2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

CNN-based Visual Servoing for Simultaneous Positioning and Flattening of Soft Fabric Parts

Fuyuki Tokuda[1,2], Akira Seino[1,2], Akinari Kobayashi[1,2] and Kazuhiro Kosuge[1,3]

***Abstract*— This paper proposes CNN-based visual servoing for simultaneous positioning and flattening of a soft fabric part placed on a table by a dual manipulator system. We propose a network for multimodal data processing of grayscale images captured by a camera and force/torque applied to force sensors. The training dataset is collected by moving the real manipulators, which enables the network to map the captured images and force/torque to the manipulator's motion in Cartesian space. We apply structured lighting to emphasize the features of the surface of the fabric part since the surface shape of the non-textured fabric part is difficult to recognize by a single grayscale image. Through experiments, we show that the fabric part with unseen wrinkles can be positioned and flattened by the proposed visual servoing scheme.**

I. INTRODUCTION

The recent labor shortage caused by the aging of the world population has increased the demand for industrial robots, and the number of robots in operation in the manufacturing industry is increasing every year [1]. In manufacturing industries, object handling is essential for replacing humans with robots. However, industrial robots can only be used for handling rigid objects.

Garment production is a typical example that requires the handling of super flexible fabric parts. The shape of a flexible fabric part depends on the environment which supports the fabric part. Whenever a robot handles a fabric part of a garment, the fabric part must be rigid, and a fixture is used to rigidify the flexible fabric part. That is, the fixture is used to fix the pose and flatten the surface shape of the fabric part. The fixture is designed for each fabric part and each production process. The specifically designed fixture for each fabric part with the same shape and dimension can be used for mass production of the same garment with the same dimension.

In this paper, we consider the problem to position and flatten a fabric part without using a specifically designed fixture using a dual manipulator system. We propose CNN-based visual servoing scheme which generates the motion of manipulators based on the captured images and the force/torque applied to the end-effector.

Visual servoing is a method to control the motion of robots based on captured images as explained in [2]. Visual servoing is categorized into position-based visual servoing (PBVS) and image-based visual servoing (IBVS) [3]. PBVS is a technique to calculate the robot trajectory using the difference between the current and target end-effector pose estimated in real time from the captured images. If the difference between the end-effector's target and current pose can be accurately estimated, the end-effector's trajectory can be easily calculated. However, accurate camera intrinsic and extrinsic parameters are required, which can only be obtained by performing a prior camera calibration and camera-robot calibration.

IBVS is a method for calculating robot trajectories in real time based on image features extracted from captured images [2] [3]. The end-effector trajectory is computed using an image Jacobian matrix that maps image features to robot motion in Euclidean space, thus intrinsic and extrinsic camera parameters are not required. The major limitation of IBVS is that the image feature should be manually designed depending on the scenes. The image coordinate of straight lines and points in the scenes are usually chosen as image features, though such features can only be applied to limited scenes.

Instead of extracting geometrical features from images, recent visual servoing is designed based on the use of image luminance as image features [4] [5] [6], which can be categorized as direct visual servoing. Such visual servoing schemes do not require image feature extraction, and positioning accuracy tends to be higher if the initial pose of the end-effector is close to the target pose. However, such methods tend to have a small convergence domain due to the high nonlinearity between the robot's workspace and the feature space of the image.

Various visual servoing schemes have been proposed to expand the convergence domain of visual servoing, including methods based on mutual information [7], image histogram [8], and Q-learning [9]. Recently, convolutional neural network-based visual servoing (CNN-based visual servoing) has been proposed [10] for positioning a camera-mounted drone in static scenes. Since then, research on CNN-based visual servoing is becoming more and more active [11] [12] [13] [14].

In addition to a simple positioning task, CNN-based visual servoing has been applied to industrial tasks such as VGA adapter insertion tasks [12] and industrial product assembly

[1]Fuyuki Tokuda, Akira Seino, Akinari Kobayashi, Kazuhiro Kosuge are with Centre for Transformative Garment Production, Units 1215 to 1220, 12/F, Building 19W, SPX1, Hong Kong Science Park, Pak Shek Kok, N.T., Hong Kong SAR {fuyuki.tokuda, akira.seino, akinari.kobayashi, kazuhiro.kosuge}@transgp.hk

[2]Fuyuki Tokuda, Akira Seino, and Akinari Kobayashi are with Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong SAR

[3]Kazuhiro Kosuge is Director of the JC STEM Lab of Robotics for Soft Materials, Department of Electrical and Electronic Engineering, Faculty of Engineering, The University of Hong Kong, Hong Kong SAR kosuge@hku.hk

