biomimetics

MDPI

*Article*

# Bio-Inspired Proprioceptive Touch of a Soft Finger with Inner-Finger Kinesthetic Perception

**Xiaobo Liu [1,2], Xudong Han [1,2], Ning Guo [1,2], Fang Wan [1,3,*] and Chaoyang Song [2,4,*]**

[1] Shenzhen Key Laboratory of Intelligent Robotics and Flexible Manufacturing Systems, Southern University of Science and Technology, Shenzhen 518055, China
[2] Department of Mechanical and Energy Engineering, Southern University of Science and Technology, Shenzhen 518055, China
[3] School of Design, Southern University of Science and Technology, Shenzhen 518055, China
[4] Guangdong Provincial Key Laboratory of Human-Augmentation and Rehabilitation Robotics in Universities, Southern University of Science and Technology, Shenzhen 518055, China
* Correspondence: wanf@sustech.edu.cn (F.W.); songcy@ieee.org (C.S.)

**Abstract:** In-hand object pose estimation is challenging for humans and robots due to occlusion caused by the hand and object. This paper proposes a soft finger that integrates inner vision with kinesthetic sensing to estimate object pose inspired by human fingers. The soft finger has a flexible skeleton and skin that adapts to different objects, and the skeleton deformations during interaction provide contact information obtained by the image from the inner camera. The proposed framework is an end-to-end method that uses raw images from soft fingers to estimate in-hand object pose. It consists of an encoder for kinesthetic information processing and an object pose and category estimator. The framework was tested on seven objects, achieving an impressive error of 2.02 mm and 11.34 degrees for pose error and 99.05% for classification.

**Keywords:** object recognition; kinesthetic; humanoid finger; inner-finger vision; deep learning



**Citation:** Liu, X.; Han, X.; Guo, N.; Wan, F.; Song, C. Bio-Inspired Proprioceptive Touch of a Soft Finger with Inner-Finger Kinesthetic Perception. *Biomimetics* **2023**, *8*, 501. https://doi.org/10.3390/biomimetics8060501

Academic Editors: Yue Dong, Yao Li and Wenjie Lu

Received: 30 August 2023
Revised: 10 October 2023
Accepted: 17 October 2023
Published: 21 October 2023



## 1. Introduction

Humans exhibit various manipulative behaviors with the ability to detect the interaction behaviors of handled objects and hands. Visual information provides rich global features for humans to perceive objects' shapes. However, without visual information, humans can still assess object properties, such as size, shape, position, and orientation, using the sense of touch alone [1]. There are many receptors in the skin at different depths on human hands to perceive mechanical stimuli during the interaction. Those receptors, cutaneous and kinesthetic, empower humans to feel objects relying on the sense of touch [2]. The cutaneous sense is the modality that depends on physical contact between skin and objects and is better for feeling the object's pressure, vibration, and temperature. In contrast, the kinesthetic sense is the awareness of the position and movement of the body. It is better to feel the object's position and orientation from the receptors within the muscles, tendons, and joints [3]. Inspired by the kinesthetic sense, we present a soft finger with an embedded camera and a deep learning architecture for object recognition.

According to the structure of human hands, many methods have been proposed for hand pose estimation (HPE) problems [4–6]. As the shape of an object and the configuration of a hand (how many fingers are used to manipulate objects and the fingers' positions) are constrained by each other [7], some hand–object joint detection methods are proposed, which are called hand–object pose estimation (HOPE) [8–10]. Like humans manipulating objects with HOPE, object pose recognition is also a fundamental and challenging task in robotics.

For a manipulation task, perception of the environment and objects is essential [11]. Vision sensors are standard solutions to perceive the environment, and many methods

have been proposed for object localization and classification [12–15]. While deep learning significantly improves performance in object recognition problems, the inevitable occlusion is still challenging, especially in dexterous manipulation tasks. Even if we obtain an object pose with high precision before manipulation, the pose during manipulation in the gripper is still unknown due to the inherent uncertainties, tolerances, and noise in the robotic system [16].

Inspired by the HOPE problem, we try to solve the robot gripper–object pose estimation problem with gripper pose estimation. For fully actuated grippers, we can obtain the joints' angles of the gripper from motors and contact states from tactile and force sensors and estimate object pose and category with that information [17–19]. For under-actuated grippers, we need additional sensors to measure extra degrees of freedom (DoF), then estimate object pose and category [20–22]. To measure the contact information, GelSight [23], DIGIT [24], and FingerVision [25] are proposed. Those sensors consist of a transparent hard layer and a thin elastic layer and predict object shape and contact force from the deformation of the elastic layer. Limited by the thickness of the elastic layer, which is 2–3 mm, the sensors' deformation is small, limiting the measuring range and shape adaptation during grasping.

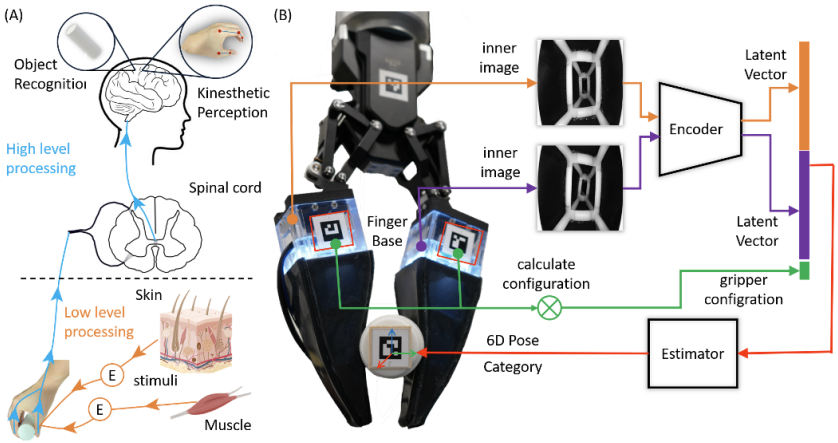



**Figure 1.** Overview of the bio-inspired finger and framework. (**A**) The raw stimulus is encoded with low-level processing and then transmitted to the central nervous system (CNS) for high-level processing, such as object recognition. (**B**) The bio-inspired fingers with flexible skeleton and silicon gel skin and a framework for object recognition: the raw images from fingers are encoded as latent vectors and then used for auxiliary tasks such as pose estimation and object classification.

Those methods mentioned above use multi-sensors in fingers for joints and tactile sensors in the fingertip for contact states and then estimate the objects' poses and categories with CAD models. In this article, we propose a soft, adaptive finger with an integrated camera to infer the finger deformation during interaction with objects, as shown in Figure 1. We mount the soft fingers on a gripper to enhance the adaption of the gripper and recognize handle objects with their proprioceptive sensing. Our method uses raw images to estimate objects' poses and categories with a camera and unknown CAD models. Instead of a two-stage method to recognize gripper state and object pose, our method is one-stage to recognize handled objects' poses and categories from the raw images. To simplify the training and enhance the reusability, we split the method into two parts: a feature extractor for interaction information embedding and a post-processor for further manipulation tasks. The feature extractor is an encoder–decoder architecture with ResNet block [26], and the post-processor is a multilayer perceptron (MLP) for classification and regression. The main contributions of this paper include the following: First, we design and fabricate a soft finger with an integrated camera inside for proprioception. Second, we propose a frame to extract and fuse the fingers' data for objects' states in a gripper. Finally, we test the effectiveness of the proposed method and obtain high accuracy in pose estimation and classification.

