Visual Feedback Control of an Underactuated Hand for Grasping Objects of Varying Stiffness

Ryogo Kai[1], Yuzuka Isobe[1][0000−0001−6725−912X], Sarthak Pathak[2][0000−0002−5271−1782], and Kazunori Umeda[2][0000−0002−4458−4648]

[1] Course of Precision Engineering, Graduate School of Science and Engineering, Chuo University, 1-13-27 Kasuga, Bunkyo-ku, Tokyo, Japan
{kai,isobe}@sensor.mech.chuo-u.ac.jp
[2] Department of Precision Mechanics, Faculty of Science and Engineering, Chuo University, 1-13-27 Kasuga, Bunkyo-ku, Tokyo, Japan
{pathak,umeda}@mech.chuo-u.ac.jp

Abstract. This paper presents a novel method for grasping objects with varying stiffness using an underactuated hand and a stereo camera. In factories, robots are required to handle a wide variety of objects. Tasks such as grasping soft objects without causing damage are particularly important in industries like food processing. While many existing approaches equip robotic hands with sensors, such as force or pressure sensors, these methods are unsuitable for food items due to hygiene concerns. To address the challenges of grasping various objects without causing damage or dropping them, underactuated hands that can conform to object shapes have gained attention.
In this study, we propose a method for controlling an underactuated hand using only a stereo camera as an external sensor. First, the target object is detected using a background subtraction method. Next, the contact between the hand and the object is detected. Then, the object is grasped with appropriate force, calculated based on four elements: the centroid shifts of the hand and the object, the deformation rate of the object, and the occlusion rate of the hand. Finally, drop detection is performed to ensure the object is not dropped during pick-and-place tasks. Experiments were conducted using six different objects to validate the proposed method.

Keywords: Robot vision · Underactuatedhand · Manipulation.

1 Introduction

With the advancement of mass production of many models in small quantities, the demand for robots to handle a wide variety of objects is increasing. Therefore, robots need to grasp not only hard objects, such as metal products, but also soft objects, such as food items. Moreover, there are objects with hard surfaces

that may crush if the grasping force is too large. Similarly, if soft objects are not grasped with the proper force, it may cause the object to drop and lead to damage. Thus, it is necessary to grasp objects of varying stiffness with the proper force.

To grasp various objects, the underactuated hands are gaining attention[1]. These hands have a high degree of freedom, allowing them to deform according to the shape of the object. Thanks to this characteristic, the contact area between the hand and the object becomes larger, enabling stable grasping and allowing the handling of easily damaged objects. However, due to their adaptability, it is difficult to detect the grasping state, such as the contact area, grasping force, or potential damage and dropping of the object.

Many studies use tactile sensors and force sensors for grasping objects [2–4]. Some studies equip a small camera on the finger [5, 6]. By obtaining force information from sensors, it may be possible to grasp objects without applying excessive force. However, the contact area between the sensor and the object becomes small, which means the deformation of the entire object might not be detected, potentially resulting in damage to the object. Furthermore, there are issues such as the inability to detect the appropriate force due to sensor contamination, as well as the increased complexity of the system caused by the integration of sensors.

Instead of embedding sensors in the hand, some research uses a camera as an external sensor to observe the hand from the outside[7, 8]. These methods recognize the grasping state by using markers attached to the hand and the object. However, it is impossible to attach markers to food items or everyday products. In the previous research, we proposed visual feedback control method for underactuated hand [9]. However, this method requires the color of the object beforehand. In addition, because a monocular camera was used, only two-dimensional image information could be acquired, making it impossible to accurately detect crushed objects or falling objects.

In this paper, we propose a visual feedback control method for an underactuated hand using a stereo camera. Using the background subtraction method, the target object can be detected even when its color information is unknown. Additionally, utilizing depth information acquired by stereo camera, enabling the detection of the 3D deformation and movement of the object. To demonstrate the effectiveness of the proposed method, grasping experiments are conducted using objects with various colors and stiffness.

2 Visual Feedback Control Method

2.1 Overview of the method

The feedback control of the underactuated hand is performed using images obtained from a stereo camera. A two-finger underactuated hand, without any embedded sensors, is used. To control the underactuated hand, both the object and the hand are captured by the stereo camera.