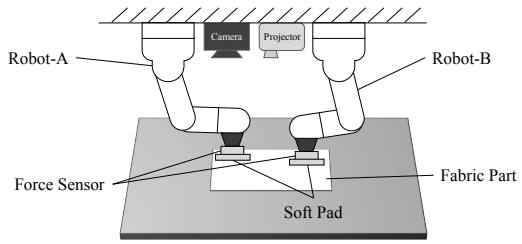


Fig. 1. The robot system for positioning a fabric part by a dual manipulator.

tasks [13]. These researches show that the CNN-based visual servoing has a larger convergence domain than conventional visual servoing and is more robust to changes in lighting conditions, occlusions, and scenes. Related works of CNN-based visual servoing are further presented in the next section.

In this paper, we present CNN-based visual servoing for simultaneous positioning and flattening of a fabric part by a dual manipulator system that consists of a camera and force sensors. We assume that a square fabric part is placed on a flat table and the end-effectors press down near the edge of the fabric part, as shown in Figure 1.

The contributions of this paper can be summarized as follows:

1) We propose a novel CNN-based visual servoing for simultaneous positioning and flattening of a soft fabric part with a dual manipulator equipped with soft pads as end-effectors.
2) The presented network architecture is intended for multimodal data processing, where the image and the force/torque applied to the end-effectors are processed in convolutional layers to calculate the difference between the current and target poses of the end-effectors.
3) We present a training dataset to enable the network to position a fabric part with various wrinkles. A pattern projection is introduced to emphasize the surface shape of the fabric parts, since the surface shape of a non-textured fabric part is difficult to recognize in a single camera image.
4) We demonstrate the positioning of a fabric part while keeping the surface flat by a dual manipulator system. We also show that the proposed method can position the fabric part from various initial pose errors and wrinkles.

Section II presents related works on visual servoing using CNN. Section III proposes a new CNN-based visual servoing for controlling the pose of a fabric part with a dual manipulator system. The experimental results of the proposed method are presented in Section IV. Section V concludes this paper.

II. RELATED WORKS

CNN has shown state-of-the-art performance in the fields of object identification [15] [16] [17], image segmentation [18] [19] [20], image generation [21] [22] [23], and so on. Recently, CNNs have been utilized for visual feedback control, which uses images captured by a camera to control the pose of a robot.

S. James et al. [24] propose a visual feedback control that uses CNN and LSTM to compute the trajectory of the manipulator to accomplish the tasks of finding a cube, reaching out, grabbing it, finding a basket, and dropping it inside. The key contribution of this method is that the network is trained with a dataset collected by simulation software and generalized to real-world images using domain randomization.

S. Levine et al. [25] propose a method that directly maps captured images to the torque of a robot's motor using a CNN trained by guided policy search methods. The experiments show that the robot can complete complex tasks such as inserting different blocks into an appropriate hole, screwing a bottle's lid, and hanging hangers on a bar.

In some research, CNN is used to predict the pose differences between the current and target end-effector/camera pose to servo the robot toward the target pose [10] [11] [12] [13] [14]. These methods are intended to decrease the difference between the target image and the current image by controlling the robot based on the difference between the target and the current end-effector pose as predicted by the CNN. These methods can be categorized as CNN-based visual servoing.

To the best of our knowledge, CNN-based visual servoing is first proposed by A. Saxena et al. [10] for pose control of a quadrotor. CNN is applied to the conventional visual servoing scheme to learn visual features for servoing the robot to the target pose in a static unstructured and unknown environment. External camera parameters, internal camera parameters, and scene geometries are not required for CNN-based visual servoing since the network is trained by camera images captured at different robot poses. Experiments of quadrotor positioning show that the quadrotor can be positioned from large initial displacements in both indoor and outdoor environments.

Q. Bateux et al. [11] propose a new method to efficiently and automatically create a training dataset from a single image for CNN-based visual servoing. They show that the data augmentation methods make the CNN-based visual servoing more robust to various disturbances such as occlusions and lighting variations. The positioning experiments using a 6 DoF manipulator with a hand-eye camera show that the manipulator's end-effector can be positioned to the target pose from a large initial displacement. The convergence domain of CNN-based visual servoing is shown to be larger compared to conventional visual servoing, though positioning accuracy remains an issue.

To increase the positioning accuracy of the CNN-based visual servoing, C. Yu et al. [12] propose a new network architecture. The proposed network is designed based on Siamese architecture [26] for highly accurate relative camera pose estimation. They demonstrate the VGA-connector insertion using a 6 DoF manipulator with a hand-eye camera and achieved sub-millimeter accuracy positioning.

