



3D bare-hand interactions enabling ubiquitous interactions with smart objects

S. K. Ong¹ · X. Wang¹ · A. Y. C. Nee¹

Received: 22 March 2019 / Revised: 14 November 2019 / Accepted: 11 February 2020 / Published online: 17 March 2020
© The Author(s) 2020

Abstract Ubiquitous augmented reality (UAR) implementation can benefit smart shop floor operations significantly. UAR from a user's first-person view can support and provide the user with suitable and comprehensive information without him/her being distracted from ongoing tasks. A natural hand-based interaction interface, namely, a mobile bare-hand interface (MBHI), is proposed to assist a user in exploring and navigating a large amount of information for a task in the user's first-person view. The integration of a smart shop floor and UAR-based MBHI is particularly challenging. A real shop floor environment is composed of challenging conditions for the implementation of UAR, e.g., messy backgrounds and significant changes in illumination conditions. Meanwhile, the MBHI is required to provide precise and quick responses to minimize the difficulty of a user's task. In this study, a wearable UAR system integrated with an MBHI is proposed to augment the shop floor environment with smart information. A case study is implemented to demonstrate the practicality and effectiveness of the proposed UAR and MBHI system.

Keywords Augmented reality (AR) · Mobile bare-hand interaction (MBHI) · Smart objects (SOs) · Manufacturing shop-floor

1 Introduction

With the development of Industry 4.0, the Internet of Things (IoT) [1, 2], especially technologies involving smart sensors and interfaces, will permeate all contexts of the manufacturing industry, offering convenient and intelligent digital access to physical functionalities of interconnected things. Lee et al. [2] developed an industrial IoT-based smart hub that served as a gateway for existing IoT devices and managed IoT devices at distributed locations, such as smart factories, warehouses, and offices. In smart factories, common objects equipped with some augmented functions, attributes, intelligence, etc., become smart objects (SOs), which are primarily elements built into the structure of a smart environment [3, 4]. In such environments, the SOs are supposed to seamlessly react to changes occurring within the surrounding smart environment and interactions from users to provide useful and customized services [5, 6]. Interaction with SOs should become ubiquitous, i.e., at anywhere and anytime, and as transparent and natural as conducting daily activities to enable users to discover and interact with SOs, e.g., smart machines and workpieces, in a smart manufacturing environment. Actual interactions between users and SOs require an intuitive interaction interface that presents the object's functionality, e.g., providing guidance for an ongoing task to a user and transforming user input (e.g., natural hand gestures) into interaction commands that trigger actions at the SOs.

As a popular computing-assisted technology, augmented reality (AR) can provide a seamless interface that serves as a connection between the real and virtual world, such that the connections between users and the smart environment can be enhanced [7, 8]. AR has been widely implemented as an intuitive graphical interface for users to interact with

✉ S. K. Ong
mpeongsk@nus.edu.sg

¹ Department of Mechanical Engineering, Faculty of Engineering, National University of Singapore, Singapore

and program the SOs [9, 10]. Ubiquitous AR (UAR) is AR integrated in ubiquitous multimedia environments supported by ubiquitous computing (e.g., ubiquitous tracking, ubiquitous information presentation, and ubiquitous interaction) [11]. Screen-based graphical user interfaces (GUIs) for interacting in smart environments have been widely implemented for a long time [12, 13]. However, traditional screen-based GUIs, e.g., desktop-based, projection, and mobile device screens, display information on a monitor such that the information is detached from the physical environment, regardless of the screen's location. Moreover, a screen-based interface requires users to interact with the display device physically; this prevents users from using multiple devices simultaneously or devices that they cannot reach physically. Compared with traditional devices, more flexible interaction methods and tools have been developed in some computer-vision based AR systems. Vision-tracked fiducial markers have been used for realizing interactions between users and SOs [14, 15]. A handheld device-based interaction tool has been proposed for two-dimensional (2D) to three-dimensional (3D) AR modeling [16]. Although these devices are user-friendly, they are still not natural.

Human hands have been used for interactive manipulation and navigation in AR systems to provide more natural and user-friendly human-computer interactions (HCIs) [17–19]. Compared with traditional HCI devices, human hand interactions are more intuitive and convenient, and less disturbing. Human hand interactions can be classified into two types: device-assisted and bare-hand interactions. Data gloves have been widely used to capture hand motions in device-assisted hand interaction systems. The spatial positions of the hands and joint angles can be obtained directly from the sensors on the data gloves. However, data gloves and connecting wires in wired data gloves cause inconvenience to the users. Additionally, data gloves are often unaffordable for ordinary users. In bare-hand interaction systems, no devices or wires are attached to a user's hands. Natural features on the hands and hand gestures are detected and tracked to realize a natural HCI. Human hand interaction is more efficient than traditional input devices, and users prefer bare-hand interactions over interaction involving other devices [20]. Herein, a mobile bare-hand interface (MBHI) is proposed; it allows a user's ubiquitous interactions with all connected SOs through explicit and simple manipulations as well as programmable behaviors with bare-hand gestures, which are natural to learn.

In this study, major improvements have been made to transform the previous desktop-based bare-hand interface system to an MBHI system, such that users can navigate in a smart environment and interact freely with the SOs with bare-hand gestures. The MBHI enables users to interact

with SOs in the user's first-person view directly, thereby providing a continuous ubiquitous user experience in a smart environment without distracting the user's attention when the user is exploring the surroundings. Developed on a mobile processor, the MBHI provides gesture-based freehand interactions with greater convenience and intuitiveness. Users can interact with SOs including machines, workbenches, and material rack to complete tasks, such as control machine operations, gain access to schedule information, and program the SOs. Context adaptive AR information overlays are aligned with the SOs to present their attributes and functions. A case study in a smart shop floor is implemented to demonstrate the characteristics of the MBHI system.