An overview of the proposed method is shown in Fig. 1. In this method, four types of detection are introduced for the visual feedback control of the underactuated hand: object detection, contact detection, grasping state detection, and drop detection. First, the target object is identified using a background subtraction method. After object detection, the hand begins to close. Then, contact between the hand and the object is detected based on their contour information. Next, during the grasping process, the grasping state is observed. The grasping state is represented using four indices: the centroid shifts of the hand and the object, the deformation rate of the object, and the occlusion rate of the hand. Using these indices, the object is grasped with sufficient force without causing damage. Once the grasping is completed, the object is lifted. During the lifting process, droppage and rotation of the object are detected. By conducting these detections, the proposed method enables the stable grasping and transportation of objects with various sizes and stiffness without causing damage or dropping them.

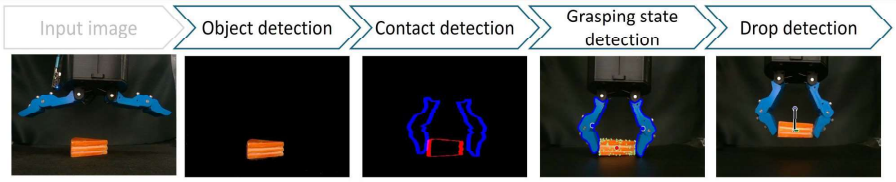

Fig. 1. Overview of the visual feedback control method

2.2 Object Detection

Object detection is a method to identify the target object from an image. In this study, we use background subtraction to detect the object to be grasped. An overview of the object detection process is shown in Fig. 2. First, as shown in Fig.2(a), an image is captured before the object is placed. Next, an image is captured after the object is placed(Fig.2(b)). Then, the two images are converted into grayscale image, and the difference between them is calculated(Fig.2(c)). From this difference image, contours are detected. Among these contours, the largest contour is considered as the target object for grasping. Based on the contour of the largest object in Fig.2(b), an HSV color threshold is calculated to extract only the region of the object. Using this HSV threshold, the target object is detected from the color image in Fig.2(b), as shown in Fig.2(d).

2.3 Contact Detection

Contact detection identifies the contact between the hand and the object based on images. An overview of the contact detection process is shown in Fig. 3.

Fig. 2. Example of object detection: (a) Background image, (b) Foreground image, (c) Difference image between (a) and (b), and (d) Extracted object.

Fig. 3(a) shows an input image. First, the input image is converted into the HSV color space. Using the HSV information of the object calculated in Section 2.2 and the HSV information of the hand which is known beforehand, each region is extracted respectively. Next, the contours of both the hand and the object are obtained from the extracted regions. Afterwards, both contours are drawn in different colors, as shown in Fig. 3(b). In Fig. 3(b), the hand's contour is drawn in blue, the object's contour is drawn in red. When the hand and the object come into contact, their contours overlap. In Fig. 3(c), the overlapping contours are represented in magenta. For each finger of the hand, if the overlap of the contours exceeds a certain threshold, it is detected as contact between the hand and the object.

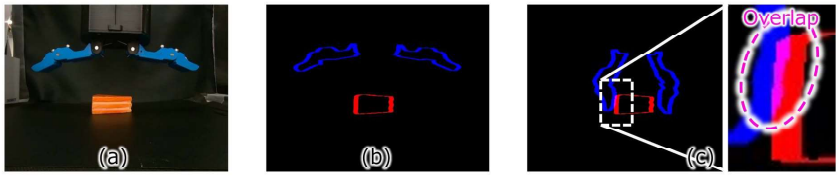

Fig. 3. Example of contact detection: (a) Input image, (b) Contours of the hand and the object, (c) Detection of contact through the overlap of both contours.

2.4 Grasp State Detection

To grasp objects without causing damage, the movement of the hand and deformation of the object are detected during grasping. Some objects, such as potato chips or box-shaped items, have hard surfaces but are fragile. On the other hand, such as tofu or balloons, have soft surfaces and are easy to be damaged. To adapt to objects of various stiffness, four indices are introduced as IfG (Index for Grasping). Indices described in Sections 2.4.1 and 2.4.2 are designed for grasping objects with hard surfaces, while those in Sections 2.4.3 and 2.4.4 are intended for grasping objects with soft surfaces. Grasping is considered complete if any of the IfG indices are satisfied.

IfG$_1$: Centroid Shift of The Hand IfG$_1$ involves calculating the movement of the hand. When grasping objects with hard surfaces, the hand is expected to stop moving after the contact with the object. If the object is brittle, continuing to apply force may cause it to break. Therefore, it is necessary to observe whether the hand is still moving.