F. Tokuda et al. [13] propose CNN-based visual servoing

with a simple image processing technique for assembling industrial objects grasped by a parallel gripper using an eye-to-hand camera. The method is proposed to achieve positioning of the grasped object when the relative pose of the grasped object and the gripper is different for each positioning attempt. The positioning experiments using a 6 DoF manipulator show that the CNN-based visual servoing can achieve precise positioning of a grasped non-textured industrial object.

Towards a higher positioning accuracy of CNN-based visual servoing, F. Tokuda et al. [14] propose a new network architecture for CNN-based visual servoing. The proposed network architecture is designed based on Siamese architecture with a regression network and a subtraction process. High positioning accuracy is achieved by estimating the pose difference of the end-effector from the difference between the encoded target image and the current image. The positioning experiments show that the network has higher positioning accuracy and a larger convergence domain compared to the previous network used for CNN-based visual servoing.

Overall, the performance of CNN-based visual servoing have been improving step by step and the application of CNN-based visual servoing to industrial tasks has been increasing. This paper proposes a new CNN-based visual servoing for positioning a fabric part with a dual manipulator system, which has never been done before in research on CNN-based visual servoing. The proposed method requires neither camera calibration nor calibration between two manipulators as long as the configuration of the two robots remains the same as the configuration when the training dataset is collected.

III. PROPOSED METHOD

A. VISUAL SERVOING CONTROL

Figure 2 shows a block diagram of the proposed CNN-based visual servoing for a fabric part positioning, where the two robots are named "Robot-A" and "Robot-B". The network is trained so as to predict the end-effector pose difference between the current and target pose of the two robots ($\boldsymbol{r}_A^* - \boldsymbol{r}_A$, $\boldsymbol{r}_B^* - \boldsymbol{r}_B$) from the region of interest (ROI) applied current image $\boldsymbol{I}_{ROI}$, ROI applied target image $\boldsymbol{I}_{ROI}^*$, current end-effector poses of the two robots ($\boldsymbol{r}_A \in \mathbb{R}^6$, $\boldsymbol{r}_B \in \mathbb{R}^6$), and force/torque applied to end-effectors of the two robots ($\boldsymbol{F}_A \in \mathbb{R}^6$, $\boldsymbol{F}_B \in \mathbb{R}^6$). ROI is applied to the images to mask the two end-effectors of the robots.

The output of the network can be expressed as

$$(\boldsymbol{r}_A^* - \boldsymbol{r}_A, \boldsymbol{r}_B^* - \boldsymbol{r}_B) = f(\boldsymbol{I}_{ROI}^*, \boldsymbol{I}_{ROI}, \boldsymbol{r}_A, \boldsymbol{r}_B, \boldsymbol{F}_A, \boldsymbol{F}_B). \quad (1)$$

The joint angular velocity of each robot can be calculated by the two following control law

$$\begin{aligned} \dot{\boldsymbol{\theta}}_A &= \lambda_A \boldsymbol{J}^{-1}_{\mathrm{robot-A}}(\boldsymbol{r}_A^* - \boldsymbol{r}_A), \\ \dot{\boldsymbol{\theta}}_B &= \lambda_B \boldsymbol{J}^{-1}_{\mathrm{robot-B}}(\boldsymbol{r}_B^* - \boldsymbol{r}_B), \end{aligned} \quad (2)$$

where $\dot{\boldsymbol{\theta}}_A \in \mathbb{R}^6$ and $\dot{\boldsymbol{\theta}}_B \in \mathbb{R}^6$ is the joint angular velocity that is commanded to each servo controller, λ_A and λ_B is the visual servoing gain for each robot, $\boldsymbol{J}^{-1}_{\mathrm{robot-A}}$ and $\boldsymbol{J}^{-1}_{\mathrm{robot-B}}$ is the inverse matrix of manipulator Jacobian for each robot.

Force sensors are attached to the manipulator's end-effectors to capture the force/torque applied to the end-effectors. The force/torque is expected to improve the positioning performance of the network since the tension applied to the fabric part provides a clue for estimation. The discussion about the effect of feeding force/torque on the network is presented in Section V.

The end-effector pose of the dual manipulator can be controlled to the target pose by controlling the joint velocities of each robot in real time according to Equation (2).

B. NETWORK ARCHITECTURE

We present Multimodal DEFINET for a dual manipulator (Figure 3), which is improved from DEFINET [14] that we previously presented for CNN-based visual servoing to position a rigid object using a single-arm manipulator. Multimodal DEFINET for a dual manipulator is a modified version of DEFINET for use with a dual manipulator, which can receive not only images but also forces applied to the sensors and joint angles of the manipulators.