The remainder of this paper is organized as follows. Section 2 provides an overview of related work pertaining to a user's interaction with SOs in a smart environment. An overview of the MBHI system is presented in Section 3, and the details of the mobile 3D bare-hand interaction methodology and interaction methods between users and SOs are presented in Section 4. Section 5 discusses a case study and the experimental results of the proposed method. Conclusions and future work are presented in Section 6.

2 Related works

2.1 Natural-hand-gesture-based interaction

Smart interaction, which refers to natural and intuitive user interfaces for the manipulation of augmented contents and interaction with SOs in a smart environment, has become a popular topic [21]. Velaz et al. [22] conducted a set of experiments with interaction approaches including mouse, haptic devices, and marker-less hand 2D/3D motion capture systems to compare the effects of these interaction approaches on the manipulation of augmented contents in a smart AR environment. A sensor-based glove was implemented with a head-mounted display (HMD) to enable users to be immersed in an AR scene and interact with the virtual smart components [23]. Recently, hand-based human computer interface has become a popular research topic in IoT and AR-based smart environments (see Table 1). Hand-based HCI aims to provide a more direct approach for manipulating and interacting with augmented contents of SOs. Bare-hand interaction method comprises two main groups: indirect and direct interactions. Indirect bare-hand interaction uses noncontact hand gestures for interaction, and these bare-hand detection and tracking systems identify bare-hand gestures from video streams and use them as commands [24–26]. Meanwhile, interaction and control are triggered when contact is established between the human bare hands and virtual objects in direct

Table 1 Comparison of the existing natural hand gesture based interaction

Interaction methods	Representative works	Advantages	Disadvantages
Device-based methods	[22]	Easy to implement	Unnatural to use
Sensor-based methods	[23]	Fast response	Need to wear sensors, prone to drift errors
Hand-based methods	[24–27]	Most natural way to interact with objects	Computation intensive

bare-hand interactions [27]. This system recognizes user operations from the geometrical information of a triangle that is defined using three marker positions. A threshold that is set to 3 cm is used to determine whether the user is holding the object. However, this will affect the user's visual sensation because objects have different sizes and shapes. A 3D vector is used to derive the angle for rotating the object. However, the object will not be rotated when the user rotates his/her hand along this 3D vector. In the proposed system, a hand coordinate system is established on the user's hand to provide more accurate pinch operations.

2.2 User-smart object interaction

The rapid development of portable viewing devices with powerful processing and graphical rendering hardware, e.g., tablets, has resulted in an increased usage of AR as an intuitive user interface for information display and interaction with machines and services in smart factories and shop floors [3, 28, 29]. In a smart environment, machines, which are SOs, should be capable of operating in a coordinated manner and learning from their operations and events. Hence, it is highly important to program the SOs. Bergesio et al. [30] proposed an interaction system to allow users to identify objects in a smart space using a smartphone.

2.3 Summary

A review of state-of-the-art natural user interfaces and interaction methods with SOs reveals that a natural user interaction system that can be used to support mobile bare-hand-based manipulation and interaction with augmented contents of SOs in a smart environment has not been developed yet. Specifically, a user's ubiquitous identification, manipulation, interaction, and programming of SOs

with bare hands during his/her navigation in a smart environment has not been considered. A natural and intuitive mobile bare-hand-based interaction method that does not require calibration and attached devices and can provide prompt and precise responses to allow a user to interact with SOs directly and ubiquitously does not exist.

3 System overview

The proposed MBHI is a smart human-machine interface, enabling the recognition of a user's hand gesture such that the user can interact with smart machines with bare hands. In this MBHI system, a smart HMD is configured with RGB-D cameras, motion capture systems, processors, and network capabilities. The system can display augmented first-person views to the users (see Fig. 1). The RGB-D camera is used to capture the scene of the surrounding physical environment, as well as recognize and track the user's hands gestures in the environment. By wearing the smart HMD, the user can navigate within the smart shop floor, and all the important manufacturing information can be highlighted directly in his/her field of view using AR contents/icons overlaid neatly over the real world. Specifically, the AR contents/icons are created and geo-registered using a Web service and visualized based on the position and direction of sight of a user. The user can access the manufacturing information by interacting directly with the AR contents/icons of the smart machines. Figure 2 shows the MBHI that allows users to interact with the smart machines, and the interaction design is presented in the next section. The MBHI comprises four modules: wearable devices, hand feature extraction and tracking, hand gesture recognition, and interaction scheme (see Fig. 2). These modules are described in detail in Sections 4 and 5.

4 Methodology

4.1 Smart machines in a smart shop floor

The service objectives of a smart machine are typically consistent with those of the corresponding nonsmart machine [31]. The modeling of smart objects has been performed extensively [30]. In this study, a smart architecture to describe smart machines is developed. In the proposed smart shop floor, a smart machine is an SO of the corresponding physical machine (see Fig. 3). It comprises seven elements: smart sensors, actuators, processors and memory, data storage, algorithms/functions, physical machine, and communication module. In a smart machine, sensors are used to detect value changes in quantities (e.g., temperature and pressure) or detect events (e.g., RFID