Here, the movement of the hand is calculated based on the centroid information from each finger. In Fig. 4, the centroid of the left and right fingers are represented by blue circles. Assuming that the hand moves along the xy-plane as shown in the upper left of Fig. 4, the 2D movement of the centroid is calculated. Denoting the centroid of the left and right fingers as (x_l, y_l) and (x_r, y_r), respectively. Then, the movement of the left and right fingers, denoted as diff_l and diff_r, can be calculated by the following equations:

$$\mathrm{diff}_l = (x_l(t-1) - x_l(t))^2 + (y_l(t-1) - y_l(t))^2 \quad (1)$$

$$\mathrm{diff}_r = (x_r(t-1) - x_r(t))^2 + (y_r(t-1) - y_r(t))^2 \quad (2)$$

Here, t and $t-1$ represent the image frames. The index IfG$_1$ is then calculated as follows:

$$\mathrm{IfG}_1 = \mathrm{diff}_l + \mathrm{diff}_r \quad (3)$$

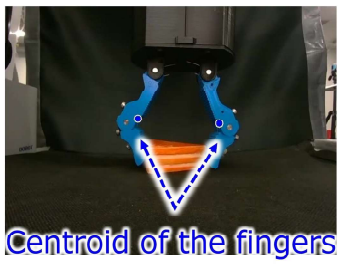

Fig. 4. IfG$_1$: Centroid Shift of The Hand

IfG$_2$: Centroid Shift of The Object The second index IfG$_2$ is the calculation of the movement of the object's centroid. When the object has a soft surface, applying grasping force causes the force to spread and deform the object. However, in the case of objects with a hard surface, the force does not escape towards the object but rather towards the hand, creating a torque in the passive joints of the hand. This results in rotational motion in the passive joints, which leads to the movement of the object. Therefore, it is necessary to observe the movement of the object. The movement of the object is determined by the centroid of the object in the image. In Fig. 5, the centroid of the object is shown as a red circle. When the centroid of the object is denoted as (x_{obj}, y_{obj}), IfG$_2$ can be calculated

using the following equation:

$$\mathrm{IfG}_2 = (x_{obj}(0) - x_{obj}(t)) + (y_{obj}(0) - y_{obj}(t)) \tag{4}$$

Here, t represents the image frame, and $t = 0$ denotes the first frame where the calculation of IfG_2 begins. This allows us to calculate the movement of the object during the grasping process.

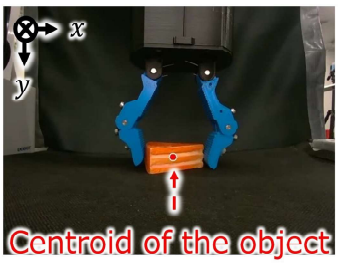

Fig. 5. IfG_2: Centroid Shift of The Object

IfG_3: Deformation of The Object The third index, IfG_3, represents the deformation of the object. For soft objects, excessive deformation may cause damage. Therefore, it is important to calculate the overall deformation of the object.

Previous studies have calculated the two-dimensional deformation of objects [9]. However, when grasping soft objects, it is also possible that the object deforms along the depth direction in the image. Thus, we calculate the deformation based on the object's contour and the three-dimensional movement of the points inside the contour. In Fig. 5, the object's contour and internal points are shown in green circles. When calculating the deformation, handling three-dimensional point clouds requires time-consuming processes such as point correspondence and downsampling. To address this problem, optical flow is applied using the Lukas-Kanade method [10]. The object's contour is obtained from the object region, and internal points are randomly distributed with a constant density. Using optical flow, the points are tracked, and three-dimensional deformation is calculated by acquiring depth information through a stereo camera. Represent point for optical flow as $\mathbf{P_i} = (x_i, y_i, z_i)$, and the total number of points N, IfG_3 is calculated using the following equation:

$$\mathrm{IfG}_3 = \frac{1}{N} \sum_{i=1}^{N} ||\mathbf{P_i(t)} - \mathbf{P_i(t-1)}|| \tag{5}$$

The displacement of each point on the object is normalized by dividing by the total number of point N. This allows to detect the deformation in objects of

different sizes, even if they belong to the same category. With this approach, the deformation of the object can be successfully detected.