## 2. Materials and Methods

### *2.1. Design and Fabrication of the Soft Finger with Inner Vision*

In our previous work [27], we leveraged the soft finger with an AruCo marker inside to sense contact force and torque, which encodes the deformation of the finger. In this study, we introduce several improvements to the finger design, as shown in Figure 2:

- Added silicone skin on the finger to isolate the outside environment for a clear background.
- Added an LED light for illumination as the skin blocked the outside light.
- Removed the AruCo marker and used the finger's skeleton as a deformation feature.

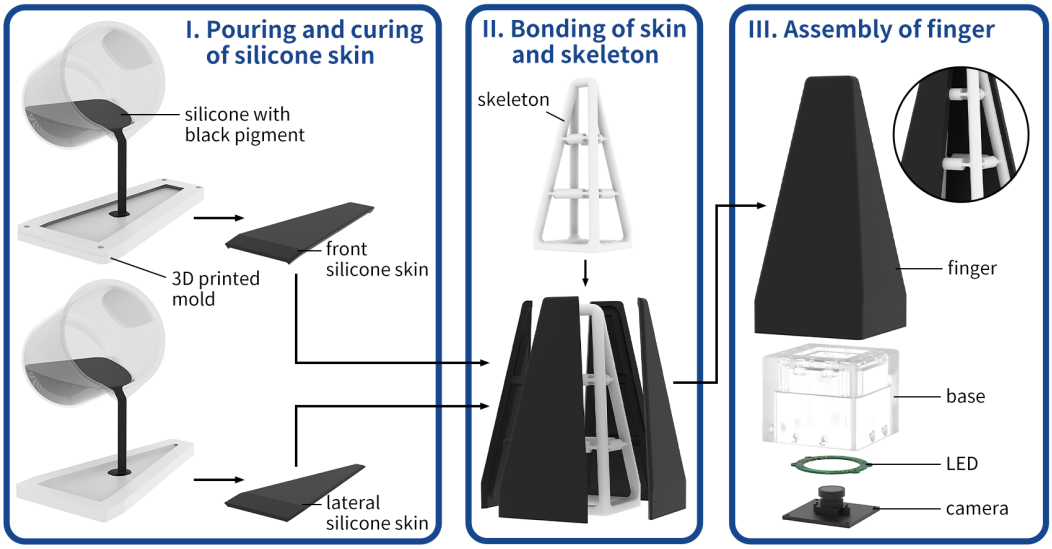



**Figure 2.** The soft finger's design and fabrication. (**I**) The fabrication process of the silicone skin; (**II**) attaching the skin to the basic finger skeleton; (**III**) the integrated finger with an LED and an inner camera.

As shown in Figure 2, this new design finger contains a finger skeleton with black skin, a base frame, an LED light, and a camera. The finger skeleton was fabricated with vacuum molding using polyurethane elastomers (Hei-cast 8400 from H&K). The three components were mixed in a 1:1:0 proportion to achieve 90A hardness with robust performance according to previous experience. Alternatively, other fabrication methods, such as fused deposition modeling (FDM) or stereolithography (SLA), could be cheaper. The size of the skeleton is 50 mm in bottom side length and 120 mm in height. The skin is made of Smooth-On Ecoflex$^{TM}$ 00-30 silicone rubber, and we added black pigment to change the color to block the ambient light effectively. Moreover, the silicone skin's thickness is 3 mm, fabricated individually, and attached to the finger skeleton with an adhesive Valigooo® V-80. The white LED light has enough luminous flux for the camera's exposure. The chosen camera is Chengyue WX605 from Weixinshijie, with a 640 × 360 resolution at 330 fps, and the lens is manually adjustable. When grasping objects, the deformed finger could be captured by the camera and encoded as the interaction information. Therefore, we use this proposed finger to recognize the outside object.

### *2.2. Framework for Handled Object Recognition with the Soft Finger*

In this section, we present a framework illustrated in Figure 3 to extract kinesthesia features and estimate the object state handled by the gripper. This framework contains two parts: an encoder–decoder architecture for feature extraction and two auxiliary multilayer perceptrons (MLP) for estimating the object's pose relative to the gripper's coordinate system and category, respectively. The input of the framework is two fingers' inner images and the gripper configuration.

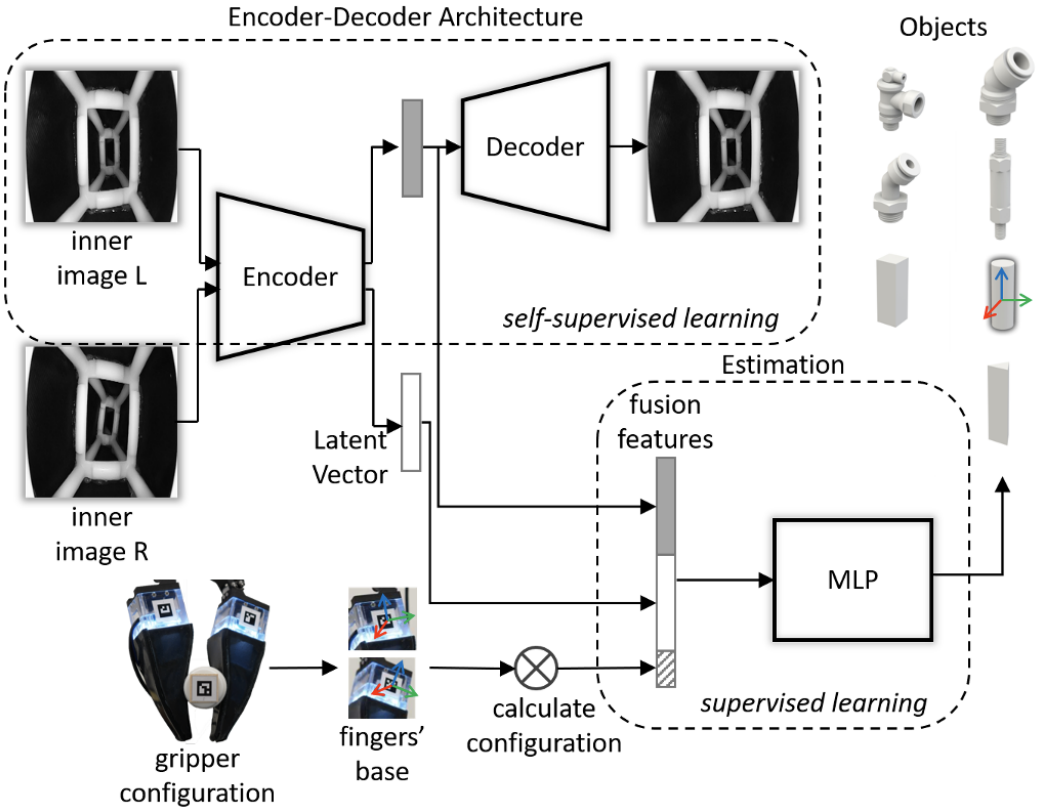



**Figure 3.** The architecture of the proposed framework: it takes two resized grayscale images and a gripper configuration as inputs and predicts the 6D pose and category of the object.

#### 2.2.1. Encoder–Decoder Architecture

Specific details of the encoder–decoder architecture are shown in Figure 4; the blue block is the encoder, the green block is the decoder, and the yellow vector is the extracted latent vector. The encoder–decoder architecture is a fully convolutional topology and takes a resized grayscale image $I = \mathbb{R}^{1\times320\times320}$ as input. It extracts the features representing the finger's deformation and outputs an N-dimensional vector; the decoder reconstructs the image from the feature vector.

