-leaf clover. If it is a long, boring task, they may have chances to put a three-leaf clover in the box mistakenly. But smart hands won't be tired, they can always remind the user whether a target is the expected object or not.

In addition, the subjects gave their feedback. They thought the way to interact with the smart hands is not convenient. The buzzing sound is easy to understand but the recognition results are given on a monitor. They need to see the monitor first to check the detection results. Since the camera view and the user's eye view are in different angles, it is quite difficult to quickly find the target object. Head-mounted display with AR technology may be a better interaction solution.

V. Conclusion

This study introduces smart hands to assist human workers in repeated, boring pick-and-place tasks. The smart hand

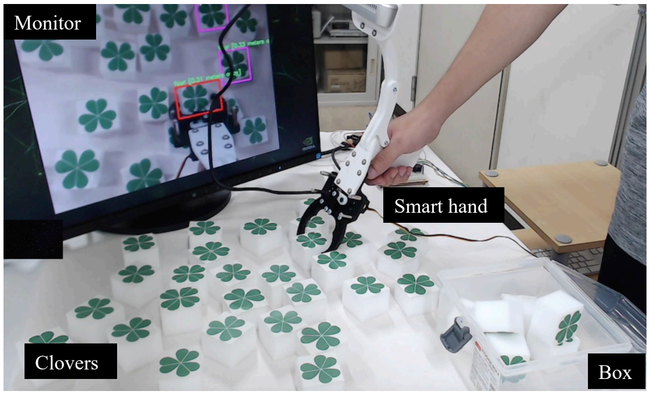


Fig. 8. Experiment setup

features functions of object detection, position estimation and motion tracking by combining vision and motion sensors. It can offer suggestions on selecting the target object, warn the users when picking up the wrong objects and placing in the wrong location. We made a prototype of smart hands. The experiment proves that the use of smart hands can avoid human errors in pick-and-place tasks. It is expected that the smart hand can reduce the workload burden and improve the product quality in a production line. In the future, we would like to improve the interaction method between the user and the smart hand. A head mounted display instead of a monitor may improve the work efficiency. In addition, the end effector of gripper can only do limited work in a production line. More end effectors (e.g. screw driver) should be developed to make the smart hand fit most usage scenarios in a production line.

ACKNOWLEDGMENT

This work was supported by JSPS KAKENHI Grant Number JP19K04296.

REFERENCES

[1] A. Pettersson, S. Davis, J. Gray, T. Dodd, and T. Ohlsson, “Design of a magnetorheological robot gripper for handling of delicate food products with varying shapes,” *Journal of Food Engineering*, vol. 98, no. 3, pp. 332–338, 2010.

[2] Y. Ota and H. Takahashi, “Automatic screw tightening apparatus,” Jul. 14 2015, uS Patent 9,079,275.

[3] H. Shibata, “Screw tightening apparatus,” Dec. 13 2005, uS Patent 6,973,856.

[4] C. Zhihong, Z. Hebin, W. Yanbo, L. Binyan, and L. Yu, “A vision-based robotic grasping system using deep learning for garbage sorting,” in *2017 36th Chinese Control Conference (CCC)*. IEEE, 2017, pp. 11 223–11 226.

[5] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, “Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection,” *The International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 421–436, 2018.

[6] A. Grunnet-Jepsen, J. N. Sweetser, and J. Woodfill, “Best-known-methods for tuning intel® realsense$^{\text{TM}}$ d400 depth cameras for best performance,” *New Technologies Group, Intel Corporation, Rev*, vol. 1, 2018.

[7] J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” *arXiv preprint arXiv:1804.02767*, 2018.

[8] Y.-Y. Chuang, “Camera calibration,” Tech. Rep., 2005.

[9] T. Islam, M. S. Islam, M. Shajid-Ul-Mahmud, and M. Hossam-E-Haider, “Comparison of complementary and kalman filter based data fusion for attitude heading reference system,” in *AIP Conference Proceedings*, vol. 1919, no. 1. AIP Publishing LLC, 2017, p. 020002.