The architecture of the backbone of the network is designed based on two VGG16 with the same parameters and weights [26]. Note that VGG16 is selected as the feature extractor in terms of estimation accuracy and prediction time. Each network accepts ROI applied current image $\boldsymbol{I}_{ROI}$ and target image $\boldsymbol{I}_{ROI}^*$ and outputs the image feature vector.

The two image features extracted from each VGG16 are provided to the subtraction layer in order to extract a feature representing the difference between the two input images. Then the global average pooling is applied to the subtracted image features to reduce the spatial dimension. The image feature from the global averaging pooling layer is concatenated with the extracted feature of the current end-effector poses of Robot-A and Robot-B ($\boldsymbol{r}_A$, $\boldsymbol{r}_B$) and force applied to the end-effectors ($\boldsymbol{F}_A$, $\boldsymbol{F}_B$). The features of the current pose of the end-effectors and the force/torque applied to the end-effectors are both extracted from fully connected layers with 64 nodes.

The concatenated image features are fed to two fully connected layers with 256 nodes and then fed to two different fully connected layers with 6 nodes. ReLU activation function is applied to FC layers with 256 nodes and 64 nodes, and the linear activation function is applied to FC layers with 6 node.

C. TRAINING DATASET

The task space of each end-effector of a dual manipulator is defined, and the training dataset is collected by randomly moving the end-effectors within that range. The training dataset is collected by the steps shown in Algorithm 1. The fabric part is first pushed down by the end-effectors at a randomly selected point inside each task space. Note that the fabric part is kept flat when pushing down. An image

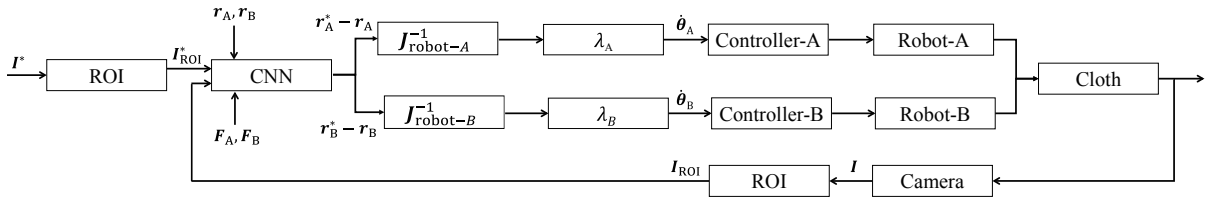


Fig. 2. The CNN-based visual servoing control system for positioning a soft fabric part.

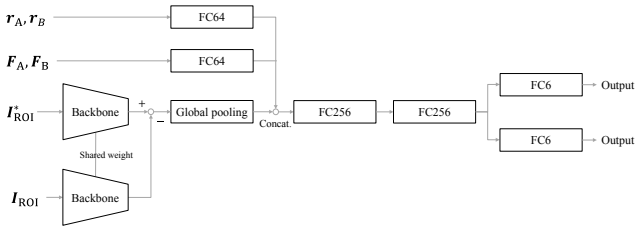


Fig. 3. The network architecture of Multimodal DEFINET for a dual manipulator.

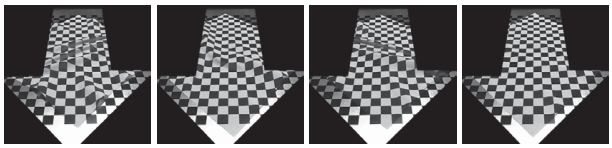


Fig. 4. The example images of the training dataset.

$\boldsymbol{I}_{i,j}$, pose of end-effectors $(\boldsymbol{r}^A_{i,j}, \boldsymbol{r}^B_{i,j})$, force/torque applied to the end-effectors $(\boldsymbol{F}^A_{i,j}, \boldsymbol{F}^B_{i,j})$ are captured and stored. Then, the two end-effectors are moved to randomly selected points inside each task space to make the fabric part wrinkled. After $j = 1$, one training data is formed by calculating the pose differences of the end-effector based on the captured data and the data captured at the previous loop. Note that, regarding form_data() function in Algorithm 1, $\boldsymbol{I}_{i,j}$ is regarded as the desired image and $\boldsymbol{I}_{i,j-1}$ is regarded as the current image. These processes are repeated for max_step_2$= 5$ times which creates 4 training data.

The end-effectors of manipulators are moved back to pose $(\boldsymbol{r}^A_{i,0}, \boldsymbol{r}^B_{i,0})$ which makes the fabric part flat again and lifted up to change the position to press down the fabric part. Again, the fabric part is pushed down at randomly selected points within each task space and training data are formed for max_step_2$= 5$ times. This entire process is repeated for max_step_1$= 400$ times which creates 1600 training data in total. The pressing down position is changed so as to include various relative poses between the end-effectors and the fabric part to the training dataset .

