

Novel Control Scheme for Prosthetic Hands through Spatial Understanding

Yunan He¹, Osamu Fukuda², Nobuhiko Yamaguchi³, Hiroshi Okumura⁴, Kohei Arai⁵
Computing Division, Graduate School of Science and Engineering, Saga University
Saga 840-8502, Japan

Abstract—A novel control scheme for prosthetic hands through spatial understanding is proposed. The proposed control scheme features an imaging sensor and an inertial measurement unit (IMU) sensor, which makes prosthetic hands capable of visual and motion sensing. The imaging sensor captures the scene where the user is willing to grasp the object. The control system recognizes the target object, extracts its surface features and estimates its pose from the captured images. Then the spatial relationship is constructed between the hand and the target object. With the help of IMU sensors, the relationship can be tracked and kept wherever the prosthetic hand moves even the object is out of the view range of the camera. To interact with the user, this process is visualized using augmented reality (AR) technology. A test platform based on the proposed control scheme is developed and a case study is performed with the platform.

Keywords—Prosthetic hand; vision-inertial fusion; pose estimation; motion tracking; internal measurement unit; augmented reality; control scheme; spatial features

I. INTRODUCTION

Humans grasping objects have two stages: they first glance at an object and instantly know what orientation, position and shape it is. Eyes then coordinate the hand to properly grasp the object. Inspired by the way that human grasps objects, many studies integrate cameras in the prosthetic hand control system [1], [2], [3], [4], [5], [6], [7], [8]. Such systems accept images as input and extract necessary information (e.g. Shape, orientation, position) for controlling prosthetic hands to adjust grasp postures. Vision-based control has developed rapidly in recent years due to the deep learning revolution in the field of computer vision [9]. State of the art deep learning algorithms can accurately detect object class and recognize object pose from a single image [10]. The information can be further used in planning a grasping movement. For example, Došen et al. [1] built a control system that uses an ultrasound distance sensor and an imaging sensor to locate the target object and estimate object size, so as to determine the grasp type and open size. Bando et al. [11] used a convolutional neural network (CNN¹) to classify 20 classes of objects, the classification results help to select the grasp posture from a group of predefined postures. Shima et al. [12] takes advantage of object spatial information measured by depth sensor and classifies the objects in terms of their shapes. The shape of the object finally results in the grasp posture. In these studies, the grasp posture can be estimated but to successfully perform a grasp movement, the user needs to control the residual upper limb to orientate the prosthetic hand

in a proper position relative to the grasp target. Sometimes it has difficulties for users to do that.

If we look back to the way that humans grasp objects, it can be found that the vision-based control systems mentioned above only realize the first stage of grasping an object: recognizing object class, object position and object orientation. The orientation adjustment of the prosthetic hands still needs to be coordinated by human eyes. Since the control system has already integrated an imaging sensor, it is possible to use the imaging sensor instead of the human eyes to coordinate the prosthetic hand to grasp the objects. An intuitive method is to construct the spatial relationship between the prosthetic hand and the object using features from every image frame captured in a grasping session. The spatial relationship helps the prosthetic hand to adjust its orientation automatically. However, the algorithm complexity makes it hard to run in real time, especially when the grasp movement is relatively fast. In addition, the vision field of the imaging sensor will be very narrow at the end of the movement because the imaging sensor is usually fixed in the prosthetic hand, thus it will be difficult to extract the features.

Essentially, the camera-prosthesis coordination is to track the spatial relationship between the camera and the target object. Such a relationship guides the control system to adjust the orientation of the prosthetic hand. We introduce the combination of accelerometers and gyroscopes here to measure hand movement and further track the spatial relationship between the hand and target object. Compared to estimating the spatial relationship in every image frame, the spatial relationship tracked by accelerometers and gyroscopes costs extremely small computation resources and it is not restricted by the distance between the hand and the target object. The introduction of the accelerators and gyroscopes to the vision-based control system solves the camera-prosthesis coordination problem, which enables automatic orientation adjustment of prosthetic hand in reach-to-grasp movements.

