Touch Detection in Augmented Omni-Surface for Human-Robot Teaming

Fujian Yan[1]
Edgar Chavez[1]
Yimesker Yihun[2]
Hongsheng He[1*]
[1]*School of Computing, Wichita State University, USA*
[2]*Department of Mechanical Engineering, Wichita State University, USA*
Hongsheng.he@wichita.edu

Abstract

This paper proposes an architecture that augments arbitrary surfaces into an interactive touching interface. The proposed architecture can detect the number of touching fingertips of human operators by detecting and recognizing the fingertips with a convolutional neural network (CNN). The inputs of the CNN model are images that are ac[1]quired by an RGB-D sensor. The aligned depth information acquired by the RGB-D sensor generates the plane model, which determines whether fingertips are touching the surface or not. Instead of the traditional plane modeling method that can only be used for flat surfaces, the proposed system can also work on curved surfaces. Corresponding gestures are defined based on the detected touching fingers on the surface. The feedback of the robots is projected on the working surface accordingly with an interactive projector. Compared to conventional programming interfaces, directly touching is much more natural for human beings. Based on the experiments, the proposed system could reduce the massive training time of operators.

Keywords: Touching tracking, human-robot teaming, sensor fusion

1. Introduction

When robots operate in proximity to human workers, the relevance of human-machine, or more particularly, the human-robot interface grows. Compared to conventional industrial robots, which are used to working in a solo-working manner or static environments, advanced industrial robots are expected to collaborate with human co-workers. The study (Lasota & Shah, 2015) has indicated that the manufacturing process becomes quicker, more efficient, and less costly with human-robot collaboration. For most industrial robotic applications, a teaching pendant is provided to human co-workers for interacting with robots. Gestures are a common method of communication among human workers. They can also be used to interact with robots (Sheikholeslami, Moon, & Croft, 2017; Yan, Wang, & He, 2021).

For gestured-based human-robot collaboration interfaces, gesture ambiguity increases as the tasks get complicated. In contrast, touch interaction is unambiguous, tactile, familiar to users, and comfortable for extended periods (Li, Tan, & He, 2020; Li, Zhang, Li, & He, 2021; Xiao, Schwarz, Throm, Wilson, & Benko, 2018). New interaction models have emerged due to the advancements in display and vision technology, allowing for informative and real-time communication in shared workspaces. With the development of commercial RGB-D sensors, the prospect of converting these ordinary surfaces into

This work is supported by NSF CMMI 2129113.

Submitted June 30, 2022
Revised August 31, 2022

large, touchable, and interactive devices is increasing (Ntelidakis, Zabulis, Grammenos, & Koutlemanis, 2015; Yan, Tran, & He, 2020; Yan, Wang, & He, 2020). Therefore, a human-robot teaming method based on touching can benefit human-robot collaboration. For this purpose, this paper proposes an augmented Omni surface that can directly interact with robots.

To enable surfaces with interactive touching feedback, previous work has focused on refurbishing existing surfaces by adding other sensors such as acoustic sensors (Harrison & Hudson, 2008). Environments limit these reconstructed or refurbished surfaces. Two difficulties prevent human-robot teaming with a touching method without refurbishing the current environments, such as adding sensors to a current construction. The first problem is touch detection, and the second one is surfacing fitting. Touch detection includes two parts, fingertips detection, and contact detection. A conventional image processing method was proposed by (Bhuyan, Neog, & Kar, 2012) to detect and recognize fingertips. However, this method is limited by the complexity of the background (Xiao, Hudson, & Harrison, 2016). A deep learning approach is proposed by (Koller, Ney, & Bowden, 2016), where the number of visible fingertips in an image is random in particular cases. To detect each fingertip, different neural networks are trained (Alam, Islam, & Rahman, 2022). A touch detection method equipped with depth cameras is proposed by (Xiao et al., 2016), but this method is only suitable for flat surfaces. In particular, the working place for human-robot teaming has not only flat surfaces but also curved surfaces such as the carbine of a plane, the frame of a car, or the blades of a turbine.

In this paper, we propose a method that enhances human-robot teaming by directly interacting with robots on the surface of the working space. To adapt to different working environments, the touch detection method can enable humans to interact with robots by touch in different human-robot teaming environments, such as flat, curvy, convex, and concave surfaces with a single commercial RGB-D camera. Fingertips are detected with a convolutional neural network (CNN), and touch is detected based on the RGB-D sensor.