Finally, the ROI is designed manually based on the captured images and applied to the training dataset. The example of images included in the training dataset is shown in Figure 4. The two end-effectors of manipulators and background image pixels are masked by the ROI. The upper mask in the image covers the two end-effectors, while the lower mask covers the reflection of the projector light and the shadow of the robot. The image pixels of the two end-effectors must be covered since the network should be trained only to position the fabric part regardless of the pose of the end-effectors. The ROI is the same in all images.

When collecting the training dataset, a 12.5 mm square grid checker pattern is projected from a projector mounted on top of the manipulators. Since the fabric part we use is non-textured, the shape of the surface is difficult to recognize from images captured by a monocular camera. A checker pattern is projected to emphasize the geometry of the fabric part surface, making it easier to recognize the surface shape.

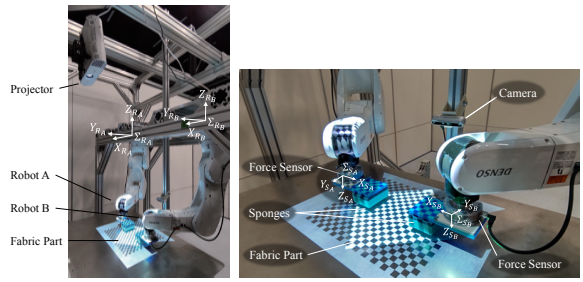
The training dataset is collected by randomly moving the fabric part to collect a dataset that maps various wrinkles of the fabric part to the pose differences in the end-effector space. In other words, the training dataset enables the network to learn how to manipulate the fabric part into the desired pose and shape as shown in the desired image. When performing inference, an image of a fabric part with a flat surface is fed to the network. Note that it is assumed that the flattening of the fabric part from the first given wrinkle is always controllable.

Algorithm 1 Training Dataset Collection

```
while i <max_step_1 do
  press_down_randomly()
  while j <max_step_2 do
    (I_{i,j}, r^A_{i,j}, r^B_{i,j}, F^A_{i,j}, F^B_{i,j}) ← capture_data()
    move_randomly()
    if 1 ≦ j then
      ([I_{i,j}, I_{i,j-1}, r^A_{i,j-1}, r^B_{i,j-1}, F^A_{i,j-1}, F^B_{i,j-1}],
      [r^A_{i,j} − r^A_{i,j-1}, r^B_{i,j} − r^B_{i,j-1}]) ← form_data()
    end if
  end while
  back_to_initial_pose(r^A_{i,0}, r^B_{i,0})
  lift_up()
end while
ROI_setting()
```

D. NETWORK TRAINING

As a loss function, Euclidean loss is calculated between the output and ground truth vector. The loss function is defined as



(a) (b)

Fig. 5. The fabric part positioning system with a dual-arm manipulator.

$$E = \alpha\|\Delta\tilde{\boldsymbol{t}}_A - \Delta\boldsymbol{t}_A\|_2 + \beta\|\Delta\tilde{\boldsymbol{\eta}}_A - \Delta\boldsymbol{\eta}_A\|_2 + \gamma\|\Delta\tilde{\boldsymbol{t}}_B - \Delta\boldsymbol{t}_B\|_2 + \epsilon\|\Delta\tilde{\boldsymbol{\eta}}_B - \Delta\boldsymbol{\eta}_B\|_2, \quad (3)$$

where $\Delta\tilde{\boldsymbol{t}}$, $\Delta\tilde{\boldsymbol{\eta}}$, $\Delta\boldsymbol{t}$, $\Delta\boldsymbol{\eta}$ are the ground-truth vector of the translational difference, the ground-truth vector of the rotational difference, the predicted vector of the translational difference, and the predicted vector of the rotational difference. α, β, γ, and ϵ are parameters to control the learning speed of the translation and rotation difference vectors of Robot A and Robot B. In this chapter, $\alpha = 1.0$, $\beta = 1.0$, $\gamma = 1.0$, and $\epsilon = 1.0$ are used. As an optimization method, Adadelta [27] is chosen and the network is trained for 200 epochs by API provided by Keras [28].