The next section describes the proposed control system followed by some experiments. Then some discussions are described followed by conclusions with some additional discussions together with future research works.

II. CONTROL SYSTEM

The overview of the proposed control system is illustrated in Fig. 1. The control system integrates three sensors

¹https://en.wikipedia.org/wiki/Convolutional_neural_network

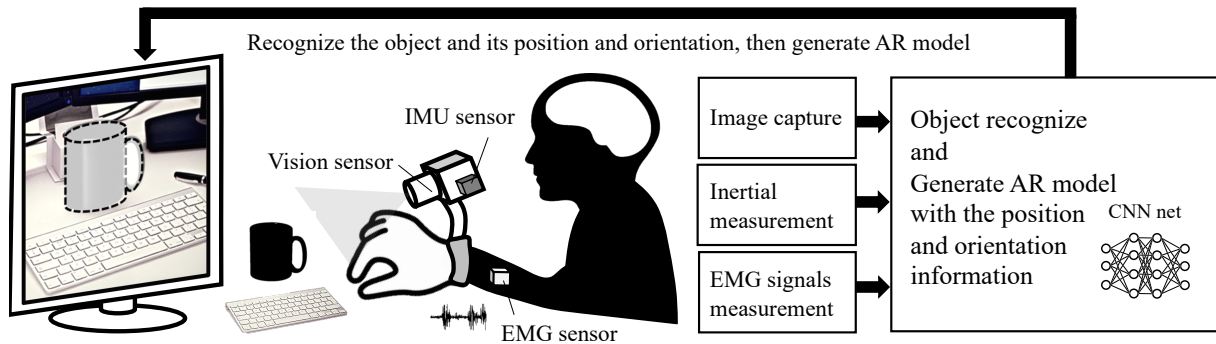


Fig. 1. System Overview.

(imaging sensor, IMU² sensor and EMG³ sensor) to help collect information from the surrounding environment and the user. Imaging sensors help to recognize the target object and extract its pose information, IMU sensors help to track the extracted pose information, and EMG sensors are used to measure the EMG signals. The signals are further used to estimate the user intention that if the user is willing to start/quit/finish a grasp session [13]. The sensors work together to let the prosthetic hand be aware of the object pose. To visualize this information and let the user know if the pose is accurately extracted, we use augmented reality (AR) technology to generate a 3D object model and match the model with the object that resides in the real world. The generated graphics can be viewed through a display. It is better to use a head-mounted display to merge the generated graphics with human vision [14].

When a grasp session is triggered by the EMG signal to start, the camera glances at the scene where the user tries to grasp an object, and the prosthetic hand instantly knows what class the object belongs to, where the object stands/lies on and which orientation it is toward. The pose of the object can be continuously tracked and updated until a grasping session finishes. Its processing pipeline is shown in Fig. 2. The imaging sensor captures images of the surrounding environment and recognizes the target object in the scene. The system then retrieves the corresponding 3D features of the recognized object from the feature database and matches the features with the detected object to extract position and orientation. The system then generates a 3d graphical model and overlays the model with the detected object in the image to visualize the result of the pose estimation. The 3D model also provides the dimensions and shape information. Finally, the inertial sensors track the movement of the prosthetic hand and keep updating the extracted object pose. Since the position, orientation, and shape of the object are all known by the prosthetic hand, it is possible to control the prosthetic hand in a fine way. For example, finger-level control can be achieved.

Basically, the processing pipeline includes three main stages: object recognition, object matching and object tracking.