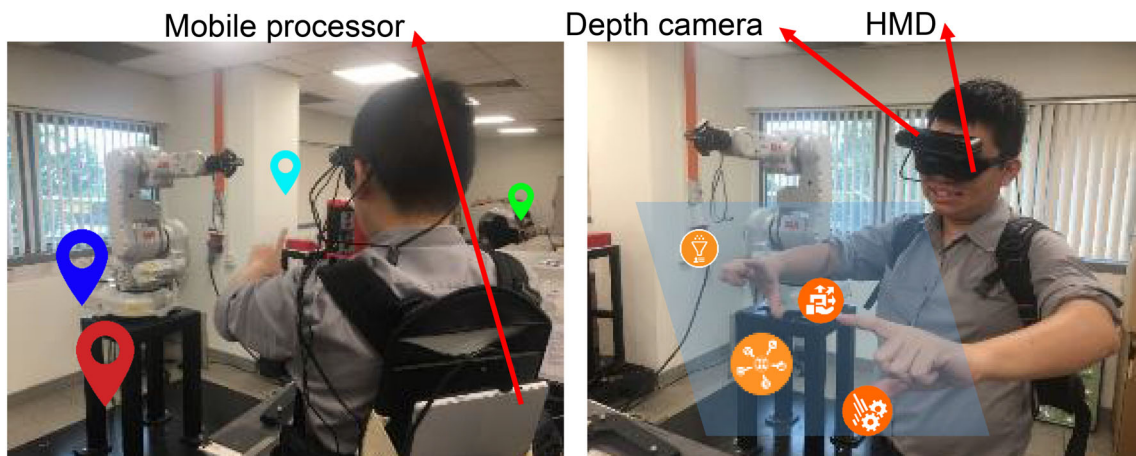


Fig. 1 MBHI

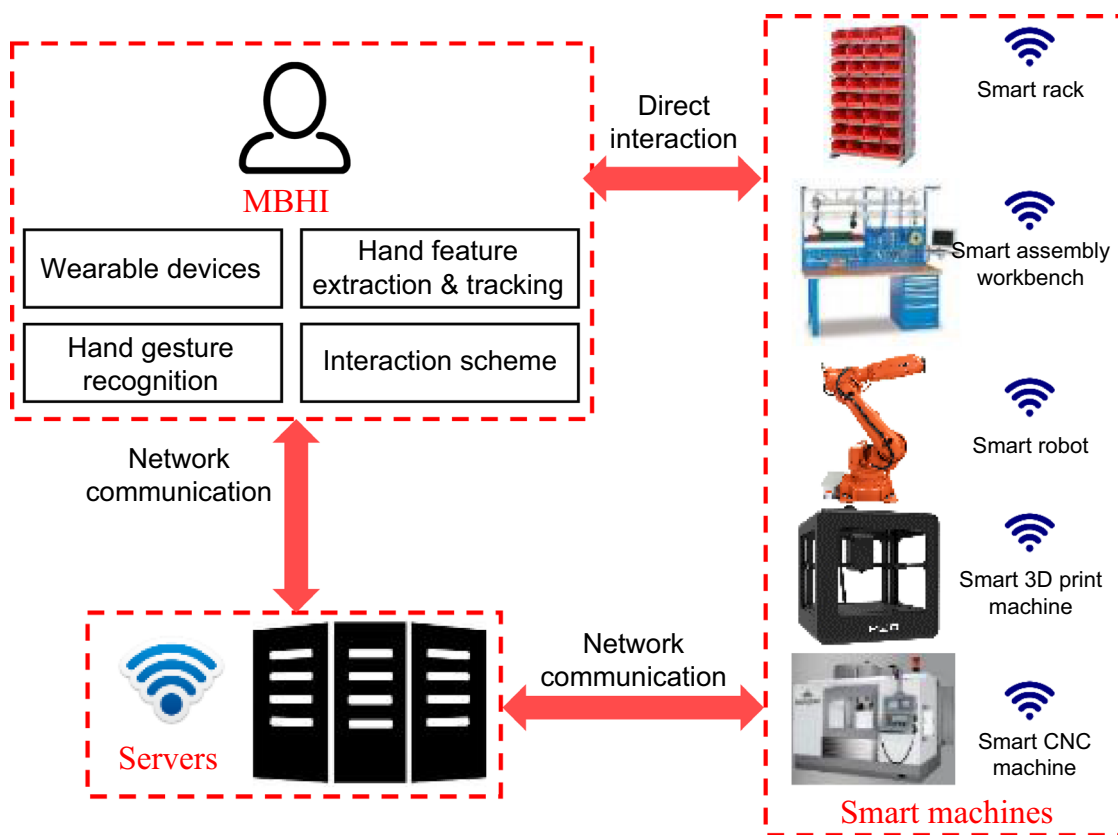


Fig. 2 Architecture of MBHI

reader), based on the demand of the system/user or continuously. The detected values/events are propagated and stored in data storage. Actuators are enabled based on the functions of a smart machine (e.g., display the task instructions and adjust machining speed and/or feed). An actuator is enabled or disabled according to the status data based on algorithms/functions. The functions of a smart machine are a set of algorithms and rules that obtain an input from the captured status data and interaction

information from the user, update the status data, and provide feedback to the user. It determines the change(s) in an SO based on data received from the sensors on the corresponding physical machine. It controls the communications between SOs and smart machines. Processors are responsible for executing the algorithms and provide results from the execution of functions. Depending on the complexity of a smart machine, the processor can be a high-performance central processing unit (CPU), mobile