IV. EXPERIMENTS

This chapter presents the results of experiments in which the proposed method is used to place the fabric part in the target pose while keeping it flat. Figure 5 shows the dual manipulator system for the fabric part positioning experiment. The dual manipulator system consists of two 6-DOF manipulators (Denso VS-068), two force sensors (ATI Axia80) attached to the manipulators, and a camera (Intel RealSense D455) that captures grayscale images at 60 fps. Note that we use only the RGB camera of Intel RealSense D455. This is because the depth image of RealSense is not accurate enough for our task and has defects in pixel points when capturing a texture-less fabric part. The force sensors are connected to each robot controller (Denso RC8) by EtherCAT connection. Two soft polyester pads are attached to the manipulator's end-effector to press down on the fabric part. In the dual manipulator system, a projector (Epson EB-U50) is placed above the two manipulators to project the checker pattern. The base coordinate of each manipulator $\sum_{R_A}$, $\sum_{R_B}$ are attached at each of the base links of the manipulators and the force sensor coordinates $\sum_{S_A}$, $\sum_{S_B}$ are attached to each of the force sensors. The fabric part is 25 cm square, made of cotton and synthetic fibers, and wrinkles easily when force is applied.

The target image is captured when the fabric part surface is flat and in the target pose. Before each visual servoing experiment, the fabric part is randomly moved from the target pose and wrinkled manually. The task space is defined as [-5, 5] mm in translation in the X and Y axes and [-5, 5] deg. in translation around the Z axis from each end-effector's reference pose. The reference end-effector pose of Robot-A and Robot-B is r = (−320.20 mm, 379.77 mm, −899.64 mm, 180 deg., 0 deg., −180 deg.) and r = (276.70 mm, 364.06 mm, −888.09 mm, 180 deg., 0 deg., −180 deg.) with respect to each base coordinate. Note that translation in the Z axis direction and rotation in the X and Y axes are fixed during training data collection and positioning experiments. The visual servoing gain is set to λ=1.0 so as not to exceed the constraint of each joint angular velocity.

Figure 6 shows time-series images of visual servoing from 0 to 7 seconds. The left images are the images used for visual servoing and the right images are the images of the experiment. The result shows that the surface of the fabric part is flattened by the proposed method in about 3 to 4 seconds. After 3 to 4 seconds, there is no significant change in the pose and shape of the fabric part. Figure 7 (a) and (b) shows the trajectory of each end-effector poses along X,Y axes and rotation around Z axis during visual servoing, respectively. Soon after the start of visual servoing, the pose of each axis begins to change, and most of the axes converge to a certain pose during visual servoing. Figure 8 (a), (b), (c), (d), and (e) show the initial image, target image, initial error image, final error image, and the sum of squared difference (SSD) between the target image and current images during visual servoing. Note that the error image is the synthesized image between the current and target image. After the start of visual servoing, SSD decreases monotonically and converges in about 5 seconds.

Next, visual servoing is performed on a fabric part with various initial pose errors and wrinkles to check the performance of the proposed scheme to un-seen wrinkle. Figure 9 (a), (b), (c), (d) show the initial error images and final error images. In the experiments shown in (a) and (b), the vertical wrinkle and the diagonal wrinkle are applied to the fabric part, respectively, at the initial state. In the experiments shown in (c) and (d), random pose errors and wrinkles are applied to the fabric part at the initial state. The results show that the proposed method is capable of manipulating the fabric part to the target pose while keeping it flat from various initial pose errors and wrinkles.

To verify the effect of feeding force/torque on the network, we conducted positioning experiments using the network trained without feeding force/torque applied to the end-effectors. Figure 10 shows an example of initial and final error images when the fabric parts are positioned from random initial pose errors and wrinkles. The pose of each end-effectors tends to diverge during visual servoing (Figure 10 (a)) or stuck to local minima (Figure 10 (b)). In most cases, the network fails to position the fabric part to the target pose while keeping it flat. The results indicate that feeding force/torque to the network improves the generalization performance of the network for positioning a fabric part with unknown wrinkles.

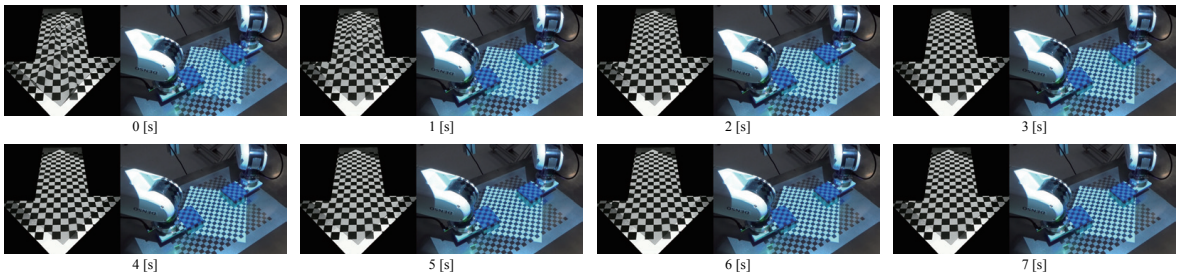


Fig. 6. The time-series images of the positioning experiment of a non-textured soft fabric part.