2. Augmented Omni-Surface

The developed Omni-surface architecture for human-robot teaming is shown in Figure 1. To enable augmented Omni-surface for human-robot teaming, touch detection is essential. The proposed augmented Omni surface superimposes mutual understanding between humans and robots on the surface of the working space.

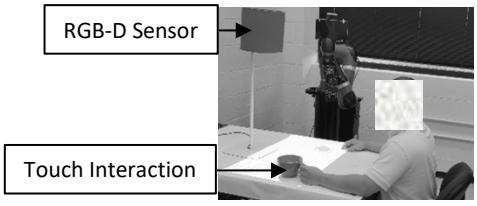

Figure 1. Augmented Omni-Surface.

The workflow of touch detection is illustrated in Figure 2. The proposed method contains two parts: surface fitting and touch detection. The RGB-D sensor captures an image having the user's hand, which will be the input for the fingertip's detection and surface modeling modules.

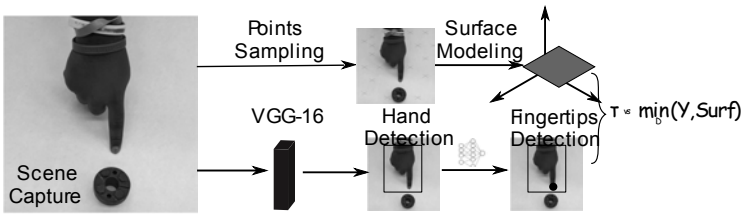

Figure 2. The framework of touch detection.

2.1. Arbitrary Surface Fitting

To fit the surface based on the collected depth scans, we assume the targeted surface is visible to the RGB-D camera. Once the camera has a clear sight of the plane, we collect random pixels $I(u, w, d)$ from the targeted surface. To fit the surface in a 3D space, we need to transform the 2D pixel coordinates into 3D world coordinates by

$$S(x, y, z)^T = K[R|t]^{-1} I(u, w, d)^T \quad (1)$$

where K is the intrinsic matrix of the RGB-D sensor and $[R|t]^{-1}$ is the inverse transformation matrix. With these random pixel coordinates, we can perform deprojection to calculate the depth of our random pixel coordinates. Specifically, our proposed method used Global Approximation (Piegl, 1996). In global surface approximation, surface error E and data are used as inputs to fit the approximation function. In general, the number of control points is not known to achieve the desired accuracy for the surface approximation. Thus, the approximation method is performed iteratively. In our proposed method, we used global surface approximation over the interpolation method due to the limited number of control points being automatically determined. Since surfaces are defined on a 2D plane, they can be described by

$$S'(\alpha, \beta) = \sum_{i=1}^{n} \sum_{j=1}^{m} N_{i,p}(\alpha) M_{j,q}(\beta) C_{i,j} \quad (2)$$

where α and β are parameters for the basis function $N_{i,p}$ and $N_{j,q}$ respectively. To determine a point using the α and β parameters, the basis functions for each of the parameters, $N_{i,p}(\alpha), and\ M_{j,q}(\beta)$, will be evaluated and multiplied with all the control points, $C_{i,j}$. In this sense, $N_{i,p}$ and $M_{j,q}$ are the knot vectors containing the size and degrees for our parameters.

The least-squares algorithm uses interpolation on the corner control points of the targeted surface to approximate the remaining control points of the approximated surface. All the parameters are used to create an approximation of the targeted surface. With the approximated surface, we begin to extract any curves that may exist in the approximated surface. This will return any curves that lie on the control points' surface.

2.2. Surface Touch Detection

Without installing additional hardware such as touch screens (Kim, Son, Lee, Kim, & Lee, 2013), it is challenging for the system to get feedback from human users' input. From an economic perspective and user-friendly, the proposed system is designed to detect touch by an RGB-D sensor. Typically, human beings operate touchable devices with fingertips, and fingertips have also played an essential role in human-robot interaction (Mitra & Acharya, 2007).

To detect fingertips from input images, two problems need to be solved. First, the hands need to be recognized, and second, we need to be able to retrieve the coordinates of each fingertip of the hand.

Earlier methods (Lai et al., 2016) isolate the hand regions from the acquired images using manually selected features such as color, depth, and contours. Some methods are not general, and the performance of these methods is easily affected by other natural factors such as the angles of taking images and ambiance lights (Nguyen, Kim, Kim, & Na, 2017). To obtain the coordinate of fingertips, the previous method (Liao, Zhou, Zhou, & Liang, 2012) employed a convex model that can model the endpoints of the extracted hand region. However, this method does not always find optimal results and is computationally costly (Jang, Noh, Chang, Kim, & Woo, 2015).