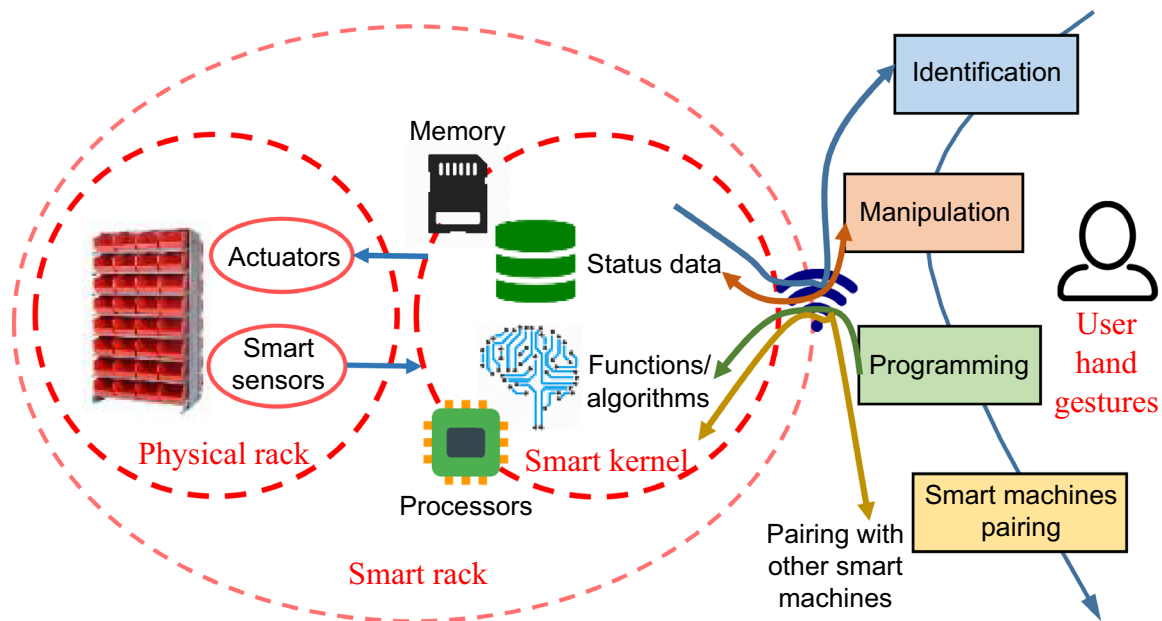


Fig. 3 Model of smart machine

processor, or microcontroller. Data storage is used to store the current and historical status data and task data (e.g., task schedules, and task guidance). The communication module enables a smart machine to exchange status data and task data with other smart machines, with users and/or with the server. Figure 3 shows all the elements and their roles during an interaction. The design details of the interaction between users and smart machines are illustrated in Section 4.3.

4.2 Hand feature extraction and tracking

In this study, human hands were used as an interaction tool in a UAR smart environment. In the first step of hand feature extraction and tracking, the hand regions are obtained from the input video stream, and the fingertips are extracted from these hand regions. An RGB-D camera was used in this study to retrieve the 3D information of these fingertips. The flowchart of the algorithm for 3D hand feature extraction and tracking is shown in Fig. 4.

4.2.1 Hand segmentation

In the 3D direct bare-hand interaction methodology proposed herein, the continuously adaptive mean-shift

algorithm [32] was used to track human hands using a 1D histogram, which comprises quantized channels from the hue saturation value (HSV) color space. The hues were separated from the saturation in the HSV color space; the intensity was used to create a model of the desired hue using a color histogram. The resulting discrete hue model was used to segment the hand region in the input video stream.

For each frame of the input video stream, a trained skin color histogram was used to categorize every pixel as a skin-color or nonskin-color pixel. It was assumed that a large portion of the input color image was the hand region. The hand contours were detected and extracted using the OpenCV library [33]. The connected component (a number of connected pixels) with the largest perimeter was selected as the hand contour.

4.2.2 Hand feature extraction and tracking

In the hand feature extraction method proposed herein, fingertips were identified from the hand contour using a curvature-based algorithm [34]. The dot product of $P_i P_{i-1} (P_{i,i-1})$ and $P_i P_{i+1} (P_{i,i+1})$ based on Eq. (1) was used to determine the curvature of a contour point, where P_i is the i th point in the hand contour; P_{i-1} and P_{i+1} are the

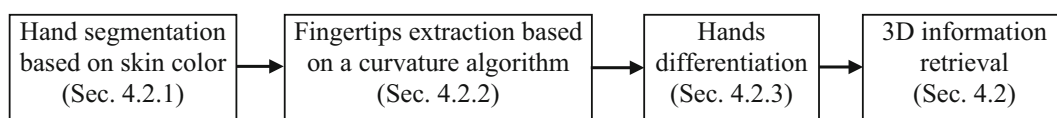


Fig. 4 Flowchart of the bare-hand interaction methodology

preceding and following points, respectively; l is the point index on the hand contour. In the proposed system, l is set to 15.

$$C_l(P_i) = \frac{P_{i,i-1} \cdot P_{i,i+1}}{\|P_{i,i-1}\| \|P_{i,i+1}\|}. \quad (1)$$

Points with curvature values greater than a threshold of 0.5 were selected as fingertip candidates. The directions indicated by the cross product of the two vectors, P_{i-1} and P_{i+1} , were used to determine whether a candidate was a fingertip point or a valley point between two fingers. An ellipse was fitted to the hand contour using the least-squares fitting method, and the center point of the ellipse was set as the center of the hand. All the candidates were separated into different candidate groups. The distance between each candidate and the hand center was computed. The candidate with the largest distance from the hand center was identified as a fingertip. The hands and fingertips could be tracked by executing the detection and extraction processed for each frame, as the hand segmentation and features extraction processes were fast.

4.2.3 Differentiation of hands

The system can differentiate a user's hands automatically to realize a dual-hand interaction. At the start of the hand differentiation process, a user is required to place either one or both of his/her hands with the palm(s) facing down in the camera's view to ensure that all five fingertips can be detected. For each hand, the tip of the thumb P_{th} is determined as the farthest fingertip from the mean position P_m of all the five fingertips. The cross product D_H of $P_{HC}P_{th}$ ($P_{HC,th}$) and $P_{HC}P_m$ ($P_{HC,m}$) is calculated according to Eq. (2), where P_{HC} is the hand center.

$$D_H = \frac{P_{HC,th} \times P_{HC,m}}{\|P_{HC,th}\| \|P_{HC,m}\|}. \quad (2)$$

The hands of a user can be differentiated by examining the direction of the vector D_H , which is represented as Eq. (3).

$$D_H = H_x i + H_y j + H_z k. \quad (3)$$

The number of hands in the camera's view can be determined automatically by the system. When only one hand is present, it will be specified as either the "right hand" or "left hand," and this specification will remain unchanged until the hand is out of the camera's view. If both of the user's hands are in the camera's view, the centers of these two hands are tracked using a matching algorithm that minimizes the displacement between these centers over two successive frames after the hand differentiation process. Hence, these two hands can be differentiated in each frame. It is important to note that the palms

must be facing down, and that the two hands must not cross each other such that the proposed method can be used.