Fig. 6. IfG$_3$: Deformation of The Object

IfG$_4$: Occlusion of The Object The fourth index, IfG$_4$, represents the percentage of the hand's area that is occluded by the object. Since only one camera is used, certain areas where the object may deform cannot be captured by the image. Therefore, by detecting the change in the area of the hand, it is possible to prevent excessive force from being applied to the object, even if it is occluded. In Fig. 7, the contour of the hand is shown in blue. The index IfG$_4$, which indicates the occlusion rate of the hand with respect to the object region, can be calculated using the following equation.

$$\mathrm{IfG}_4 = \frac{A_{hand}(0) - A_{hand}(t)}{A_{obj}(0)} \tag{6}$$

Here, A_{hand} is the area of the hand, A_{obj} is the area of the object, and t is the image frame. Since the force required for grasping differs depending on the size of the object. Therefore, evaluating only the change in the hand's occlusion area is not sufficient. Therefore, in IfG$_4$, the change in the hand's area is normalized by the object's area. This approach enables to adjust grasping force for objects of varying sizes within the same category.

2.5 Drop Detection

After grasping completion, the object lifting action is performed. During this process, drop detection is conducted. The drop detection consists of two parts: object drop detection and object rotation detection.

Object Drop Detection In object drop detection, not only the movement of the object outside the hand's workspace is considered, but also the state where the object is sliding while moving within the hand is detected. When

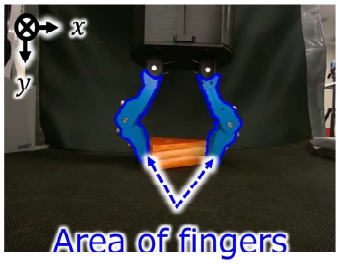

Fig. 7. IfG$_4$: Occlusion of The Object

an object is dropping, the relative position between the hand and the object changes. Therefore, the hand and object's centroid information is used for drop detection. The contour information is obtained from the input image, and the average centroid of the object and the two fingers of the hand is calculated. The distance between these two points is computed for each frame. In Fig. 8, the object's centroid is shown in red, the average centroid of the hand's two fingers is marked with a blue dot, and the line connecting the two points is drawn in black. The change in distance between the two points, d_{diff}, is calculated using the following equation.

$$d_{diff} = d(t) - d(0) \tag{7}$$

Here, $d(0)$ represents the distance before the arm movement, and $d(t)$ represents the distance during the movement. If d_{diff} exceeds a certain threshold, the object is considered as dropped. In this case, the value of $d(t)$ is updated by substituting it for $d(0)$, enabling continuous drop detection in subsequent frames.

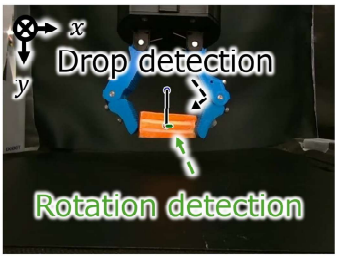

Fig. 8. Object drop and rotation detection

Object Rotation Detection In rotation detection, the sliding of the object is detected. In this method, principal component analysis (PCA) is applied to the object's contour, and the rotation of the object is determined by the change in

the angle between the first principal component and the horizontal axis. In Fig.8, the direction of the object's first principal component is shown with a green line. The change in angle θ_{diff} can be calculated using the following equation:

$$\theta_{diff} = \theta(t) - \theta(0) \tag{8}$$

Here, $\theta(0)$ represents the angle before the arm movement, and $\theta(t)$ represents the angle during the movement. If θ_{diff} exceeds a threshold, the object is detected to be sliding while rotating. At that time, $\theta(t)$ is updated by substituting it for $\theta(0)$, allowing for repeated rotation detection.

3 Experiments

3.1 Experimental Condition

To verify the effectiveness of the proposed method, grasping experiments were conducted. The experimental environment is shown in Fig. 9. In this experiment, the Yale OpenHand Project Model T-42 [10] was used for the underactuated hand, the DOBOT MG400 for robotic arm, and the Intel RealSense D405 for stereo camera. The objects used in the experiment are shown in Fig.11, and five

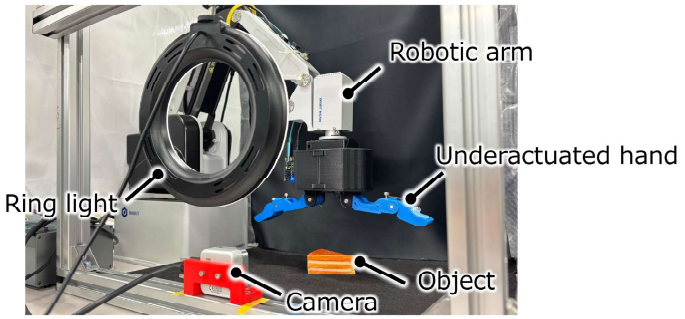

Fig. 9. Experimental environment

trials were performed for each object. The objects included two multi-colored toys and four actual food items, arranged to display the xy plane as shown in the top-left corner of Fig.11. After successfully grasping the objects, the robotic arm followed a predefined path, simulating a situation where disturbances were applied while lifting the object. The effectiveness of the method was evaluated using the success rate, which represents the proportion of successful experiments in which the object was grasped without dropping it. Additionally, for the food items shown in Fig. 11, the number of visible damages was evaluated.