In this work, we formulate the problem of recognizing the hand and detecting the fingertips as a unified problem that combines classification and regression. We used a deep learning model to achieve this. To detect fingertips, we first detect the hand with 16 layers of visual geometry group (VGG) structures (Simonyan & Zisserman, 2014). Then, we recognize fingertips with a deep learning model that includes convolution layers, flatten layers, and coordinate layers. The structure of the fingertip detection model is shown in Figure 3. The model contains hand detection and fingertip recognition. Hands are detected by VGG-16 layers, and the output of VGG-16 is $\hat{Y}_h$. To extract features from $\hat{Y}_h$, a convolutional layer with the dimension of $(4 \times 4 \times 512)$ is employed. After the three convolutional layers, the output is $\hat{Y}_t$, which is a 2D array. Furthermore, it represents the positional information of the fingertips in the input image. To recognize fingertips, $\hat{Y}_t$ is flattened and fed into two fully connected (FC) layers.

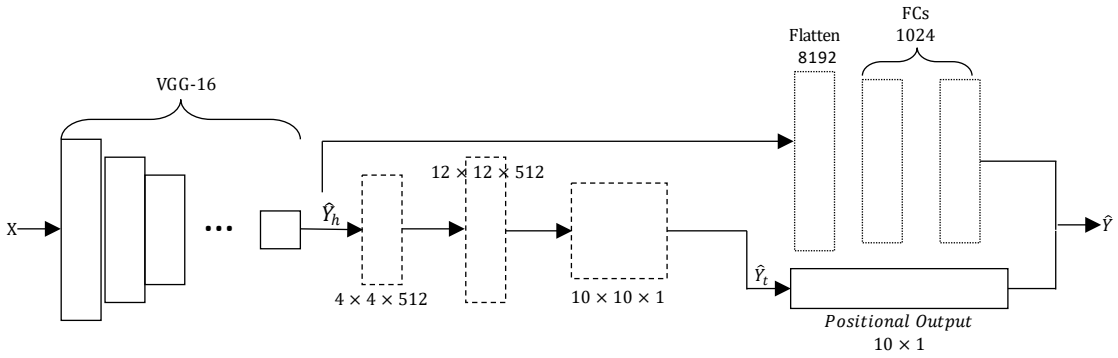

Figure 3. The structure of the model.

The input of the fingertip detection model is an image containing a single hand, and the model's output is the recognized fingers and their respective coordinates. The observed fingertips in the image are labeled as "1", and the unobserved fingertips are labeled as "0". The input of the model is $\mathrm{X}(u_i,\ w_i, f)$, where f is the label of each fingertip, and $f \in \{f_t, f_i, f_m, f_r, f_p\}$. The output of the model is $Y(u_i, w_i, P)$, where P is the probability of each detected fingertip's category, and $P \in \{P_t, P_i, P_m, P_r, P_p\}$. To minimize loss, the cross-entropy (CE) function and mean squared error (MSE) are employed with the loss

$$m = min\left[\boldsymbol{CE}\left(\widehat{\boldsymbol{P}}, \boldsymbol{P}\right) + \boldsymbol{MSE}\left(\widehat{\boldsymbol{Y}}(u_i, w_i)\ \boldsymbol{wi}, \boldsymbol{Y}(u_i, w_i)\right)\right] \quad (3)$$

where $\hat{P}$ and $\hat{Y}(\cdot)$ are the predicted results from the model, and P and $\mathrm{Y}(\cdot)$ is the coordinate of fingertips of labeled data used as ground truth.

To detect whether the user touched the surface, we computed the Euclidean distance between the detected fingertip $\hat{Y}(u_i,\ w_i)$ and the approximated surface $S_i(x, y, z),\ i \in \{0,1,2,3 \dots n\}$. We convert the detected fingertips in the image coordinates to the world frame coordinate $\hat{Y}(x_i, y_i, z_i)$, and then we calculate the distances between the detected fingertips and the approximated surface points, $d_m,$ where $m = 1,2,3, \dots, m$. We found the minimum distance d^* and compared it with z_i. If $d^* \leq \tau +$

z_i, then the detected fingertips are touching the surface, and τ is the thickness of a human's fingers.

3. Experiment

To evaluate the performance of the method to augment Omni surfaces, we executed the proposed system in different scenarios to investigate the generality of the method. To investigate the performance, we have computed the accuracy of our touch detection method. We have also evaluated the user's satisfaction level based on the designed questionnaire.