4.3 Interaction design

The interaction workflow illustrated in Fig. 3, i.e., smart machine identification, manipulation, programming, pairing, etc., highlights the role of the smart machines and user, as well as the need for a common interaction scheme between them such that the user can perform ubiquitous interactions with bare hands.

During a user's navigation in a smart environment, when he/she is near a smart machine, the smart distance sensors, which operate continuously, capture the distance between the user and smart machine and transfer the distance information to the smart kernel of the smart machine. When the user is sufficiently close to the smart machine, i.e., the distance is less than a threshold, the smart machine will contact the server automatically to request the user information, including the user ID, IP address of the MBHI system, and 3D positions and poses in real time. The 3D positions and poses are tracked in real time using the motion capture system. After the smart machine has obtained the user information, it will connect with the user through the MBHI system and send the smart machine information, including the smart machine's capabilities, task information, and instructions to the tasks to the user. Therefore, the user becomes aware of the smart machines and their capabilities in the surroundings. The user's AR view is updated based on the real camera pose to render 3D menus from the corresponding viewpoint. Hence, virtual menus are aligned with surfaces of physical objects. The menus are rendered as AR overlays to blend with the video stream and then displayed via the HMD. Simultaneously, users can interact with smart machines using bare-hand gestures. These interactions are interpreted as sockets, which can achieve communication between smart machines through the underlying network.

When a smart machine has been identified by a user, obtaining access to its capabilities and stored task-related data is a common human-smart-machine interaction. In a smart environment, all stored task-related data can be accessed freely using simple hand gestures via the MBHI. Users wearing the HMD can view virtual icons of the available task data (e.g., task schedule, ongoing task information, and task instructions) overlaid around smart machines (see Fig. 5). Essential information is displayed next to the icons. Users can easily access the information and add additional instruction files from other smart machines and/or other computers/servers. For example, users can drag and move the designated files from the source smart machine to the destination smart machine using simple selected gestures and 3D translation, and the



Fig. 5 Virtual icons overlay on the real machine

files will be sent automatically to the target. The entire process is intuitive and does not require users to access the two smart machines. As the MBHI is built based on the IoT concept, in which various different smart machines/devices are connected in a network to allow seamless interactions between them, the MBHI enables information/data to be transferred between smart machines. Using natural user hand gestures, users are not required to learn and remember the built-in interfaces of different devices.

In addition to interactions with smart machines, users can program the smart machines using bare-hand gestures in the AR smart world, thereby creating a personalized interface that combines the capabilities of different devices and complying with the event-condition-action (ECA) paradigm [10]. In this stage, a user can personalize the smart space to according to the requirement of the tasks and his/her preferences (see Fig. 6).

The MBHI method allows device pairing using hand gestures from users. Therefore, the MBHI can consolidate smart machines in a natural and user-friendly manner. In a smart manufacturing environment, many wireless devices can be paired with one another via the MBHI to perform various functions. Supported peripherals include smart sensor peripherals (e.g., air quality sensors) and smart actuators (e.g., fans and air freshener). Traditionally, users must perform tedious operations to pair these devices. However, using the MBHI, users can view the virtual representations and menu options of these devices via their HMDs. The MBHI allows users to pair or connect devices conveniently by pulling the overlays of the devices from one to another in the AR interface. The pairing will be completed automatically in near real time. For example, when a user wishes to pair the air quality sensors of smart machines to the fan/air freshener (see Fig. 7), he/she can drag the virtual representation of the air quality sensors to the fan/air freshener. When the two devices have been paired, they will be displayed in the AR interface as paired.

Next, the user can program the behavior of these paired devices. The procedure of unpairing two devices is similar to that for pairing. Users only need to drag a paired device away from the other paired device and these devices will be disconnected. The MBHI enables devices to be paired in a timely manner and connection of multiple devices.

5 Implementation

The proposed method was implemented using visual C++ in the Microsoft Windows 10 operating system. The system configuration is shown in Fig. 1. An Astra depth camera was used as the input video device of the system. The output display was a VUZIX WRAP 920 Video iWear, which was a HMD with a 640×480 display resolution in a 67" screen and 31° diagonal field-of-view. The processing unit was a personal computer with a 2.50 GHz Intel Core i5 7200U processor and 8 GB RAM. The Astra camera was used to detect the hand regions to calculate the depth information of the fingertips. All the fingertips were augmented on the image captured by the Astra camera attached on the user's head such that the view shown in the HMD was consistent with the user's real view. Frame rate is commonly used as a measurement of how quickly an application displays consecutive images. An application is generally considered real time if the frame rate is at least 15 frames per second. The frame rate of this system was approximately 25 frames per second. For the case study, a server was developed to link the cloud services, workers, and resources on the shop floor via a TCP/IP network. The workers used a tablet (Microsoft Surface) as the UAR device. The UAR application was executed on the surface tablet and communicated with the server to obtain updated data. RFID sensors as well as environment sensors (e.g., temperature and distance) were distributed in the smart shop floor environment.