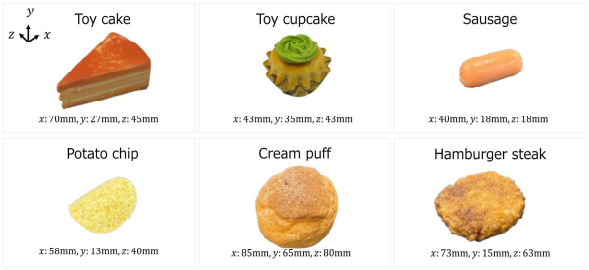

Fig. 10. Objects for the experiment

3.2 Experimental Results

The results are shown in Table 1. The successful grasping results are shown in Fig. 11. From the results, all objects were grasped successfully without being

Table 1. Experimental result

Object	success rate [%]	number of visible damages
Toy cake	100	-
Toy cupcake	100	-
Sausage	100	0
Potato chip	80	0
Cream puff	100	0
Hamburger steak	90	0

dropped, except for the potato chips and hamburger. This is due to the ability to adapt the grasping force according to the hardness of the object in the object state detection during grasping. Moreover, drop detection was well performed so as not to drop the object during the arm control. Furthermore, for the food items, no noticeable damage was observed visually during the grasping of any object. This suggests that the object deformation and hand occlusion detection in the object state detection were effective in preventing damage to the items.

Next, consider the failed attempts with the hamburger and potato chips. The examples of failure scenarios are shown in Fig.12. In Fig.12(a), with the potato chips, the object's posture changes during the grasping process. This is the factor that caused the failure of grasping the object due to insufficient force applied to the object. Therefore, it seems necessary to improve the method by observing the posture of the object during the grasp. Additionally, it is necessary to add control that performs a re-grasp if the force is not effectively added to the object.

In Fig. 12(b), with the hamburger, the object slips and drops during the lifting process. The hamburger was covered with a fatty layer, making it slippery, and it had some elasticity. The grasp was made near the fingertips, resulting in a small contact area. Due to the physical properties of the hamburger, the grasping

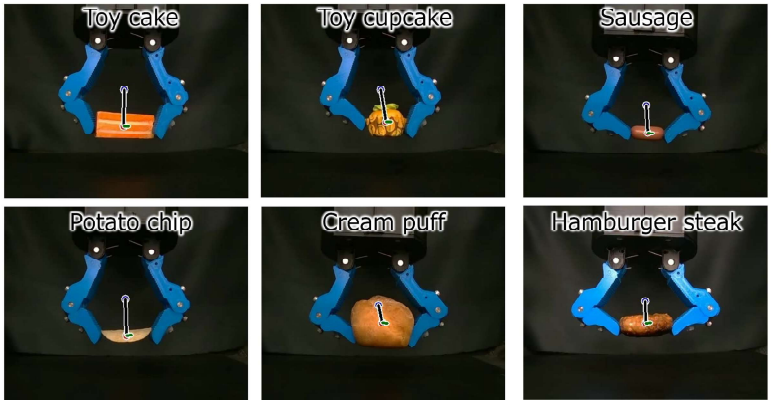

Fig. 11. Examples of successful grasps for objects

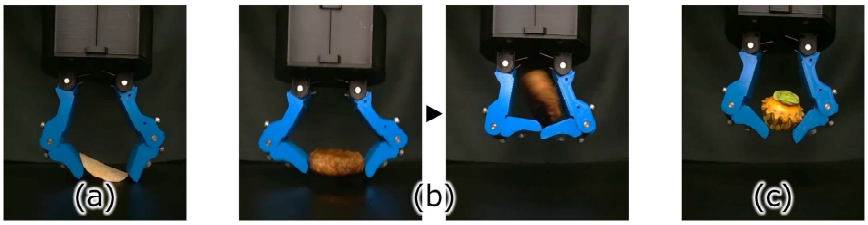

Fig. 12. Examples of failure in grasping

force was not sufficiently applied, causing the object to slip and fall. Therefore, in the future, controlling the grasping point by considering the direction and magnitude of the force applied to the object will help for a more stable grasp.