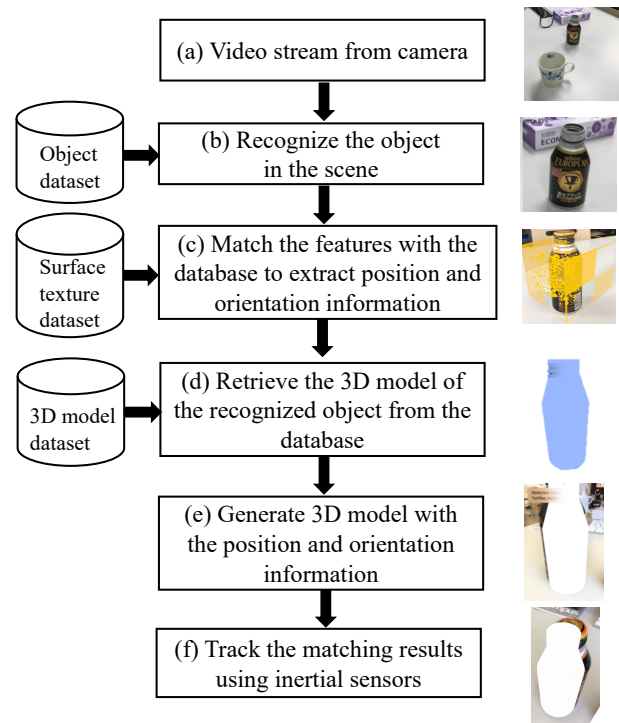


Fig. 2. Control Pipeline of the System.

Object recognition includes determining the grasp target object among several objects and cropping the target object from the original image for the convenience of feature matching. Object matching refers to matching the graphic 3D model with the real object in the image. It includes feature matching and pose estimation. Object tracking means the continuous tracking of the object pose without specifically estimating pose from every frame that camera captures. The three main stages are discussed sequentially in the following.

A. Object Recognition

Object recognition is a computer vision technique for identifying objects in images, outputting their categories and locations in an image. The state-of-the-art deep learning models can classify objects and regress their locations in a high degree of confidence [15]. In this study, object recognition is performed for two reasons. First, we want to

²www.bwsensing.com/product-5.html

³<http://www.emgpickups.com/>

identify the categories of the objects so that the corresponding 3D graphical models and features can be successfully retrieved from the database. Second, the locations of the objects are expected to be used to crop the object image from the original image to have a pure surface texture for feature matching.

The control system uses YOLO⁴ for object detection. YOLO is a real-time object detection system. On a GPU of Pascal Titan X⁵ it processes images at 30 FPS and has a mAP (mean average precision) of 57.9% on COCO⁶ which is a large-scale object detection dataset [10]. YOLO has some variations depending on the structure of the backbone and the regression/classification header. The main difference between these variations is that they have different number of convolutional layers. The selection of YOLO is to find a trade-off between accuracy and speed. In our study, the control system only needs to recognize the objects in one frame at the very beginning of a grasping session, so the detection speed is not the first thing to consider. We use the latest version of YOLO which has 53 convolutional layers in the backbone (feature extractor) to ensure the object recognition has an acceptable accuracy.

In the context of our study, a grasping session can only deal with one object. But in most cases, there is more than one object in camera view. It is necessary to determine the target among several objects. The control system follows a simple rule that the object which is closest to the center of the image is considered as the target object. This rule is also used in study [4]. Another thing that needs to be concerned is that sometimes the regression of the object location is not quite accurate, and some parts of the object may not be included in the cropped object image. It will have an effect on the feature matching. To avoid this, on the result of the location regression, we increase both the width and height size of the bounding box by 10%.

B. Object Matching

Object matching refers to matching the graphic 3D model with the real object in the image. The first step is to find the transformation matrix between the camera coordinate system (CS) and the world CS using feature matching. The second step is to map the points of a 3D graphical model from the world CS to the image CS. They are discussed in the following.

1) *Find transformation matrix:* The transformation matrix is represented by a 4×4 matrix. It contains rotation and translation information. With the transformation matrix, the relative position and orientation between the object and the camera is always known. So the transformation matrix step is the key to doing object matching. We use feature matching to calculate the transformation matrix.