Yunan He received the B.E. degree in mechanical engineering from Northeastern University, Shenyang, China, in 2013 and the M.E. degrees in mechanical engineering from Saga University, Saga, Japan, in 2017.

He is now a PhD student in Department of Information Science in Saga University, Saga, Japan. His main research interests are in human interface and intelligent robots.

Osamu Fukuda received his B.E. degree in mechanical engineering from Kyushu Institute of Technology, Iizuka, Japan, in 1993 and the M.E. and Ph.D. degrees in information engineering from Hiroshima University, Higashi-Hiroshima, Japan, in 1997 and 2000, respectively.

From 1997 to 1999, he was a Research Fellow of the Japan Society for the Promotion of Science. He joined Mechanical Engineering Laboratory, Agency of Industrial Science and Technology, Ministry of International Trade and Industry, Japan, in 2000. Then, he was a member of National Institute of Advanced Industrial Science and Technology, Japan from 2001 to 2013. Since 2014, he has been a Professor of Graduate School of Science and Engineering at Saga University, Japan. Prof. Fukuda won the K. S. Fu Memorial Best Transactions Paper Award of the IEEE Robotics and Automation Society in 2003. His main research interests are in human interface and neural networks. Also, he is currently a guest researcher of National Institute of Advanced Industrial Science and Technology, Japan. Prof. Fukuda is a member of IEEE and the Society of Instrument and Control Engineers in Japan.

Daisuke Sakaguchi received the B.E. degree in information engineering from Saga University, Saga, Japan, in 2019.

He is currently a graduate student in the Department of Information Science in Saga University, Saga, Japan. His research interest is human-machine interface.

Nobuhiko Yamaguchi received the Ph.D. degree in intelligence and computer science from Nagoya Institute of Technology, Japan, in 2003.

He is currently an Associate Professor of Faculty of Science and Engineering at Saga University. His research interests include neural networks. He is a member of Japan Society for Fuzzy Theory and Intelligent Informatics.

Hiroshi Okumura received the B.E. and M.E. degrees from Hosei University, Tokyo, Japan, in 1988 and 1990, respectively, and the Ph.D. degree from Chiba University, Chiba, Japan, in 1993.

He is currently a full Professor of Graduate School of Science and Engineering at Saga University, Japan. His main research interests are in remote sensing and image processing. He is a member of the International Society for Optics and Photonics (SPIE), the Institute of Electronics, Information and Communication Engineers (IEICE) and the Society of Instrument and Control Engineers (SICE).

Kohei Arai He received BS, MS and PhD degrees in 1972, 1974 and 1982, respectively. He was with The Institute for Industrial Science and Technology of the University of Tokyo from April 1974 to December 1978 also was with National Space Development Agency of Japan from January, 1979 to March, 1990. During from 1985 to 1987, he was with Canada Centre for Remote Sensing as a Post Doctoral Fellow of National Science and Engineering Research Council of Canada. He moved to Saga University as a Professor in Department of Information Science on April 1990. He was a councilor for the Aeronautics and Space related to the Technology Committee of the Ministry of Science and Technology during from 1998 to 2000. He was a councilor of Saga University for 2002 and 2003. He also was an executive councilor for the Remote Sensing Society of Japan for 2003 to 2005. He is an Adjunct Professor of University of Arizona, USA since 1998. He also is Vice Chairman of the Science Commission “A” of ICSU/COSPAR since 2008 then he is now award committee member of ICSU/COSPAR. He wrote 37 books and published 570 journal papers. He received 30 of awards including ICSU/COSPAR Vikram Sarabhai Medal in 2016, and Science award of Ministry of Mister of Education of Japan in 2015. He is now Editor-in-Chief of IJACSA and IJISA. http://teagis.ip.is.saga-u.ac.jp/index.ht