5.1 System accuracy

Root mean square (RMS) errors were used to determine the accuracy of the MBHI method developed in this study. For a set of n values $\{x_1, x_2, \dots, x_n\}$, the RMS value x_{rms} was calculated using Eq. (4).

$$x_{\text{rms}} = \sqrt{\frac{x_1^2 + x_2^2 + \dots + x_n^2}{n}} \quad (4)$$

5.2 Accuracy of fingertip detection

The RMS error estimation method [35] was used to determine the accuracy of the fingertip detection method

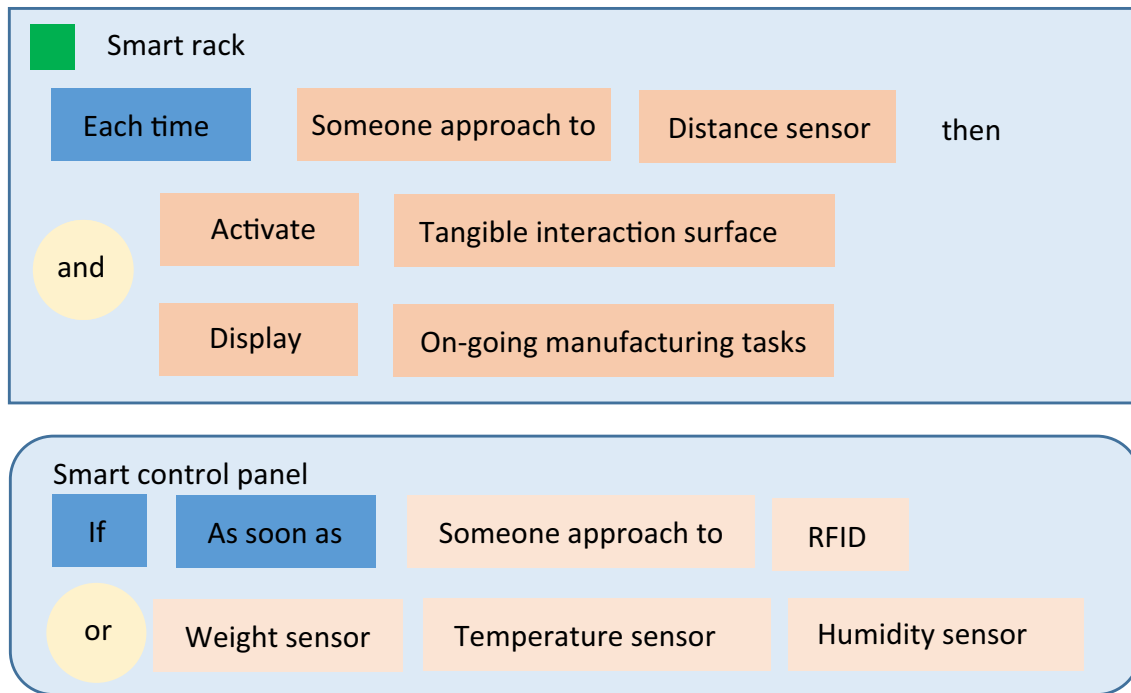


Fig. 6 Visual programming editor for MBHI

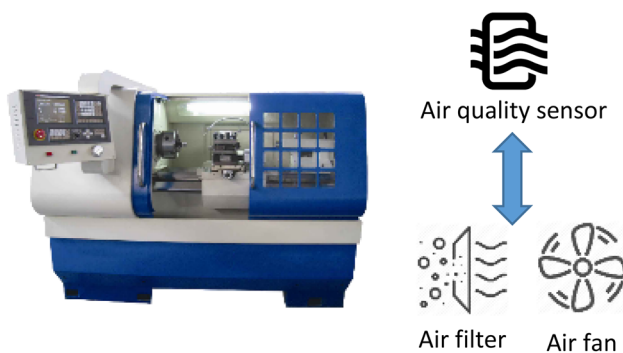


Fig. 7 Pairing of the smart objects

for 2D images. A user points at a reference point using the tip of his/her index finger during the error estimation method, as shown in Fig. 8, where the red triangles are the reference points and the circles are the 2D positions of the fingertips. A total of 1 000 fingertip positions were collected for each reference point during the error estimation process. The RMS errors were calculated based on the differences between the coordinates of the index fingertip and the coordinates of the reference points. For the MBHI method in this study, the RMS errors were determined to be 1.13 and 1.24 pixels with reference to the depth camera space in the x - and y -axes, respectively.

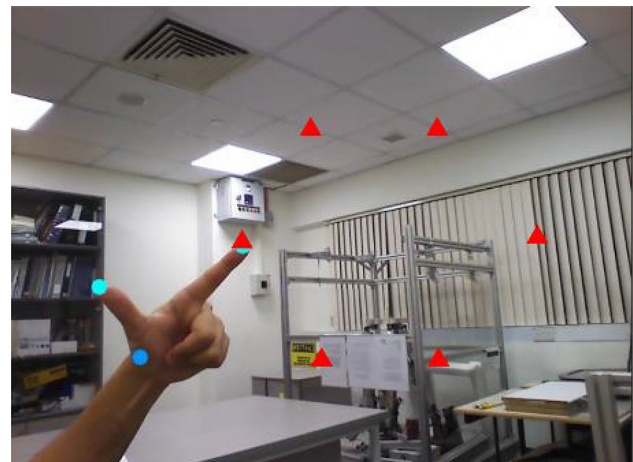


Fig. 8 Accuracy estimation of the fingertip detection method

5.3 Case study

In the proposed smart environment, a first-person view was provided, in which the user could move around and operate within the environment developed. The MBHI was implemented to empower users with tools to control, augment, and interact with the smart environment (smart shop floor) ubiquitously. Contrary to the method of artificial intelligence that generates services automatically by mining data stored on the cloud, the proposed method introduces the user into the development loop of a smart environment, such that he/she can program the SOs to