Feature matching is a method that can find the corresponding points in the same object in two scenes. It detects object

surface texture features and compares them with pre-scanned object features (reference) to estimate the object position and orientation [16]. Given pre-scanned 3D feature points of an object, $(P_1, P_2, P_3, P_4, \dots)$ and a bunch of detected 3D feature points of the same object, $(P'_1, P'_2, P'_3, P'_4, \dots)$ but detected in run time. The correspondence can be found with matching features. The matching results are defined as $(P_1, P'_1), (P_2, P'_2), (P_3, P'_3), \dots, (P_n, P'_n)$. If we put the origin of the CS where pre-defined features points defined overlap with the origin of the camera CS and align their axis, the pre-scanned feature points will be represented in the camera CS. At the same time, the detected feature points (transformed points) are also in the camera CS. If the i -th pre-scanned feature point is defined as $P_i = (P_{ix}, P_{iy}, P_{iz}, 1)^T$, and its corresponding feature point detected in run time is $P'_i = (P'_{ix}, P'_{iy}, P'_{iz}, 1)^T$, the following equation shows their relationship.

$$\begin{bmatrix} P'_{ix} \\ P'_{iy} \\ P'_{iz} \\ 1 \end{bmatrix} = \begin{bmatrix} r_{00} & r_{01} & r_{02} & t_0 \\ r_{10} & r_{11} & r_{12} & t_1 \\ r_{20} & r_{21} & r_{22} & t_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_{ix} \\ P_{iy} \\ P_{iz} \\ 1 \end{bmatrix} \quad (1)$$

Thus, the transformation matrix is calculated by substituting a list of matched feature points to Eq. 1.

2) *Project to the image frame:* The second step is to map the points of a 3D graphical model from the world CS to the image CS using the transformation matrix calculated from the first step and the camera projection matrix (can be calculated from the calibration process) for visualization. See Fig. 3. A camera projection matrix is a 3×4 matrix which describes the mapping of a pinhole camera from 3D points in the world to 2D points in an image [17]. The visualization process uses Eq. 2 and Eq. 3 to project the 3D points to the 2D image frame. There are three CS involved: camera, image and world.

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = \begin{bmatrix} r_{00} & r_{01} & r_{02} & t_0 \\ r_{10} & r_{11} & r_{12} & t_1 \\ r_{20} & r_{21} & r_{22} & t_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \quad (2)$$

$$\begin{bmatrix} \mu \\ \nu \\ 1 \end{bmatrix} = \begin{bmatrix} fk & 0 & \mu_0 & 0 \\ 0 & fk & \nu_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} \quad (3)$$

C. Object Tracking

After we got the transformation matrix, we know the spatial relationship between the object and the camera. The spatial relationship (transformation matrix) needs to be updated when the hand prosthesis moves. And the movement of the prosthetic hand is tracked by the combination of the accelerometer and gyroscope. No matter where the camera moves with the prosthetic hand, the position and orientation object are always known to the control system. But this tracking cannot be kept in a long term due to the sensor noise and drifting. Since the reach-to-grasp movement is usually happened in a short term, it guarantees the accuracy of the tracking to some extent. If the translation and the