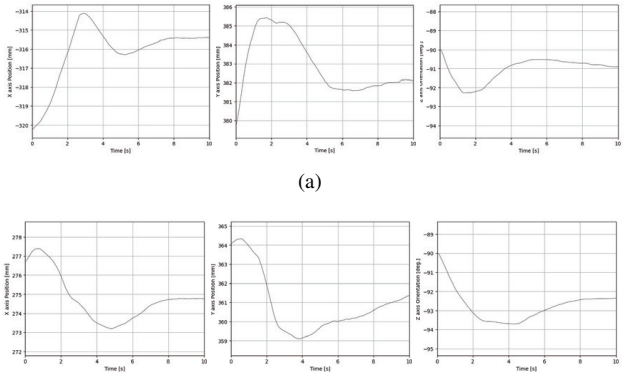


Fig. 7. The trajectory of end-effectors for each manipulator. (a) Translation along X axis, Y axis, and rotation around Z axis of Robot-A end-effector. (b) Translation along X axis, Y axis, and rotation around Z axis of Robot-B end-effector.

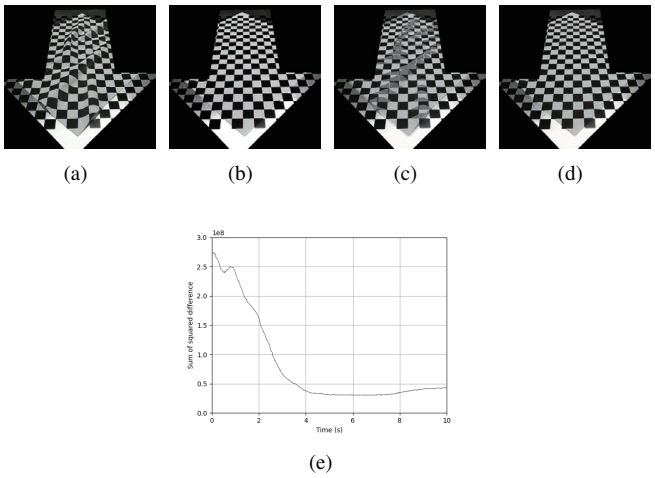


Fig. 8. The results of the positioning experiment. (a) The initial image. (b) The target image. (c) The initial error image. (d) The error image after visual servoing. (e) The sum of the squared difference between the target image and the current image.

V. CONCLUSION

In this paper, we presented a CNN-based visual servoing for positioning a non-textured soft fabric part in the target pose while keeping it flat using a dual manipulator system. We proposed Multimodal DEFINET for dual manipulator, and presented a method to collect a training dataset. Experiments have shown that the proposed method is capable of positioning a soft fabric part while keeping the surface flat.

Future work will focus on the increase of positioning accuracy and expanding the convergence domain by analyzing optimal projection patterns and applying new network architectures. A data augmentation should also be considered to enable the positioning of fabric parts with various shapes, thicknesses, colors, and textures by the same network, which is left for future work.

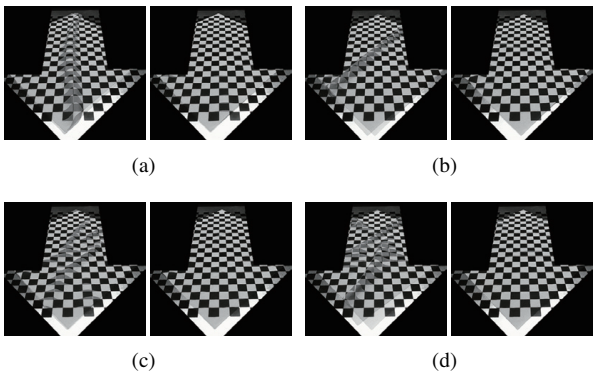


Fig. 9. The initial error and final error images when positioning the fabric part from various initial poses and wrinkles. (a) The error images when positioning a fabric part with vertical wrinkle, (b) with diagonal wrinkle, (c),(d) with randomly applied wrinkle.

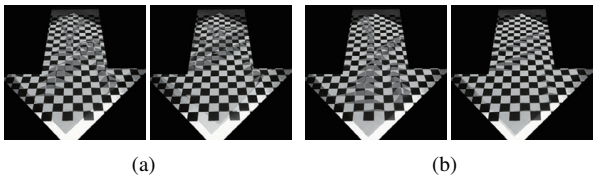


Fig. 10. The initial and final error images when a fabric part is positioned by the network trained without force/torque.