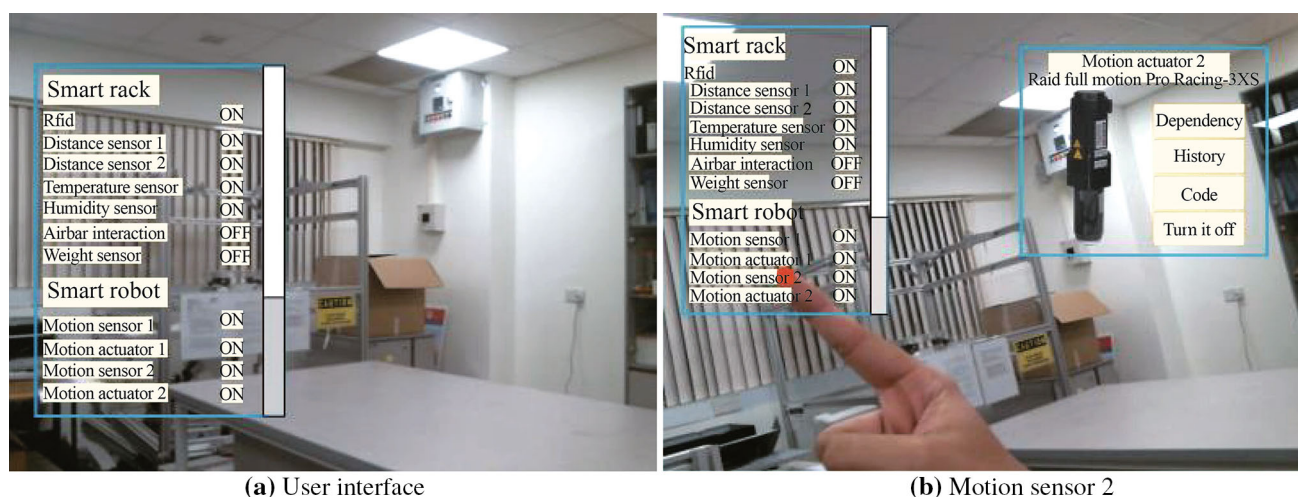


Fig. 9 Basic activities enabled for the user to interact with smart objects

enable the smart environment to realize customized functions. The user's AR view is updated based on real camera poses to render 3D menus from the user's viewpoint; hence, the virtual menus will be aligned with surfaces of physical objects. The menus are displayed as AR overlays to merge with the video stream and then display via the HMD. Users can interact with smart machines using bare-hand gestures, and these bare-hand interactions are interpreted as sockets to achieve communication between smart machines through the underlying network. When a smart machine is identified by a user, obtaining access to its capabilities and storing task-related data is a common human-smart-machine interaction. In a smart environment, all the stored task-related data are interactive and can be accessed freely with simple hand gestures via the MBHI.

Figure 9 shows the user interface for the basic activities implemented in the smart shop floor. The list in the figure shows the smart devices, with their status information organized by dependency relations. When the user selects motion sensor 2, which is attached to the smart robot, a new panel pops up and shows the status information and control buttons (see Fig. 9b). As shown in the panel, the user can verify the dependency relations of this sensor to the SO that it is attached to, acquire the history sensor readings, program the customized functions, and turn the device on/off. The user-friendly buttons allow users to activate the various options with bare-hand gestures.

The case study illustrated is a proof-of-concept investigation of the proposed solution. Therefore, it lacks the complexity of a typical shop floor, which may consist of many machines and equipment. However, the basic principles have been illustrated using bare-hand gestures to interact with equipment controls, and AR displays can provide the users with timely and useful information in addition to the views of the equipment.

6 Conclusion

Herein, an MBHI method was proposed for ubiquitous interactions with SOs in a smart manufacturing shop floor. This direct 3D bare-hand interaction method was efficient and user-friendly. Hand and fingertip differentiation algorithms were developed to achieve a dual-hand interaction interface. Using depth vision technologies, 3D information of the fingertips could be retrieved and used for direct bare-hand interaction operations. It was experimentally observed that users could use the MBHI method to manipulate virtual objects and interact with SOs ubiquitously. The MBHI provided a dual-hand interface that afforded direct and intuitive interactions between a user and the virtual objects and SOs in an AR environment. The case study demonstrated the applicability of the system. AR-based applications, such as AR-based smart environments, education, and rehabilitation would benefit from the proposed MBHI interaction methodology. Further research is necessary to obtain an effective algorithm for addressing 3D manipulation gestures. The self-occlusion problem will be addressed for a more robust interaction process.

Acknowledgements This research was supported by the Singapore A*STAR Agency for Science, Technology and Research Public Sector Research Funding Programme (Grant No. 1521200081).