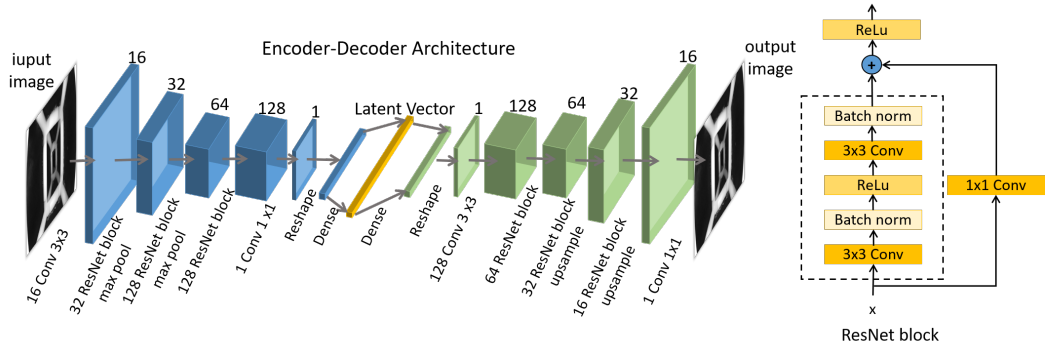



**Figure 4.** The encoder–decoder architecture of the feature extraction: one resized grayscale image as inputs and the same size image as output.

The basic blocks of the encoder–decoder architecture are $3 \times 3$ convolution and ResNet block for extracting features and $1 \times 1$ convolution for compressing features. The dense layer is set to change the latent vector's dimensions and explore the feature dimensions' effect on recovering the image.

Define the input as $I$, the encoder function as $E$, the latent vector as $V$, the decoder function as $D$, and the output as $Z$. The encoder–decoder architecture can be described as

$$V = E(I), \tag{1}$$

$$Z = D(V), \tag{2}$$

$$(\hat{\theta}_e, \hat{\theta}_d) = \underset{\theta_e, \theta_d \in \Theta}{arg\,\min} Loss(Z, I) \tag{3}$$

where $\hat{\theta}_e, \hat{\theta}_d$ are the well-trained encoder and decoder parameters, and $Loss$ is the loss function between $Z$ and $I$.

#### 2.2.2. Pose Estimation and Classification

After extracting the latent feature $V$, we designed two MLP models to estimate the object's pose and category as shown in Figure 5. These two models have the same inputs, and the output of the regression model is a 6D pose. In contrast, the output of the classification model is seven classes with a softmax activation function. In this article, the input vector is 129 dimensions, aggregating the two fingers' feature $V$ and gripper configuration. In the follow-up work, we set the dimension of $V$ as 64 and the dimension of the gripper configuration as one since the gripper we used is one degree of freedom (DoF), so the input vector is $64 \times 2 + 1 = 129$ dimensions. The regression model consists of five hidden layers with 200, 200, 100, 100, and 100 neurons, with an activation function rectified linear unit (ReLu) [28] and batch normalization [29]. The classification model consists of three hidden layers with 200, 200, and 100 neurons, with activation function ReLu.

Define two images taken from the inner cameras of the fingers as $I^L = \mathbb{R}^{C \times H \times W}$, $I^R = \mathbb{R}^{C \times H \times W}$ with height $H$ and width $W$, gripper configuration as $G_c$, regression model as $F_r$, classification model as $F_c$, object 6d pose as $S_p$, and object category as $S_c$, and the two MLP models are described as

$$V_L, V_R = E(I_L), E(I_R), \tag{4}$$

$$V_{aggregation} = Func(V_L, V_R, G_c), \tag{5}$$

$$S_p = F_r(V_{aggregation}), S_c = F_c(V_{aggregation}), \tag{6}$$

Here, $Func$ is a function to combine the vectors in order.

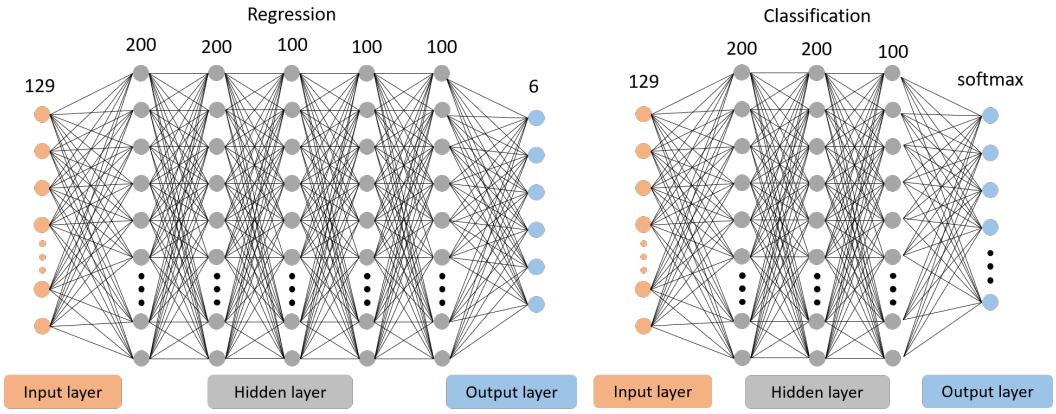



**Figure 5.** Two MLP models of object recognition. Left: a regression model for 6d pose estimation. Right: a classification model for object categories.

### *2.3. Data Collection and Training Setups*

#### 2.3.1. Data Collection Setup

We built an experimental platform to collect training data efficiently for training the framework above, as shown in Figure 6. The designed fingers are mounted on a DH-Robotics AG-160-95 adaptive gripper to replace its tips, and we pasted AruCo codes on the

fingers and grippers to represent their poses. An extra camera is mounted on an optical breadboard to collect the AruCo marker poses, and two cameras in the fingers collect the interaction deformations. The AruCo markers are 4 × 4 squares of 16mm width with different indexes and are detected by OpenCV [30]. The outside camera's resolution is set at 1920 × 1080 to increase the detection success rate and precision of AruCo markers detection. The inner camera's resolution is 640 × 360 and resized to 320 × 320 to decrease the model's size and prediction time.

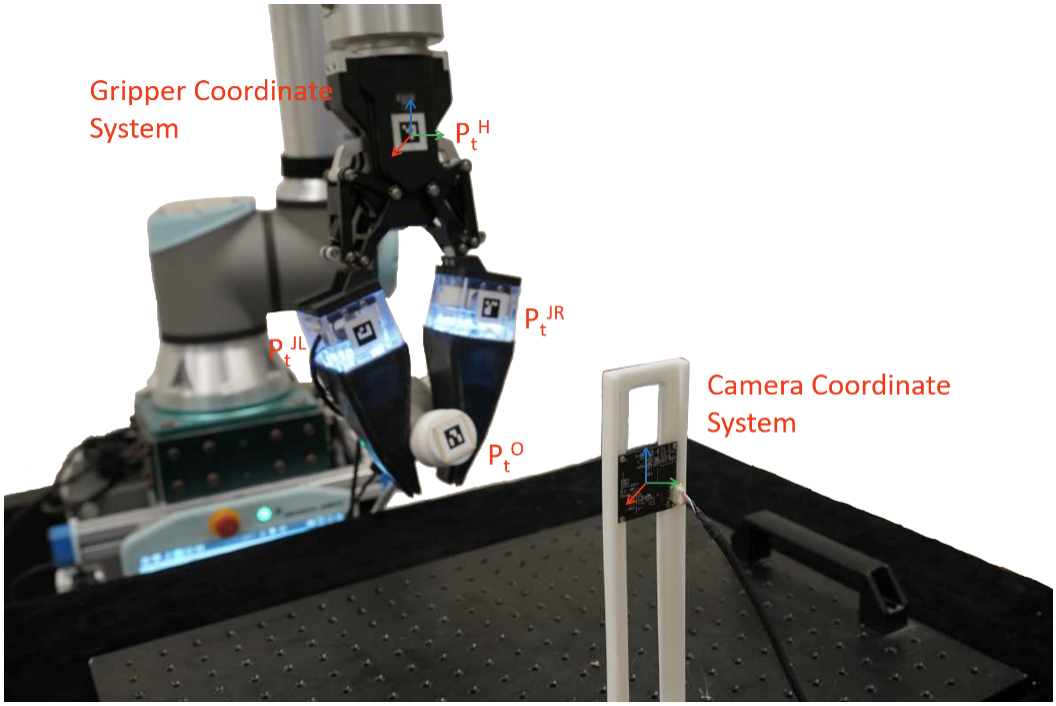
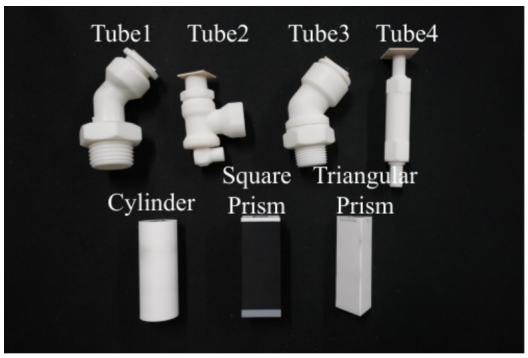