3.1. Performance of Finger Touching

We tested the augmented Omni-surfaces method on different scenarios, and some of the results are shown in Figure 4. In this experiment, the augmented Omni-surfaces method has been applied to different scenarios, including flat wooden surfaces, laminated flat surfaces, cloth concave surfaces, convex Styrofoam surfaces, cardboard flat surfaces, curved plastic surfaces, and bumpy brick surfaces. These scenarios cover different shapes and materials of the surface. Our results conclude that the convex Styrofoam surface and the small cardboard had a false detection, while the rest of the other surfaces showed correct detection results. Therefore, the augmented Omni-surfaces method is reliable in practical usage. There are two fault-detected results that are grouped by the dashed rectangles in Figure 4. The reason for these fault results is that the working surface is small, and the depth collections include more noise than desired points. Therefore, the modeled surfaces are not correct. One solution would be to decrease the area of the collected points and increase the number of points collected.

To evaluate the performance of the proposed method, we have run simulations on different types of surfaces. For each type of surface, we simulated 10,000 points, including touching or not touching. We computed precision, recall, and F-1 scores. The results are shown in performance Table1. Compared with other types of surfaces, the curved surface has the lowest result in precision, recall, and F-1 scores. The reason for this is that the curved surface used in this experiment takes an extreme case to test the performance of the proposed model.

Table 1. Touch detection performance

Surface Type	Precision	Recall	F-1
Concave Surface	84.46%	100%	91.66%
Curved Surface	62.2%	97.24%	75.87%
Flat Surface	68.68%	98.16%	80.81%
Overall	71.78%	98.47%	82.78%

3.2. User Acceptance

To evaluate the acceptance of augmented Omni-surface interaction, we designed a questionnaire consisting of four questions with four evaluation criteria. These criteria are correctness, ambiguity, effectiveness, and complexity. We also assigned the weight for each criterion to compute the final score for each perspective. 1) The correctness is defined as whether the user can instruct the robot with the proposed interaction method; 2) The ambiguity is defined as whether the user can precisely give the robot's instructions; 3) Compared with other interaction methods such as programming, or physical interaction with buttons, the designed interaction is effective; 4) Compared with other interaction