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright

holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Monostori L, Kadar B, Bauernhansl T et al (2016) Cyber-physical systems in manufacturing. *CIRP Ann Manuf Technol* 65(2):621–641
2. Lee CKM, Zhang SZ, Ng KKH (2017) Development of an industrial Internet of things suite for smart factory towards re-industrialization. *Adv Manuf* 5:335–343
3. Yew AWW, Ong SK, Nee AYC (2016) Towards a griddable distributed manufacturing system with augmented reality interfaces. *Robot Comput Integr Manuf* 39:43–55
4. Voinov N, Chernorutsky I, Drobtintsev P et al (2017) An approach to net-centric control automation of technological processes within industrial IoT systems. *Adv Manuf* 5:388–393
5. Kortuem G, Kawsar F, Sundrammorthy V (2010) Smart objects as building blocks for the internet of things. *IEEE Internet Comput* 14(1):44–51
6. Burmeister D, Burmann F, Schrader A (2017) The smart object description language: modeling interaction capabilities for self-reflection. In: *Proceedings of 2017 IEEE international conference on pervasive computing and communications workshops*, pp 503–508
7. Grubert J, Langlotz T, Zollmann S et al (2016) Towards pervasive augmented reality: context-awareness in augmented reality. *IEEE Trans Vis Comput Graph* 23(6):1706–1724
8. Wang X, Ong SK, Nee AYC (2016) A comprehensive survey of augmented reality assembly research. *Adv Manuf* 4(1):1–22
9. Teyssier M, Cliquet G, Richir S (2016) ArLive: unified approach of interaction between users, operable space and smart objects. In: *Proceedings of the 2016 virtual reality international conference*, ACM
10. Wirtz H, Ruth J, Serror M et al (2015) Enabling ubiquitous interaction with smart things. In: *Proceedings of 2015 12th annual IEEE international conference on sensing, communication, and networking*, pp 256–264
11. Wang X, Yew AWW, Ong SK et al (2019) Enhancing smart shop floor management with ubiquitous augmented reality. *Int J Prod Res*. <https://doi.org/10.1080/00207543.2019.1629667>
12. Sollich H, von Zadow U, Pietzsch T et al (2016) Exploring time-dependent scientific data using spatially aware mobiles and large displays. In: *Proceedings of the 2016 ACM on interactive surfaces and spaces*, pp 349–354
13. Besancon L, Issartel P, Ammi M et al (2017) Hybrid tactile/tangible interaction for 3D data exploration. *IEEE Trans Vis Comput Graph* 23(1):881–890
14. Chong JWS, Ong SK, Nee AYC (2007) Methodologies for immersive robot programming in an augmented reality environment. *Int J Virtual Real* 6(1):69–79
15. Krichenbauer M, Yamamoto G, Taketomi T et al (2014) Towards augmented reality user interfaces in 3D media production. In: *Proceedings of 2014 IEEE international symposium on mixed and augmented reality (ISMAR 2014)*, pp 23–28
16. Feng L, Yang X, Xiao S (2017) MagicToon: a 2D-to-3D creative cartoon modeling system with mobile AR. In: *Proceedings of 2017 IEEE virtual reality*, pp 195–204
17. Wang X, Ong SK, Nee AYC (2016) Real-virtual components interaction for assembly simulation and planning. *Robot Comput Integr Manuf* 41:102–114
18. Wang X, Ong SK, Nee AYC (2016) Multi-modal augmented-reality assembly guidance based on bare-hand interface. *Adv Eng Inf* 30(3):406–421
19. Arroyave-Tobon S, Osorio-Gomez G, Cardona-McCormick JF (2015) Air-modeling: a tool for gesture-based solid modelling in context during early design stages in AR environments. *Comput Ind* 66:73–81
20. Lin S, Cheng HF, Li W et al (2017) Ubii: physical world interaction through augmented reality. *IEEE Trans Mobile Comput* 16(3):872–885
21. Korinth M, Sommer-Dittrich T, Reichert M et al (2019) Design and evaluation of a virtual reality-based car configuration concept. In: *CVC 2019 advances in computer vision*, pp 169–189
22. Velaz Y, Arce JR, Gutierrez T et al (2014) The influence interaction technology on the learning of assembly tasks using virtual reality. *J Comput Inf Sci Eng* 14(4):041007
23. Valentini PP (2009) Interactive virtual assembling in augmented reality. *Int J Interact Des Manuf* 3(2):109–119
24. Dhawale P, Masoodian M, Rogers B (2006) Bare-hand 3D gesture input to interactive systems. In: *Proceedings of the 7th international conference on NZ Chapter of the ACM's SIG on human-computer interaction*, vol 158, pp 25–32
25. Wang RY, Paris S, Popovic J (2011) 6D hands: Markerless hand tracking for computer aided design. In: *Proceedings of the 24th annual ACM symposium on user interface software and technology*, pp 549–557
26. Hilliges O, Izadi S, Wilson AD et al (2009) Interactions in the air: adding further depth to interactive tabletops. In: *Proceedings of the 22nd annual ACM symposium on user interface software and technology*, pp 139–148
27. Kojima Y, Yasumuro Y, Sasaki H et al (2001) Hand manipulation of virtual objects in wearable augmented reality. In: *Proceedings of the seventh international conference on virtual systems and multimedia*, pp 463–469
28. Erkoyuncu JA, del Amo IF, Mura MD et al (2017) Improving efficiency of industrial maintenance with context aware adaptive authoring in augmented reality. *CIRP Ann Manuf Technol* 66(1):465–468
29. Mourtzis D, Vlachou A, Zogopoulos V (2017) Cloud-based augmented reality remote maintenance through shop-floor monitoring: a product-service system approach. *J Manuf Sci Eng* 139(6):061011
30. Bergesio L, Bernardos AM, Casar JR (2017) An object-oriented model for object orchestration in smart environments. *Proc Comput Sci* 109:440–447
31. Yachir A, Djamaa B, Mecheti A et al (2016) A comprehensive semantic model for smart object description and request resolution in the internet of things. *Proc Comput Sci* 83:147–154
32. Hu W, Sun Y, Li X et al (2016) Cam-shift target tracking approach based on back projection method with distance weights. In: *Proceedings of 2016 IEEE international conference on wavelet analysis and pattern recognition (ICWAPR)*, pp 252–257. <https://opencv.org>. Accessed on Oct 19 2017
33. Rautaray SS, Agrawal A (2015) Vision based hand gesture recognition for human computer interaction: a survey. *Artif Intell Rev* 43(1):1–54
34. Shen Y, Ong SK, Nee AYC (2011) Vision-based hand interaction in augmented reality environment. *Int J Hum Comput Interact* 27(7):523–544



S. K. Ong is Associate Professor at NUS. Her research interests are virtual and augmented reality applications in manufacturing and life cycle engineering. She is Fellow of CIRP. She has published 5 books and over 330 international refereed journal and conference papers. She received 2004 OYME Award from SME and 2009 Emerging Leaders Award in Academia (US Society for Women Engineers).



A. Y. C. Nee received his Ph.D. and D.Eng. degrees from UMIST, Manchester. He is currently Emeritus Professor in the Department of Mechanical Engineering of NUS. He is Fellow of CIRP, SME and Academy of Engineering Singapore. His research interests are in manufacturing engineering, augmented reality and digital twin. He is editor-in-chief of IJAMT and Executive editor-in-chief of AIM.



X. Wang received a BS degree in Mechanical Engineering from Tsinghua University, P.R.C., in 2010. He received his PhD degree from the Mechanical Engineering Department, National University of Singapore, in 2016, specializing in bare-hand interaction in augmented reality application for product assembly and simulation.