**Figure 6.** Date collection setup. Four markers are attached to the gripper, fingers, and object. An outside camera monitors the four markers for the object's pose; simultaneously, fingers' deformations are captured by two inner cameras.

Referring to article [31], we chose the McMaster dataset (https://www.mcmaster.com accessed on 20 June 2023) as our test objects. In addition, we also chose three basic geometric solids. All objects are resized to adjust the gripper width and 3D-printed for final usage as shown in Figure 7. When collecting data, we set the gripper force mode and control the gripper width to grasp the object, then leave the gripper static and shake the object manually to collect the object poses. Due to the finger's adaptation, the gripper width does not need to precisely match the objects' size, which is predefined at 15 mm. We collected 5000 samples for each object.




| object | size (mm) | weight (g) |
|---|---|---|
| Tube1 | 125×85×55 | 153 |
| Tube2 | 95×70×35 | 92 |
| Tube3 | 115×80×50 | 118 |
| Tube4 | 145×20×20 | 57 |
| Cylinder | 100×40×40 | 73 |
| Square Prism | 100×40×40 | 73 |
| Triangular Prism | 100×40×40 | 36 |

**Figure 7.** Test objects and their properties. (**Left**): 3D-printed objects. (**Right**): objects' sizes and weights.

After collection, all poses are transferred to the gripper coordinate system for standardization. Define $\mathbf{P} = (x, y, z, rx, ry, rz)$ as a pose, where $(x, y, z)$ is translation and $(rx, ry, rz)$ is orientation. Instead of using the object CAD model, we use the relative change to represent the object's pose without the object model. Define reference pose $\mathbf{P}_0 = (x_0, y_0, z_0, rx_0, ry_0, rz_0)$, current pose $\mathbf{P}_t = (x_t, y_t, z_t, rx_t, ry_t, rz_t)$ at time $t$, and the translation matrix $\mathbf{M}_t = [R|T]$, so

$$\mathbf{P}_t = \mathbf{P}_0 \mathbf{M}_t, \tag{7}$$

and we use $\mathbf{M}_t$ to represent the current object pose.

The left superscript $G$ and $C$ represent the gripper and camera coordinate system variables. $G$ is the gripper coordinate system, and $C$ is the camera coordinate. The in-camera coordinate system, the gripper pose, gripper configuration, and object pose indicated by the Aruco marker attached are ${}^{C}\mathbf{P}_t^O$, gripper poses ${}^{C}\mathbf{P}_t^H$, gripper joint poses ${}^{C}\mathbf{P}_t^{JL}$ and ${}^{C}\mathbf{P}_t^{JR}$ in time $t$.

The transfer to gripper coordinate system is as follows:

$$ {}^{C}\mathbf{P}_t^O = {}^{C}\mathbf{P}_t^H \cdot {}^{G}\mathbf{P}_t^O, \tag{8}$$

$$ {}^{G}\mathbf{P}_t^O = [{}^{C}\mathbf{P}_t^H]^{-1} \cdot {}^{C}\mathbf{P}_t^O, \tag{9}$$

$$ {}^{G}\mathbf{P}_0^O = [{}^{C}\mathbf{P}_0^H]^{-1} \cdot {}^{C}\mathbf{P}_0^O, \tag{10}$$

$$ {}^{G}\mathbf{P}_t^O = {}^{G}\mathbf{P}_0^O \cdot {}^{G}\mathbf{M}_t, \tag{11}$$

In the gripper coordinate system, the object transfer pose ${}^{G}\mathbf{M}_t$ is

$$\begin{aligned} {}^{G}\mathbf{M}_t &= [{}^{G}\mathbf{P}_0^O]^{-1} \cdot {}^{G}\mathbf{P}_t^O \\ &= [[{}^{C}\mathbf{P}_0^H]^{-1} \cdot {}^{C}\mathbf{P}_0^O]^{-1} [[{}^{C}\mathbf{P}_t^H]^{-1} \cdot {}^{C}\mathbf{P}_t^O] \end{aligned} \tag{12}$$

The collected dataset comprises seven objects and 5000 samples per object, each consisting of two inner images, four poses from the outside camera, and the objects' categories. The resolution of the inner images is the same and is 640 × 360, and resized to 320 × 320 for input, and the values are normalized to 0–1. The objects' pose distributions are shown in Figure 8.

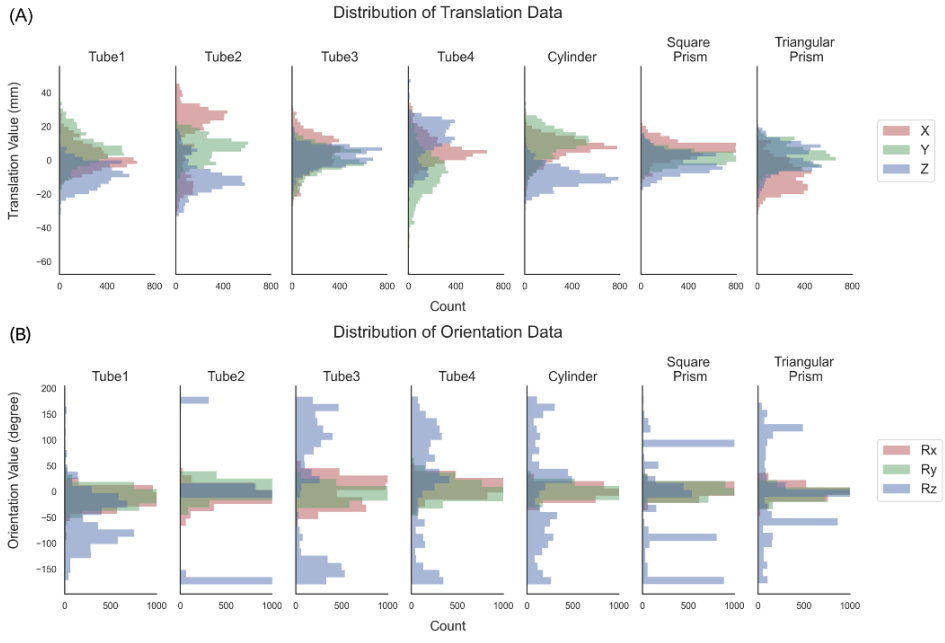



**Figure 8.** Distribution of the dataset. All data are exhibited in the gripper coordinate system: (**A**) translation distribution and (**B**) orientation distribution.