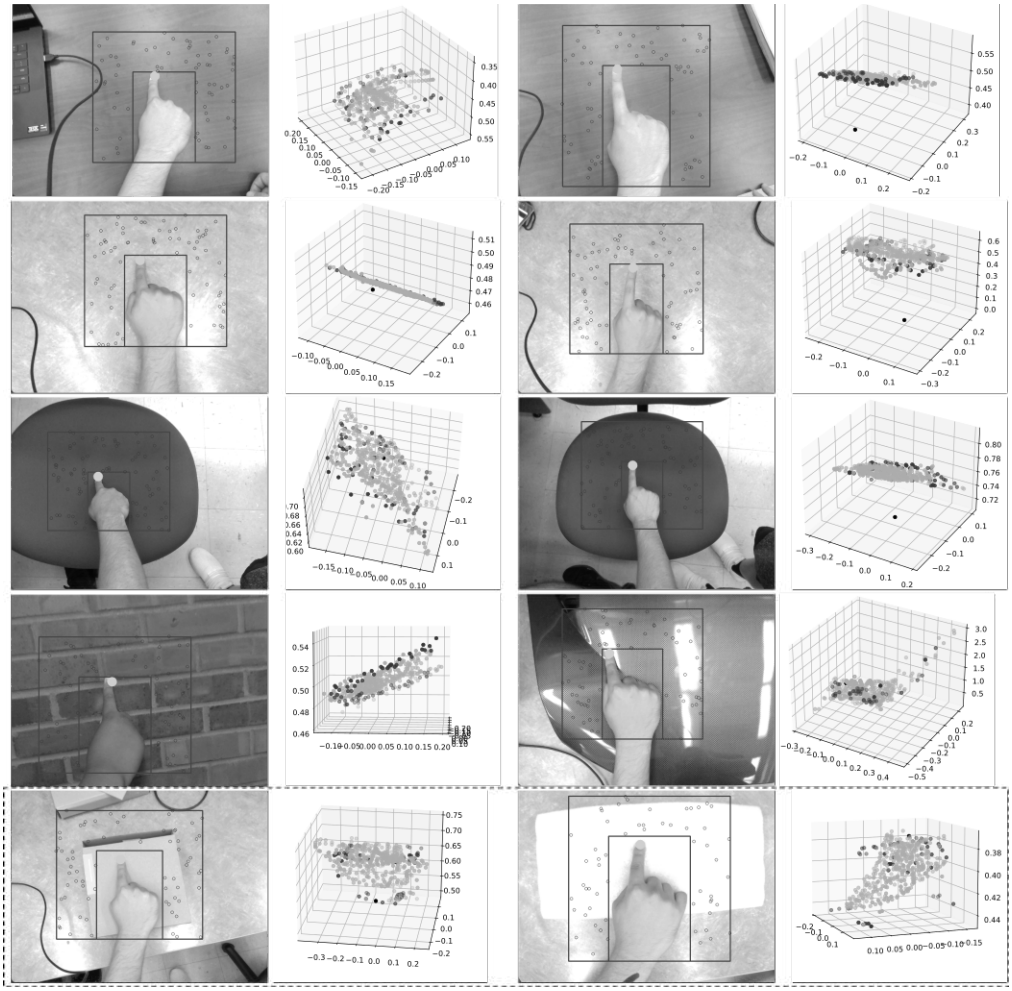

Figure 4. Touch detection results on different surfaces. The left image shows the detected results, and the right image shows the modeled surfaces. The bottom row demonstrated unsuccessful detections.

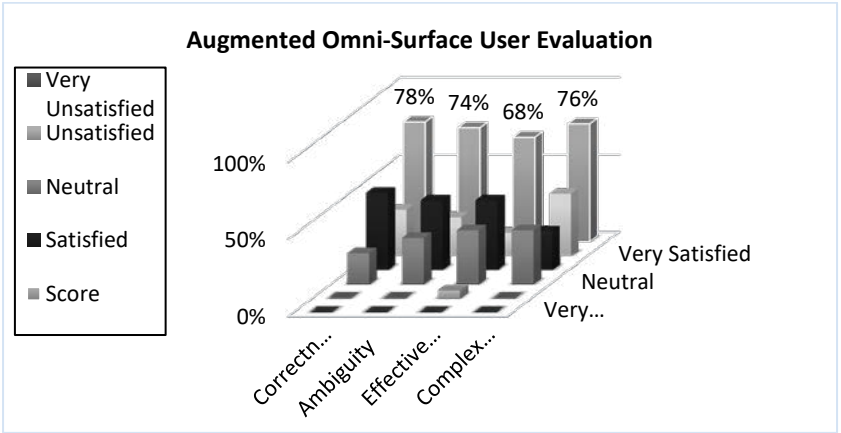

Figure 5. User satisfaction evaluation. There are five satisfaction levels for four evaluation criteria.

methods such as programming, or physical interaction with buttons, the designed interaction is complicated. Twenty participants took this questionnaire, and the results are shown in Figure 5. Based on the feedback from the questionnaires, users are satisfied with the proposed interaction method.

4. Conclusion

A method of touch detection for human-robot teaming has been proposed based on RGB-D sensing to detect a touch on arbitrary surfaces to enhance human-robot teaming. From the results of simulations, the proposed method achieved F-1 scores at 82.87%, which indicated the proposed method is promising. The proposed interface demonstrated improved user acceptance of traditional interaction methods according to the results of user satisfaction questionnaires.

5. Reference

Alam, M. M., Islam, M. T., & Rahman, S. M. M. (2022). Unified learning approach for egocentric hand gesture recognition and fingertip detection. In Pattern Recognition. 121, 108200. Elsevier BV. https://doi.org/10.1016/j.patcog.2021.108200

Bhuyan, M., Neog, D. R., & Kar, M. K. (2012). Fingertip detection for hand pose recognition. *International Journal on Computer Science and Engineering, 4*(3), 501.

Harrison, C., & Hudson, S. E. (2008). *Scratch Input: Creating Large, Inexpensive, Unpowered and Mobile Finger Input Surfaces.* Paper presented at the Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology, New York, NY, USA. ACM Press. https://doi.org/10.1145/1449715.1449747

Jang, Y., Noh, S.-T., Chang, H. J., Kim, T.-K., & Woo, W. (2015). 3d finger cape: Clicking action and position estimation under self-occlusions in egocentric viewpoint. *IEEE Transactions on Visualization and Computer Graphics, 21*(4), 501-510. https://doi.org/10.1109/tvcg.2015.2391860

Kim, S., Son, J., Lee, G., Kim, H., & Lee, W. (2013). *TapBoard: making a touch screen keyboard more touchable.* Paper presented at the Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. 553-562. http://doi.org/10.1145/2470654.2470733

Koller, O., Ney, H., & Bowden, R. (2016). *Deep hand: How to train a cnn on 1 million hand images when your data is continuous and weakly labelled.* Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition. Las Vegas, NV, USA.