ACKNOWLEDGMENT

This work was supported in part by the Innovation and Technology Commission of the HKSAR Government under the InnoHK initiative. The research work described in this paper was in part conducted in the JC STEM Lab of Robotics for Soft Materials funded by The Hong Kong Jockey Club Charities Trust.

REFERENCES

[1] "IFR presents world robotics 2021 reports," (Date last accessed 15-September-2022). [Online]. Available: https://ifr.org/ifr-press-releases/news/robot-sales-rise-again

[2] F. Chaumette and S. Hutchinson, "Visual servo control. i. basic approaches," *IEEE Robot. Automat. Mag.*, vol. 13, no. 4, pp. 82–90, 2006.

[3] S. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control," *IEEE Trans. Robot. Automat.*, vol. 12, no. 5, pp. 651–670, 1996.

[4] C. Collewet and E. Marchand, "Photometric visual servoing," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 828–834, 2011.

[5] K. Deguchi, "A direct interpretation of dynamic images with camera and object motions for vision guided robot control," *Int. J. Comput. Vis.*, vol. 37, no. 1, pp. 7–20, 2000.

[6] E. Marchand and C. Collewet, "Using image gradient as a visual feature for visual servoing," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 5687–5692.

[7] A. Dame and E. Marchand, "Mutual information-based visual servoing," *IEEE Transactions on Robotics*, vol. 27, no. 5, pp. 958–969, 2011.

[8] Q. Bateux and E. Marchand, "Histograms-based visual servoing," *IEEE Robotics and Automation Letters*, vol. 2, no. 1, pp. 80–87, 2017.

[9] H. Shi, X. Li, K.-S. Hwang, W. Pan, and G. Xu, "Decoupled visual servoing with fuzzy ¡italic¿q¡/italic¿-learning," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 1, pp. 241–252, 2018.

[10] A. Saxena, H. Pandya, G. Kumar, A. Gaud, and K. M. Krishna, "Exploring convolutional networks for end-to-end visual servoing," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2017, pp. 3817–3823.

[11] Q. Bateux, E. Marchand, J. Leitner, F. Chaumette, and P. Corke, "Training deep neural networks for visual servoing," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2018, pp. 3307–3314.

[12] C. Yu, Z. Cai, H. Pham, and Q.-C. Pham, "Siamese convolutional neural network for sub-millimeter-accurate camera pose estimation and visual servoing," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 935–941.

[13] F. Tokuda, S. Arai, and K. Kosuge, "Object positioning by visual servoing based on deep learning," *Transactions of the Society of Instrument and Control Engineers*, vol. 55, pp. 717–725, 01 2019.

[14] ——, "Convolutional neural network-based visual servoing for eye-to-hand manipulator," *IEEE Access*, vol. 9, pp. 91 820–91 835, 2021.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[16] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 580–587.

[17] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[18] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[19] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[20] F. Visin, A. Romero, K. Cho, M. Matteucci, M. Ciccone, K. Kastner, Y. Bengio, and A. Courville, "Reseg: A recurrent neural network-based model for semantic segmentation," in *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2016, pp. 426–433.

[21] Y. Yu, Z. Gong, P. Zhong, and J. Shan, "Unsupervised representation learning with deep convolutional neural network for remote sensing images," in *Image and Graphics*. Springer International Publishing, 2017, pp. 97–108.

[22] M. Chen, A. Radford, R. Child, J. Wu, H. Jun, D. Luan, and I. Sutskever, "Generative pretraining from pixels," in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. PMLR, 13–18 Jul 2020, pp. 1691–1703. [Online]. Available: https://proceedings.mlr.press/v119/chen20s.html

[23] T. Xu, P. Zhang, Q. Huang, H. Zhang, Z. Gan, X. Huang, and X. He, "Attngan: Fine-grained text to image generation with attentional generative adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[24] S. James, A. J. Davison, and E. Johns, "Transferring end-to-end visuomotor control from simulation to real world for a multi-stage task," in *Proceedings of the 1st Annual Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, S. Levine, V. Vanhoucke, and K. Goldberg, Eds., vol. 78. PMLR, 13–15 Nov 2017, pp. 334–343. [Online]. Available: https://proceedings.mlr.press/v78/james17a.html

[25] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 1334–1373, 2016.

[26] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a "siamese" time delay neural network," in *Proceedings of the 6th International Conference on Neural Information Processing Systems*, ser. NIPS'93. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993, p. 737–744.

[27] M. D. Zeiler, "Adadelta: An adaptive learning rate method," *arXiv preprint arXiv:1212.5701*, 2012.

[28] F. Chollet *et al.*, "Keras," https://keras.io, 2015.