#### 2.3.2. Network Training Setup

To improve the reusability and expansibility of the network, we trained the encoder–decoder reconstruction and the auxiliary tasks in two stages using the dataset collected in the previous section.

In the first stage, the encoder–decoder reconstruction is self-supervised learning. The dataset is randomly split into 8:2; 56,000 images are used for training, and 14,000 are used for evaluation. It was trained with a batch size of 32 using an Adam optimizer with a learning rate of 0.001 on mean squared error loss (MSELoss). The latent vector $V$ is set to 8, 16, 32, 64, 128, and 256 to determine the best network configuration. The training epoch is set to 200, and we save the weights with the lowest training loss.

We froze the encoder's weights in the second stage and only trained the following auxiliary tasks. For the regression model, we trained an MLP model for each object. Using the split dataset above, 4000 samples are used for training, and 1000 are used for evaluation for each object. The batch size is 32, the optimizer is Adam optimizer, and the learning rate is 0.001. As the 6D pose consists of two parts, translation, and orientation, and we define the training loss in Equations (13)–(15), where $L_t$ is translation loss and $L_r$ is orientation loss. The hyper-parameters $\alpha$ and $\beta$ are set to 0.01 and 10, while the translation unit is millimeters and rotation is radians. The training epoch is set to 100.

$$L_t = \frac{1}{3}\sum_{n=1}^{3}(x_n^t - \hat{x}_n^t)^2, \tag{13}$$

$$L_r = \frac{1}{3}\sum_{n=1}^{3}(x_n^r - \hat{x}_n^r)^2, \tag{14}$$

$$L = \alpha L_t + \beta L_r. \tag{15}$$

For the classification model, we trained an MLP model for all objects together. Using the split dataset above, 28,000 samples are used for training and 7000 for evaluation. The batch size is 256, the optimizer is the Adam optimizer, the learning rate is 0.001, and the training loss is cross-entropy loss. The training epoch is set to 100.

## 3. Results and Discussion

### *3.1. Dimension of the Latent Vector*

To find an optimal dimension of the latent space, we varied the dimension of the latent vector and compared the reconstruction errors using the same training and validation dataset. The dimension of the latent vector is set to 8, 16, 32, 64, 128, and 256, and the corresponding results are shown in Table 1.

**Table 1.** Effect of the latent vector dimension.

| | **Latent Vector Dimension** | | | | | |
|---|---|---|---|---|---|---|
| | **8** | **16** | **32** | **64** | **128** | **256** |
| Normalized MSELoss | 1.37 | 1.74 | 1.36 | 1.09 | 1.35 | 1 |
| Parameters (M) | 0.57 | 0.67 | 0.88 | 1.29 | 2.11 | 3.74 |

We scaled all losses such that the loss of 256-dimensional latent space was one. As the dimension increases, the reconstruction loss decreases, and the number of parameters of the model increases. To balance the precision and computational efficiency of the auto-encoder, we chose the 64-dimensional latent space, whose loss is comparable to the 256-dimensional space with only 24% of the number of parameters.

### *3.2. Quantitative Evaluation of Object Recognition*

In this section, we report the accuracy of pose estimation and classification. The translation error is measured by the Euclidean distance $\|p_{est} - p_{gt}\|_2$ between the esti-

mated position $p_{est} = (x, y, z)_{est}$ and the ground truth position $p_{gt} = (x, y, z)_{gt}$ [32]. The orientation error $|\alpha|$, measured by an absolute angle error, is computed as

$$2\cos|\alpha| = \mathrm{Tr}(R_{gt}^{-1} R_{est}) - 1, \tag{16}$$

where $R_{gt}$ and $R_{est}$ are the estimated and ground truth rotation matrices, and Tr is the trace of the matrix.

As shown in Figure 9, the translation and orientation errors are significantly different for different objects. The mean translation error of each object is between 2.02 mm and 4.00 mm, and the mean orientation error is between 11.34 degrees and 31.87 degrees. Object tube1 has the slightest translation error of 2.02 mm and the slightest orientation error of 11.34 degrees. The object cylinder has the most significant translation and orientation error of 4 mm and 31.87 degrees. The decompositions of the errors are shown in Figure A1.

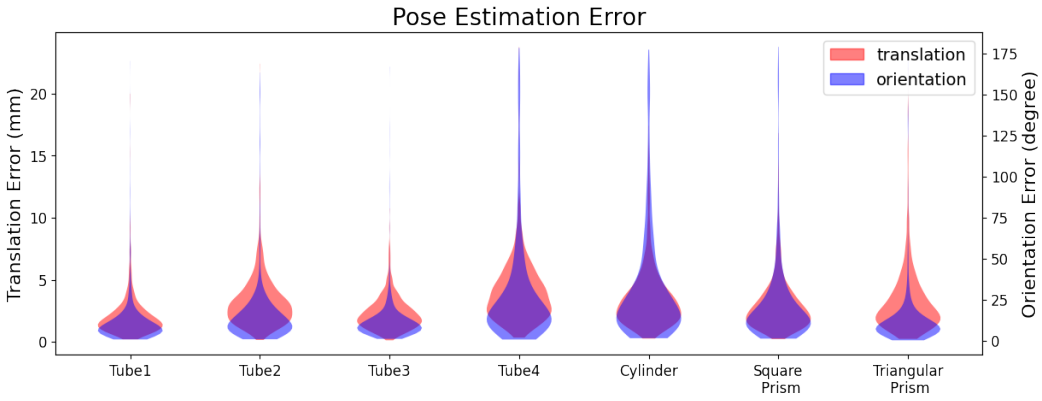



**Figure 9.** The histogram of pose estimation errors of each object. Translation error is the Euclidean distance, and rotation error is the absolute orientation error $|\alpha|$.

Despite lacking a standard experiment setup for in-hand object pose estimation, some work has still been explored. The authors of [33] used RGB cameras and GelSights to estimate in-hand object pose with 15 mm accuracy in position and 15 degrees accuracy in orientation. The authors of [34] fused vision and tactile data to obtain 2.99 mm accuracy in position and 8.074 degrees accuracy in orientation. The authors of [21] presented a method based on a bootstrap particle filter to estimate the pose with 2.14−7.31 mm accuracy in position and 0.812−3.392 degrees accuracy in orientation due to the ground truth not detecting almost any rotation. Comparing the state-of-the-art methods, our method obtains a comparable result with a small translation error.

The inner camera can only perceive the objects' geometric shape and size as the skin isolates the ambient environment. Objects with complex shapes provide abundant shape features and improve the pose estimation accuracy. On the contrary, the geometric shapes of tube4, cylinder, square prism, and triangular prism are more similar to a cylinder, resulting in more significant orientation errors among the seven objects. Those objects are symmetrical, but the features around the symmetry axis lack uniqueness, increasing the difficulty of orientation estimation. The objects' cross-section shape influences translation error. The columnar objects (tube4, cylinder, and prisms) have a more significant translation error due to their similar cross-section shape among the seven objects.

Objects' geometric and texture features are essential elements, and the designed finger is limited to obtain the texture as the black coat, influencing the pose estimation precision. A prominent method to improve the precision is to add more features, such as mounting a camera on the gripper or changing the transparent black skin to obtain image features. More features increase the complexity of the device and algorithm but improve performance.