⁴<https://pjreddie.com/darknet/yolo/>

⁵<https://www.nvidia.com/en-us/geforce/products/10series/titan-x-pascal/>

⁶<https://cocodataset.org>

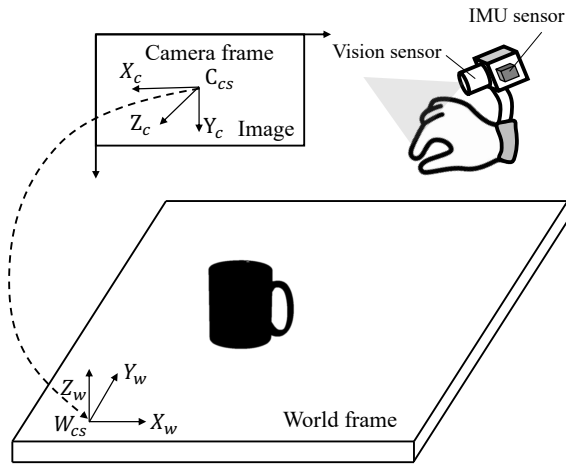


Fig. 3. Projection from world CS to camera CS

rotation of the camera can be tracked, we can update the transformation matrix using the translation and the rotation vector. The movement of the camera is represented by $(\alpha, \beta, \gamma, T_x, T_y, T_z)$, where α, β, γ define the roll, pitch, and yaw respectively while T_x, T_y, T_z define the translation in three different axes.

The camera motion tracking has several ways, for example, visual odometry, visual-inertial odometry or IMU sensors. The easiest way is to use IMU sensors, which usually features an accelerometer and a gyroscope. An accelerometer measures acceleration forces while a gyroscope measures orientation or angular velocity. They each serve to offset the other's noise and drift errors to provide more complete and accurate movement tracking [18]. The movement of the camera is calculated by integrating the accelerations and angular velocity using Eq. 4 and Eq. 5. Due to the noise introduced by the IMU sensor, the tracking accuracy can be only acceptable in a short term.

$$pos(t) = \int_0^t \int_0^\mu (acc(\mu) - acc(0)) d^2\mu \quad (4)$$

$$ori(t) = \lim_{h \rightarrow 0} \prod_{i=1}^{\lfloor t/h \rfloor} \Delta rot_{mat}((i-1)h, ih) \quad (5)$$

where h is the length of the time interval between two subsequent sample, Δrot_{mat} is the rotation matrix in each sample. Putting the Eq. 4 and the Eq. 5 together give the final formula for the camera pose:

$$pose(t) = \int_0^t \int_0^\mu (ori(\mu) acc(\mu) - acc(0)) d^2\mu \quad (6)$$

III. EXPERIMENTS

We evaluated the performance of the proposed control strategy from two aspects: object recognition and object matching/tracking.

A. Performance of the Object Recognition Model

The object detection model is trained from five common daily used objects: cup, bottle, spray bottle, ball and stapler. These five objects are selected because we use different postures to use/hold them. When the intersection over union (IoU) threshold is set to 0.5, the object detection network achieves 93.28% mAP. We used the trained object detection to detect a cup and a bottle on a table and then controlled the camera mounted on a prosthetic hand to perform a reach-to-object movement. The camera takes a bunch of images during the movement and these images are input to the object detection network. The detection results were reported. They are plotted in Fig. 4. As we can see from the figure, the detection accuracy in most of the reaching process is over 80%. The regressed locations included in detection results were further used to crop the object from the original pictures, which is shown in the right part of Fig. 4. The cropped image shows a better view of the surface pattern of the objects, which is good for feature detection and feature matching. But when the object is relatively far from the camera, the cropped object image is not clear enough. If the prosthetic hand detected more than one object in its view, the hand needs to determine which object is the grasp target. The simple rule is to find which object appears nearer to the center of the image.

B. Example of the Object Matching and Tracking

Most smartphones nowadays feature a camera as well as an IMU sensor. It is convenient to use the smartphone for a quick demonstration. In addition, the basic concept of the proposed method is very similar to an augmented reality (AR) application. The company of Apple and Google releases ARKit and ARCore⁷ library for developers to develop AR applications, they can be used to verify the proposed method.

In the case of an iPhone, ARKit performs features matching for estimating the transformation matrix and uses visual-inertial odometry for tracking the camera movement [19]. The visual-inertial odometry method first uses the phone's camera to identify interesting feature points and tracks how those points move over time. With a combination of the movement of these points and readings from the phone's inertial sensors, both the position and orientation of the phone are determined as it moves through space.