Additionally, there was also failure with toy cupcake, as shown in Fig.12(c). Although it did not fall, it seems to be in a state where it was not effectively grasped, almost leading to a drop. This indicates that, despite the underactuated hand having characteristics that allow it to conform to the object, it was unable to fully utilize those conforming capabilities. Therefore, it seems necessary to consider the deformation of the underactuated hand and calculate a grasping position that can conform to the shape of the object, enabling more stable grasping.

4 Conclusion

In this study, we proposed an visual feedback control method for grasping objects using an underactuated hand and a stereo camera. The proposed method enables the grasping of objects with various stiffness without using any internal sensors or attaching markers to the objects. To prevent excessive grasping force, we calculated the movements of the hand and the object, object deformation, and the occlusion of the hand. Additionally, during the lifting process by a robotic arm, object slippage was detected. In experiments, we successfully grasped objects without dropping them, except for potato chips and hamburgers. Furthermore, for food items, we confirmed that the objects could be grasped without causing visible damage.

In future work, we aim to improve the system by calculating the object's posture during grasping. This will allow regrasping when no force is applied to the object. Additionally, we plan to calculate the optimal grasping position by considering the forces applied to the object and the underactuated hand's adaptability to the object. This will enhance the stability of object grasping.

References

1. Birglen, L., Laliberté, T., Gosselin, C.: Underactuated Robotic Hands. Springer Tracts in Advanced Robotics **40**, (2008)
2. Azulay, O., Ben-David, I., Sintov, A.: Learning Haptic-based Object Pose Estimation for In-hand Manipulation with Underactuated Robotic Hands. arXiv:2207.02843, (2020)
3. Prado da Fonseca, V., Jiang, X., Emil, M. P., Eustaquio Alves de Oliveira, T.: Tactile Object Recognition in Early Phases of Grasping Using Underactuated Robotic Hands. Intelligent Service Robotics **15**, 513–525 (2022)
4. Lu, Z., Guo, H., Zhang, W., Yu, H.: GTac-Gripper: A Reconfigurable Under-Actuated Four-Fingered Robotic Gripper With Tactile Sensing. IEEE Robotics and Automation Letters, **7**(3), 7232–7239 (2022)
5. She, Y., Liu, S. Q., Yu, P., Adelson, E. H.: Exoskeleton-covered soft finger with vision-based proprioception and tactile sensing. In: 2020 IEEE International Conference on Robotics and Automation (ICRA), pp. 10075–10081. IEEE, Paris, France (2020)

6. Liu, S. Q., Yañez, L. Z., Adelson, E. H.: GelSight EndoFlex: A Soft Endoskeleton Hand with Continuous High-Resolution Tactile Sensing. In: 2023 IEEE International Conference on Soft Robotics (RoboSoft), pp. 1–6. IEEE, Singapore, Singapore (2023)
7. Andrew, S. M., Walter, G. M., Aaron, M. D.: Towards generalized manipulation learning through grasp mechanics-based features and self-supervision. IEEE Transactions on robotics, **37**, (2021)
8. Van, P. N., Quan Khanh, L., Yuzuru, T., Van Anh, H.: Wet Adhesion of Micro-Patterned Interfaces for Stable Grasping of Deformable Objects. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 9213–9219. IEEE, Las Vegas, NV, USA (2020)
9. Ryogo, K., Yuzuka, I., Sarthak, P., Kazunori, U.: Visual Feedback Control of an Underactuated Hand for Grasping Brittle and Soft Foods. In: 2024 IEEE International Conference on Robotics and Automation (ICRA), pp. 7332–7338. IEEE, Yokohama, Japan (2024)
10. Bruce D. L., Takeo, K.: An iterative image registration technique with an application to stereo vision. In: Proceedings of the 7th International Joint Conference on Artificial Intelligence, pp. 674-–679. Morgan Kaufmann Publishers Inc., Vancouver, BC, Canada (1981)
11. Raymond, R. M., Lael, U. O., Aaron, M. D.: A Modular, Open-Source 3D Printed U nderactuated Hand. In: 2013 IEEE International Conference on Robotics and Automation (ICRA), pp. 2737–2743. IEEE, Karlsruhe, Germany (2013)