As the objects have unique 3D shapes and sizes, we obtain a high classification accuracy of 99.05%, as shown in Figure 10. With the proposed method, we can estimate the handled object's category with a high accuracy proprioceptive touch of the soft fingers.

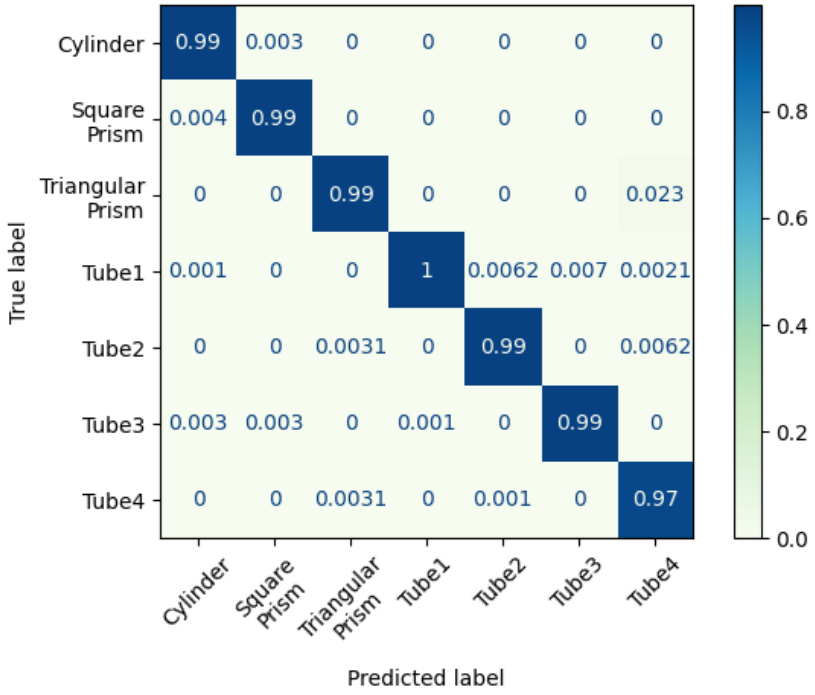



**Figure 10.** Confusion matrix for object classification.

### *3.3. Reusability and Expansibility of the Framework*

As described in the framework, the encoder–decoder architecture reduces the tactile feature dimensions and unites their format for different types of sensors. This makes the tactile information compact and simplifies the processing flow. For other sensors such as GelSight [23], BioTac (https://syntouchinc.com accessed on 19 October 2023) and magnetic skin [35], the different sensing information can also be represented as a latent vector with a convolutional neural network, graph neural network, or other methods according to the data structure.

Then, the extracted tactile features are fused depending on the gripper configuration. In this paper, the fusion features combine two-finger images and gripper configuration and are an input of the auxiliary tasks. For an N-finger gripper, we first extract the tactile information of each finger, then fuse each finger's features and the gripper configuration, such as joint rotation angles. The gripper configuration represents the joint's spatial position and can be described as a base pose and the DoF of each finger. As shown in this article, we use AruCo markers to monitor the finger base pose and tactile features of the soft fingers to represent the finger's DoF, which is independent of hardware.

Finally, the fused features are used for downstream tasks. We demonstrate two basic examples, pose estimation and classification of the handled object, and obtain sound results. We can quickly adapt the frame to other tasks using the same fusion features. Benefitting from the modular design of the framework, we can extract the tactile features independently, fuse them according to the configuration of the hand, and feed them to different auxiliary task models to complete manipulation tasks; this framework applies to scenarios with multi-sensor, multi-gripper, and multi-tasking.

## 4. Conclusions

This paper presents a bio-inspired, soft proprioceptive sensor and a framework for object pose estimation and classification. The proposed soft proprioceptive sensor can be extended to different manipulators, providing extra shape adaptation and interaction information. Based on this sensor, we propose an extendable architecture to extract the tactile information and estimate the handled object's state. This method achieves a high accuracy of 2.02 mm in translation, 11.34 degrees in orientation, and 99.05% classification accuracy for object classification with an unknown CAD model.

The interaction information is extracted from finger deformations when grasping objects. The pure black skin on the soft finger loses texture features, and the skeleton filters small shape features due to its smooth deformation, as shown in Figure A2. Those properties limit the soft finger to perceiving small objects and distinguishing similar-shaped objects. For unseen objects, our method can not be used directly. Current state-of-the-art pose estimation methods can only handle previously trained objects [36]. Instead of predicting pose directly, [36] proposed a learning-based method that finds corresponding points between an unseen object and an RGBD image, which can be transferred to non-learned objects. However, it depends on the object 3D model and the point cloud of the scene. Our method does not need an object model and uses RGB images only, and it cannot be transferred to non-learned objects easily. For an unseen object, we need to collect data and train the MLP again.

Future work will explore the transferability of this method on different tactile sensors and grippers. This framework provides a uniform feature extractor for different types of tactile sensor information and an extendable structure for different grippers. Meanwhile, more manipulation tasks can be involved with this method.

**Author Contributions:** Conceptualization, X.L; methodology, X.L. and X.H. and N.G.; software, X.L.; validation, X.L.; formal analysis, X.L.; investigation, X.L.; resources, F.W. and C.S.; data curation, X.L.; writing—original draft preparation, X.L.; writing—review and editing, F.W. and C.S.; visualization, X.L. and X.H. supervision, F.W. and C.S.; project administration, F.W. and C.S.; funding acquisition, F.W. and C.S.; All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was partly supported by the Ministry of Science and Technology of China [2022YFB4701200], the National Natural Science Foundation of China [62206119, 52335003], Shenzhen Science and Technology Innovation Commission [JCYJ20220818100417038, ZDSYS20220527171403009, SGDX20220530110804030], and Guangdong Provincial Key Laboratory of Human Augmentation and Rehabilitation Robotics in Universities.