To verify the proposed method, two 3D models: a bottle and a cup are created first based on two real objects. We want to estimate the transformation matrix between the camera coordinate system and the object local coordinate system using feature matching (see Fig. 5), then project the 3D models to the real world to make them positioned and orientated the same as the real object. The camera movement is tracked by the IMU sensors using the transformation matrix, and the transformation matrix is updated in every image frame to reconstruct the AR scene.

The demonstration is shown in Fig. 6. To show that the orientation is also recognized, we make the bottle lie on the top

⁷<https://www.newgenapps.com/blog/arkit-vs-arcore-the-key-differences/>

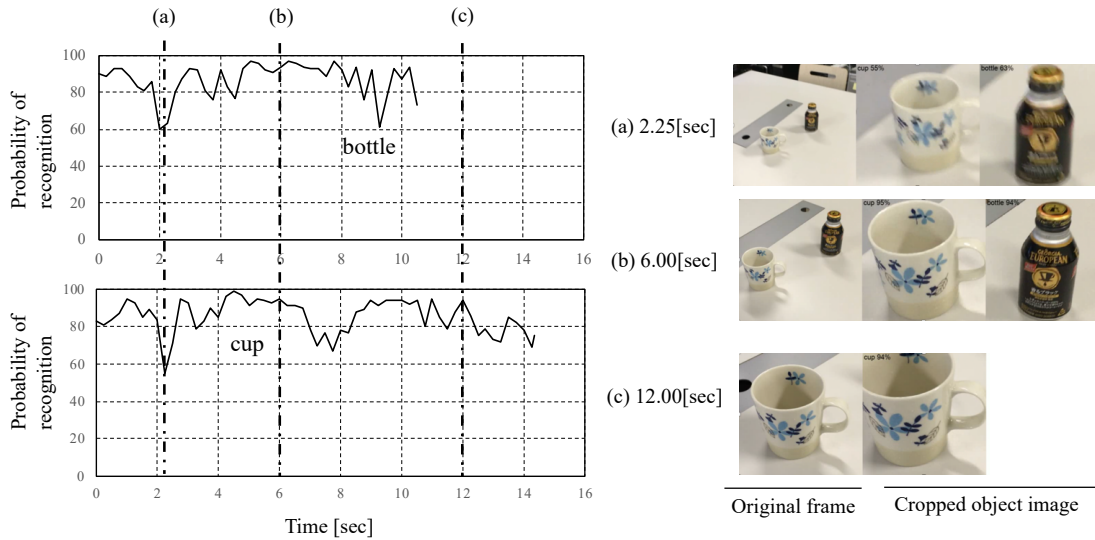


Fig. 4. Example of the Object Recognition.

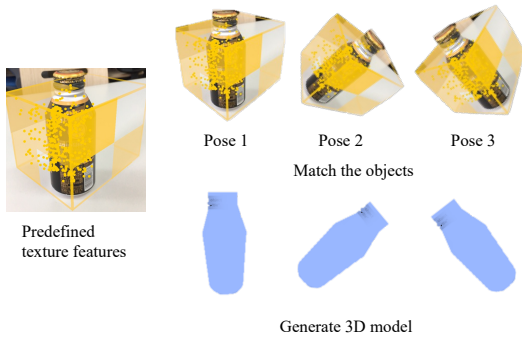


Fig. 5. Object Pose Estimation.