Lai, Z., Yao, Z., Wang, C., Liang, H., Chen, H., & Xia, W. (2016). *Fingertips detection and hand gesture recognition based on discrete curve evolution with a kinect sensor.* Paper presented at the 2016 Visual Communications and Image Processing (VCIP). Chengdu, China. 10.1109/VCIP.2016.7805464

Lasota, P. A., & Shah, J. A. (2015). Analyzing the effects of human-aware motion planning on close-proximity human-robot collaboration. *Human factors, 57*(1), 21-33. https://doi.org/10.1177/0018720814565188

Li, H., Tan, J., & He, H. (2020). *Magichand: Context-aware dexterous grasping using an anthropomorphic robotic hand.* Paper presented at the 2020 IEEE International Conference on Robotics and Automation (ICRA). Paris, France. 10.1109/ICRA40945.2020.9196538

Li, H., Zhang, Y., Li, Y., & He, H. (2021). *Learning Task-Oriented Dexterous Grasping from Human Knowledge.* Paper presented at the 2021 IEEE International Conference on Robotics and Automation (ICRA).Xi'an, China. DOI: 10.1109/ICRA48506.2021.9562073

Liao, Y., Zhou, Y., Zhou, H., & Liang, Z. (2012, August 27-29). *Fingertips detection algorithm based on skin colour filtering and distance transformation.* Paper presented at the 2012 12th International Conference on Quality Software. Xi'an, China. https://doi.org/10.1109/QSIC.2012.62

Mitra, S., & Acharya, T. (2007). Gesture recognition: A survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 37*(3), 311-324.

Nguyen, H. D., Kim, Y. C., Kim, S. H., & Na, I. S. (2017- July 29-31). *A method for fingertips detection using RGB-D image and convolution neural network.* Paper presented at the 2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD). Guilin, China. https://doi.org/10.1109/FSKD.2017.8393373

Ntelidakis, A., Zabulis, X., Grammenos, D., & Koutlemanis, P. (2015). *Lateral touch detection and localization for interactive, augmented planar surfaces.* Paper presented at the International symposium on visual computing. https://doi.org/10.1007/978-3-319-27857-5_50

Piegl, L. a. T., Wayne. (1996). *The NURBS book*. (1st ed.). Monographs in Visual Communication. Springer Berlin, Heidelberg

Sheikholeslami, S., Moon, A., & Croft, E. A. (2017). Cooperative gestures for industry: Exploring the efficacy of robot hand configurations in expression of instructional gestures for human-robot interaction. *The International Journal of Robotics Research, 36*(5-7), 699-720. https://doi.org/10.1177/0278364917709941

Simonyan, K., & Zisserman, A. J. a. p. a. (2014). Very deep convolutional networks for large-scale image recognition. https://doi.org/10.48550/arXiv.1409.1556

Xiao, R., Hudson, S., & Harrison, C. (2016). *Direct: Making touch tracking on ordinary surfaces practical with hybrid depth-infrared sensing.* Paper presented at the Proceedings of the 2016 ACM International Conference on Interactive Surfaces and Spaces. https://doi.org/10.1145/2992154.2992173

Xiao, R., Schwarz, J., Throm, N., Wilson, A. D., & Benko, H. (2018). MRTouch: Adding Touch Input to Head-Mounted Mixed Reality. *IEEE Transactions on Visualization and Computer Graphics, 24*(4), 1653-1660. 10.1109/TVCG.2018.2794222

Yan, F., Tran, D. M., & He, H. (2020). Robotic understanding of object semantics by referring to a dictionary. *International Journal of Social Robotics, 12*(6), 1251-1263. https://doi.org/10.1007/s12369-020-00657-6

Yan, F., Wang, D., & He, H. (2020). *Robotic understanding of spatial relationships using neural-logic learning.* Paper presented at the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Las Vegas, NV, USA. 10.1109/IROS45743.2020.9340917

Yan, F., Wang, D., & He, H. (2021). *Comprehension of Spatial Constraints by Neural Logic Learning from a Single RGB-D Scan.* Paper presented at the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Prague, Czech Republic. 10.1109/IROS51168.2021.9635939