**Institutional Review Board Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Appendix A

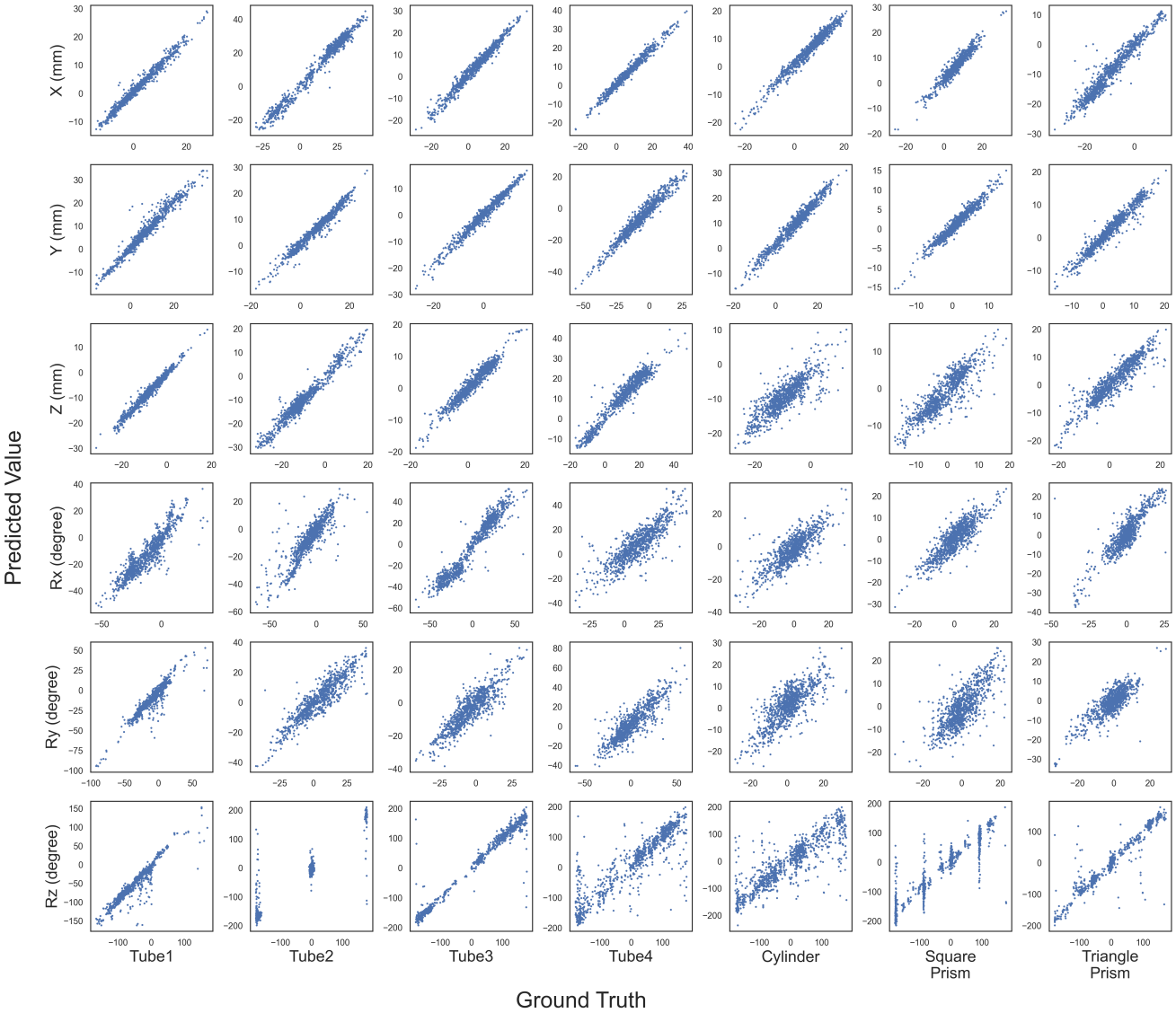



**Figure A1.** Results of the pose estimation for test objects. Six rows correspond to positions and orientations, and seven columns correspond to seven objects. The x-axis is the ground truth and the y-axis is the predicted values.

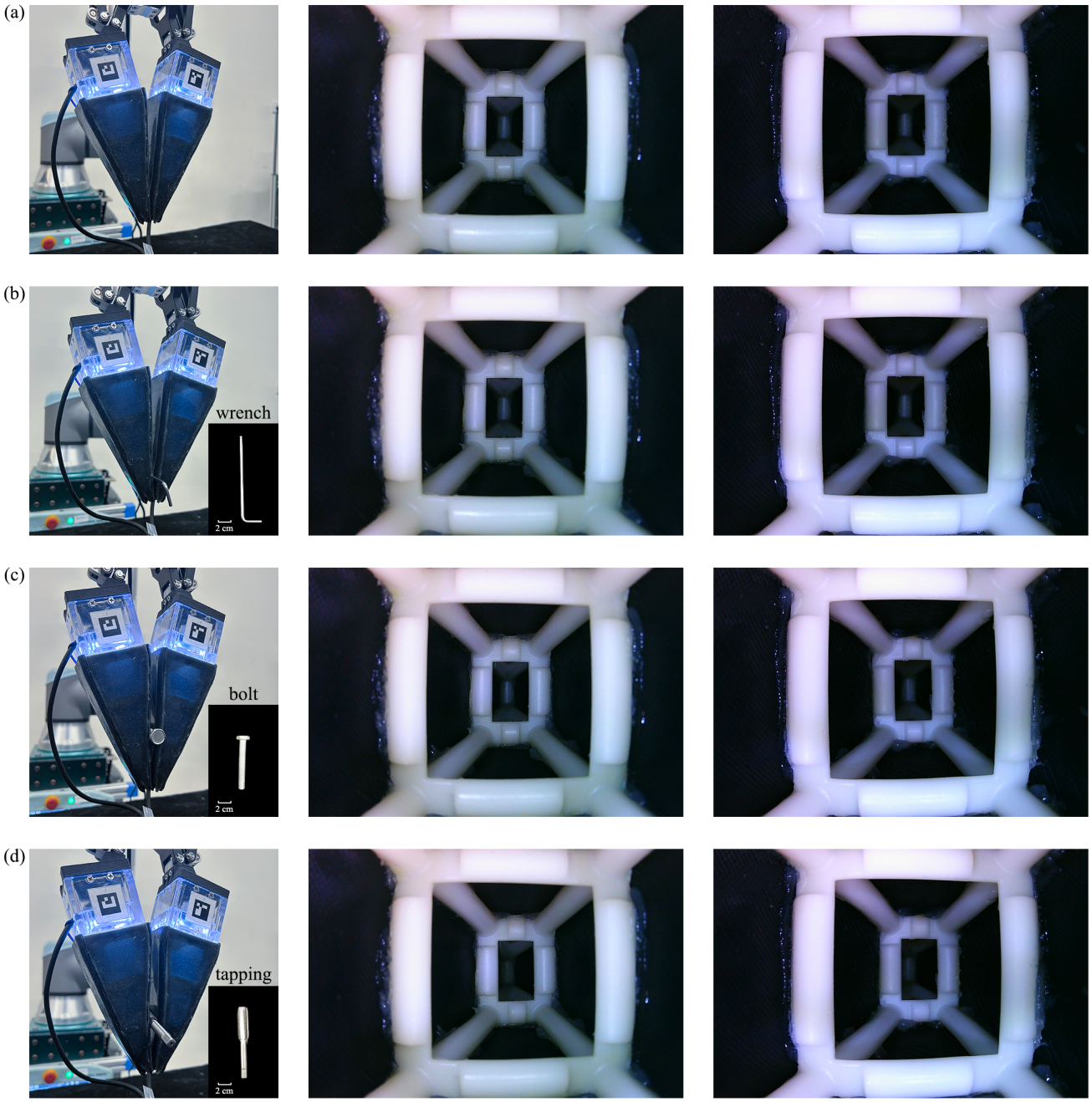



**Figure A2.** Examples of grasping objects: the left column is an outside image, the middle column is the left inner image, and the right column is the right inner image. (**a**,**b**) are comparisons before and after grasping a small wrench; (**c**) is a comparison grasping a bolt; (**d**) is a comparison grasping a tapping. The deformation of the skeleton is slight after grasping a wrench, and the features of screw threads in (**c**,**d**) are lost in the inner images.