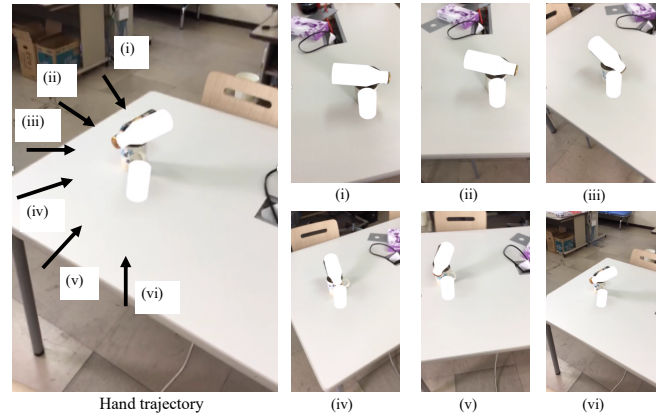


Fig. 6. Example of the Object Tracking.

of the cup. The white rigid body in the pictures is the 3D model created previously. After estimating the transformation matrix, the 3D models of the cup and the bottle are both projected to the 3D world scene to match their corresponding real objects. At the same time, the 3D world scene is again projected to a 2D image and shown in Fig. 6. We can find from the figure that the position and orientation of the cup and the bottle are both successfully estimated. Then, we try to change the view angle and move the camera farther from the object to make sure that the movement of the camera can be tracked. The scales and the perspective of the object are altered with the movement of the camera in the 2D image as the rendered 3D model exists in the real world, which proves that the hand is tracked properly.

IV. DISCUSSION

The experiment shows that it is possible to estimate the position and orientation of an object and track it in real-time using the sensor fusion. The transformation matrix represents the spatial relationship between the camera and the object. Estimating and tracking the transformation matrix is the key to constructing the spatial relationship. This relationship helps the prosthetic hands to comprehensively understand the

grasp scene and provides more evidence to control the hand prosthesis. It introduces many benefits. First, if we know the shape of an object as well as its 6D pose, the control can go in a very fine manner. Second, the timing to trigger a hand close movement in previous studies is usually determined by estimating the human intention from the EMG signals, but now it can be inferred from the transformation matrix since it represents the spatial relationship between the hand and the object. Third, the spatial awareness makes it possible for prosthetic hands to coordinate themselves.

However, during the test, we found that the orientation and position cannot be accurately estimated all the time. Some failure examples are shown in Fig. 7. The 3D model cannot perfectly match the real object in the AR scene. It may be caused for several reasons. But the drift from the IMU sensors is the biggest influencing factor. Drift is an ever-increasing difference between where the system thinks it is located and the actual location. Due to integration of a constant error in acceleration results in a linear error in velocity and a



Fig. 7. Some Tracking Failures.

quadratic error growth in position. It is hard to remove the drift completely, but we can make an effort to reduce the drift errors. Alternatively, we can use some markers with clear patterns to track the objects, like the ARuco markers used in AR applications [20]. But it is unrealistic to put markers around our living environment for detecting.

V. CONCLUSION

This study introduces a new control scheme to control the vision-based hand prosthesis by combining the camera with the IMU sensor and present a demonstration to verify the proposed control. The proposed method controls the hand prosthesis based on the construction and tracking of spatial relationship between the hand and the object. The spatial relationship is represented by the transformation matrix, which provides more evidence for controlling a dexterous hand prosthesis. But as shown in the experiment, the spatial relationship cannot be perfectly constructed using the introduced method. In the future, we would like to introduce the deep learning technique to detect key points of the object and construct the transform matrix based on these points.

ACKNOWLEDGMENT

This work was supported by JSPS KAKENHI Grant Number JP19K04296.