## References

1. Klatzky, R.L.; Lederman, S.J.; Metzger, V.A. Identifying objects by touch: An "expert system". *Percept. Psychophys.* **1985**, *37*, 299–302. [CrossRef] [PubMed]
2. Dahiya, R.S.; Metta, G.; Valle, M.; Sandini, G. Tactile sensing—From humans to humanoids. *IEEE Trans. Robot.* **2009**, *26*, 1–20. [CrossRef]
3. Boivin, M.; Lin, K.Y.; Wehner, M.; Milutinović, D. Proprioceptive Touch of a Soft Actuator Containing an Embedded Intrinsically Soft Sensor using Kinesthetic Feedback. *J. Intell. Robot. Syst.* **2023**, *107*, 28. [CrossRef]

4. Zimmermann, C.; Ceylan, D.; Yang, J.; Russell, B.; Argus, M.; Brox, T. Freihand: A dataset for markerless capture of hand pose and shape from single rgb images. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 813–822. [CrossRef]
5. Wan, C.; Probst, T.; Gool, L.V.; Yao, A. Self-supervised 3d hand pose estimation through training by fitting. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 10853–10862.
6. Chen, X.; Liu, Y.; Dong, Y.; Zhang, X.; Ma, C.; Xiong, Y.; Zhang, Y.; Guo, X. Mobrecon: Mobile-friendly hand mesh reconstruction from monocular image. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–24 June 2022; pp. 20544–20554.
7. Doosti, B.; Naha, S.; Mirbagheri, M.; Crandall, D.J. Hope-net: A graph-based model for hand-object pose estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 6608–6617.
8. Tekin, B.; Bogo, F.; Pollefeys, M. H+ o: Unified egocentric recognition of 3d hand-object poses and interactions. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 4511–4520.
9. Hasson, Y.; Varol, G.; Tzionas, D.; Kalevatykh, I.; Black, M.J.; Laptev, I.; Schmid, C. Learning joint reconstruction of hands and manipulated objects. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 11807–11816.
10. Hampali, S.; Sarkar, S.D.; Rad, M.; Lepetit, V. Keypoint transformer: Solving joint identification in challenging hands and object interactions for accurate 3d pose estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–24 June 2022; pp. 11090–11100.
11. Mason, M.T. Toward robotic manipulation. *Annu. Rev. Control. Robot. Auton. Syst.* **2018**, *1*, 1–28. [CrossRef]
12. Wan, F.; Wang, H.; Liu, X.; Yang, L.; Song, C. DeepClaw: A Robotic Hardware Benchmarking Platform for Learning Object Manipulation. In Proceedings of the 2020 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), Boston, MA, USA, 6–10 July 2020; pp. 2011–2018.
13. Wang, G.; Manhardt, F.; Tombari, F.; Ji, X. Gdr-net: Geometry-guided direct regression network for monocular 6d object pose estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 16611–16621.
14. Lipson, L.; Teed, Z.; Goyal, A.; Deng, J. Coupled iterative refinement for 6d multi-object pose estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–24 June 2022; pp. 6728–6737.
15. Su, Y.; Saleh, M.; Fetzer, T.; Rambach, J.; Navab, N.; Busam, B.; Stricker, D.; Tombari, F. Zebrapose: Coarse to fine surface encoding for 6dof object pose estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 6738–6748. [CrossRef]
16. Von Drigalski, F.; Taniguchi, S.; Lee, R.; Matsubara, T.; Hamaya, M.; Tanaka, K.; Ijiri, Y. Contact-based in-hand pose estimation using bayesian state estimation and particle filtering. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; pp. 7294–7299.
17. Chalon, M.; Reinecke, J.; Pfanne, M. Online in-hand object localization. In Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, 3–7 November 2013; pp. 2977–2984.
18. Pfanne, M.; Chalon, M.; Stulp, F.; Albu-Schäffer, A. Fusing joint measurements and visual features for in-hand object pose estimation. *IEEE Robot. Autom. Lett.* **2018**, *3*, 3497–3504. [CrossRef]
19. Tu, Y.; Jiang, J.; Li, S.; Hendrich, N.; Li, M.; Zhang, J. PoseFusion: Robust Object-in-Hand Pose Estimation with SelectLSTM. *arXiv* **2023**, arXiv:2304.04523.
20. Wen, B.; Mitash, C.; Soorian, S.; Kimmel, A.; Sintov, A.; Bekris, K.E. Robust, occlusion-aware pose estimation for objects grasped by adaptive hands. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; 2020; pp. 6210–6217.
21. Álvarez, D.; Roa, M.A.; Moreno, L. Tactile-based in-hand object pose estimation. In Proceedings of the Iberian Robotics Conference, Sevilla, Spain, 22–24 November 2017; pp. 716–728.
22. Yang, L.; Han, X.; Guo, W.; Wan, F.; Pan, J.; Song, C. Learning-based optoelectronically innervated tactile finger for rigid-soft interactive grasping. *IEEE Robot. Autom. Lett.* **2021**, *6*, 3817–3824. [CrossRef]
23. Yuan, W.; Dong, S.; Adelson, E.H. Gelsight: High-resolution robot tactile sensors for estimating geometry and force. *Sensors* **2017**, *17*, 2762. [CrossRef] [PubMed]
24. Lambeta, M.; Chou, P.W.; Tian, S.; Yang, B.; Maloon, B.; Most, V.R.; Stroud, D.; Santos, R.; Byagowi, A.; Kammerer, G.; et al. Digit: A novel design for a low-cost compact high-resolution tactile sensor with application to in-hand manipulation. *IEEE Robot. Autom. Lett.* **2020**, *5*, 3838–3845. [CrossRef]
25. Yamaguchi, A.; Atkeson, C.G. Combining finger vision and optical tactile sensing: Reducing and handling errors while cutting vegetables. In Proceedings of the 2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids), Cancun, Mexico, 15–17 November 2016; pp. 1045–1051.
26. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [CrossRef]

27. Wan, F.; Liu, X.; Guo, N.; Han, X.; Tian, F.; Song, C. Visual Learning Towards Soft Robot Force Control using a 3D Metamaterial with Differential Stiffness. In Proceedings of the Conference on Robot Learning, Auckland, New Zealand, 14–18 December 2022; pp. 1269–1278.
28. Nair, V.; Hinton, G.E. Rectified linear units improve restricted boltzmann machines. In Proceedings of the 27th International Conference on Machine Learning (ICML-10), Haifa, Israel, 21–24 June 2010; pp. 807–814.
29. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the International Conference on Machine Learning. Pmlr, Lille, France, 6–11 July 2015; pp. 448–456.
30. Bradski, G. The OpenCV Library. *Dr. Dobb'S J. Softw. Tools* **2000**, *25*, 120–123.
31. Villalonga, M.B.; Rodriguez, A.; Lim, B.; Valls, E.; Sechopoulos, T. Tactile object pose estimation from the first touch with geometric contact rendering. In Proceedings of the Conference on Robot Learning, London, UK, 8–11 November 2021; pp. 1015–1029.
32. Sattler, T.; Maddern, W.; Toft, C.; Torii, A.; Hammarstrand, L.; Stenborg, E.; Safari, D.; Okutomi, M.; Pollefeys, M.; Sivic, J.; et al. Benchmarking 6dof outdoor visual localization in changing conditions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8601–8610. [CrossRef]
33. Gao, Y.; Matsuoka, S.; Wan, W.; Kiyokawa, T.; Koyama, K.; Harada, K. In-Hand Pose Estimation Using Hand-Mounted RGB Cameras and Visuotactile Sensors. *IEEE Access* **2023**, *11*, 17218–17232. [CrossRef]
34. Dikhale, S.; Patel, K.; Dhingra, D.; Naramura, I.; Hayashi, A.; Iba, S.; Jamali, N. Visuotactile 6d pose estimation of an in-hand object using vision and tactile sensor data. *IEEE Robot. Autom. Lett.* **2022**, *7*, 2148–2155. [CrossRef]
35. Yan, Y.; Hu, Z.; Yang, Z.; Yuan, W.; Song, C.; Pan, J.; Shen, Y. Soft magnetic skin for super-resolution tactile sensing with force self-decoupling. *Sci. Robot.* **2021**, *6*, eabc8801. [CrossRef] [PubMed]
36. Gou, M.; Pan, H.; Fang, H.S.; Liu, Z.; Lu, C.; Tan, P. Unseen object 6D pose estimation: a benchmark and baselines. *arXiv* **2022**, arXiv:2206.11808.