REFERENCES

- [1] S. Došen, C. Cipriani, M. Kostić, M. Controzzi, M. C. Carrozza, and D. B. Popović, "Cognitive vision system for control of dexterous prosthetic hands: experimental evaluation," *Journal of neuroengineering and rehabilitation*, vol. 7, no. 1, p. 42, 2010.
- [2] G. Ghazaei, A. Alameer, P. Degenar, G. Morgan, and K. Nazarpour, "Deep learning-based artificial vision for grasp classification in myoelectric hands," *Journal of neural engineering*, vol. 14, no. 3, p. 036025, 2017.
- [3] M. Markovic, S. Dosen, C. Cipriani, D. Popovic, and D. Farina, "Stereovision and augmented reality for closed-loop control of grasping in hand prostheses," *Journal of neural engineering*, vol. 11, no. 4, p. 046001, 2014.
- [4] Y. He, R. Shima, O. Fukuda, N. Bu, N. Yamaguchi, and H. Okumura, "Development of distributed control system for vision-based myoelectric prosthetic hand," *IEEE Access*, vol. 7, pp. 54 542–54 549, 2019.
- [5] J. DeGol, A. Akhtar, B. Manja, and T. Bretl, "Automatic grasp selection using a camera in a hand prosthesis," in *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2016, pp. 431–434.
- [6] N. Bu, Y. Bando, O. Fukuda, and K. Okumura Hiroshi, Arai, "A semi-automatic control method for myoelectric prosthetic hand based on image information of objects," in *Proceedings of the 22nd International Symposium on Artificial Life and Robotics*, 2017, pp. 23–28.
- [7] M. Esponda and T. M. Howard, "Adaptive grasp control through multi-modal interactions for assistive prosthetic devices," *arXiv preprint arXiv:1810.07899*, 2018.
- [8] M. Markovic, S. Došen, D. Popovic, B. Graitmann, and D. Farina, "Sensor fusion and computer vision for context-aware control of a multi degree-of-freedom prosthesis," *Journal of neural engineering*, vol. 12, no. 6, p. 066022, 2015.
- [9] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [10] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.
- [11] Y. Bando, N. Bu, O. Fukuda, and K. Okumura Hiroshi, Arai, "Object classification using a deep convolutional neural network and its application to myoelectric hand control," in *Proceedings of the 22nd International Symposium on Artificial Life and Robotics*, 2017, pp. 454–457.
- [12] R. Shima, Y. He, O. Fukuda, N. Bu, H. Okumura, and N. Yamaguchi, "Object shape classification using spatial information in myoelectric prosthetic control," *International Journal of Computer and Software Engineering*, vol. 3, 2018.
- [13] O. Fukuda, T. Tsuji, M. Kaneko, and A. Otsuka, "A human-assisting manipulator teleoperated by emg signals and arm motions," *IEEE transactions on robotics and automation*, vol. 19, no. 2, pp. 210–222, 2003.
- [14] A. Sanna and F. Manuri, "A survey on applications of augmented reality," *Advances in Computer Science: an International Journal*, vol. 5, no. 1, pp. 18–27, 2016.
- [15] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.
- [16] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, and D. Schmalstieg, "Real-time detection and tracking for augmented reality on mobile phones," *IEEE transactions on visualization and computer graphics*, vol. 16, no. 3, pp. 355–368, 2009.
- [17] W. Fang, L. Zheng, H. Deng, and H. Zhang, "Real-time motion tracking for mobile augmented/virtual reality using adaptive visual-inertial fusion," *Sensors*, vol. 17, no. 5, p. 1037, 2017.
- [18] P. Neto, J. N. Pires, and A. P. Moreira, "3-d position estimation from inertial sensing: Minimizing the error from the process of double integration of accelerations," in *IECON 2013-39th Annual Conference of the IEEE Industrial Electronics Society*. IEEE, 2013, pp. 4026–4031.
- [19] U. Dilek and M. Erol, "Detecting position using arkit ii: generating position-time graphs in real-time and further information on limitations of arkit," *Physics Education*, vol. 53, no. 3, p. 035020, 2018.
- [20] F. Romero-Ramirez, R. Muñoz-Salinas, and R. Medina-Carnicer, "Speeded up detection of squared fiducial markers," *Image and Vision Computing*, vol. 76, 06 2018.

AUTHORS' PROFILE



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 the Department of Information Science in Saga University, Saga, Japan. His main research interests are in human-machine interface.



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.



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 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>.