Sensors and Materials, Vol. 38, No. 4 (2026) 2025–2051 2025
MYU Tokyo

S & M 4421

Gesture Recognition Ordering System Based on Kinect v2 Stereo Vision

Bing-Yan Chen and Cheng-Yu Peng*

Department of Electronic Engineering, National Chin-Yi University of Technology,
No. 57, Sec. 2, Zhongshan Rd., Taiping Dist., Taichung 41170, Taiwan (R.O.C.)

(Received December 16, 2025; accepted April 6, 2026)

Keywords: ordering system, stereo vision, order meals, Kinect, skeleton tracking

We propose a contactless automated ordering system utilizing Kinect v2 sensing. The system applies continuous-wave indirect time-of-flight (CW-iToF) technology to detect infrared phase shifts. This sensing method generates precise 3D data by extracting 25 skeletal feature points. The system selects six key joints to formulate four three-dimensional (3D) angular features for gesture recognition. Our adaptive geometric model utilizes triangulation to calibrate interaction regions using tester height and standing distance. This sensor-driven approach achieves a recognition success rate of 96.5% at 1.5 m and 95.1% at 2.5 m. The system identifies the selected meal and instructs a robotic arm for food preparation. This architecture establishes a fully contactless and hygienic automated dining framework.

1. Introduction

Public health concerns have significantly increased the demand for contactless services. The food and beverage industry urgently requires hygienic and intuitive ordering methods. Researchers have explored various contactless ordering technologies to improve hygiene. Lee *et al.* developed an augmented reality ordering system. This system allows customers to scan menus and unlock digital ordering interfaces.(1) Goto *et al.* utilized electrooculogram signals for interaction. Their method extracts human intentions directly from facial electrode measurements.(2) Chen *et al.* designed a vision-based approach. They used handheld cameras to capture menu selections and transmit order data.(3) Shen *et al.* created a smart ordering network that utilizes ZigBee technology to streamline restaurant service and data management.(4) Kavitha *et al.* combined virtual reality with holographic displays. Their system projects digital menus onto tables to eliminate physical menu cards.(5) Sajnani and Patel applied artificial intelligence to the ordering process. They used natural language processing and machine learning to interact with customers automatically.(6)

However, current commercial systems have significant limitations. Quick response (QR) code systems lack intuitive interaction and require personal scanning devices. Self-service

*Corresponding author: e-mail: peng@ncut.edu.tw
https://doi.org/10.18494/SAM6124

ISSN 0914-4935 © MYU K.K.

kiosks reduce direct human contact but still require physical touchscreen interaction. This physical contact retains indirect risks of surface contamination.

The Kinect v2 depth sensor provides a robust solution for these interaction challenges. It is a noninvasive optical device that acquires precise three-dimensional skeletal kinematics. This sensing technology enables highly accurate hand gesture recognition without the need for wearable devices.[7,8] Fareed *et al.* developed a gesture-controlled robotic arm. They used Kinect sensors to translate human movements into mechanical operations.[9] Li *et al.* applied this depth sensor to control a NAO robot. Their work facilitated remote human–machine interaction through skeletal tracking.[10] Balbin *et al.* used the sensor to analyze athletic movements. They evaluated golf postures to provide immediate corrective feedback.[11] Limin and Peiyi utilized the sensor for mobile robot navigation. Their system recognized changing gestures to dictate robot movement paths.[12] Wang *et al.* interpreted dynamic human behaviors. They analyzed skeletal features to understand complex activity patterns.[13] Alabbasi *et al.* tracked professional athletes' motions, which allowed trainees to mimic and compare their movements accurately.[14] Hutabarat *et al.* analyzed human gait characteristics. They recorded lower body joints to identify individuals based on distinct walking patterns.[15] Heng *et al.* applied the depth sensor for facial analysis. They tracked facial feature changes to study human expressions.[16] Medical professionals also utilize this skeleton tracking technology. Therapists capture patient joint angles to monitor remote rehabilitation progress effectively.[17,18]

Robotic arms are also increasingly vital in the automated food service industry. They provide the high precision and flexibility required for food preparation. Klievtsova and Fuschlberger developed an IoT-based robotic cocktail-mixing system. This system automates beverage preparation using a collaborative robot and liquid dispensers.[19] Raffik *et al.* emphasized the importance of robot hygiene. They outlined essential maintenance procedures for robots handling food products.[20] Mepani *et al.* created an automated cooking system. Their system features a six-degree-of-freedom robotic arm and an automated ingredient dispenser.[21] Elhousry *et al.* optimized automated food handling. They integrated a UR3 robotic arm with neural networks and computer vision.[22] The market demand for such automated food services continues to grow rapidly. These technologies represent the future of smart dining environments.[23]

We compared our system's performance with those of other gesture recognition methods using different sensing technologies. Nuzzi *et al.* utilized RGB cameras and a Faster R-CNN deep learning model and achieved a 92.12% accuracy for their complete dataset.[24] Kim *et al.* developed the deepGesture scheme using wearable motion sensors and complex deep learning, and attained a 96.18% recall rate.[25] Our proposed system utilizes indirect time of flight (iToF) depth sensing and a lightweight adaptive geometric model. It achieves a superior success rate of 95.8% without the need for wearable devices or high computational power. These results confirm that our sensor-driven geometric approach is more efficient and accurate for real-time ordering scenarios.

2. Data, Materials, and Methods

Food service environments demand hygienic contactless ordering. In this study, we propose a highly precise contactless ordering system that utilizes a Kinect v2 depth sensor. The sensor extracts 25 human skeletal feature points. The algorithm selects six key joints (right/left wrists, right/left shoulders, spine shoulder, spine mid). These joints form four 3D angular features for gesture recognition.

The system estimates the customer's exact spatial position. It calculates the shoulder-to-sensor distance, the shoulder-to-foot height difference, and the midline-to-shoulder lateral distance, then applies geometric triangulation to these spatial metrics. This calculation yields the effective hand-angle range for specific ordering regions.

This method dynamically adjusts gesture control ranges in accordance with customer height and standing distance. The system accurately identifies hand positions to confirm meal selections and seamlessly commands a robotic arm to execute automated food preparation. This framework provides a robust contactless dining solution.

2.1 Kinect v2

Kinect v2 is a depth sensor developed by Microsoft and is the second generation of the Kinect series, which is mainly used for applications such as human gesture recognition (Skeleton Tracking), 3D capture, and depth perception. Kinect v2 has a total of three cameras: from left to right, the RGB camera, and infrared receiving and transmitting cameras, as shown in Fig. 1.

The Kinect v2 features a high-resolution 1920 × 1080 camera operating at 30 frames per second and utilizes iToF technology. By emitting and receiving infrared light, Kinect v2 measures the phase difference to calculate the ToF, enabling precise skeleton tracking. It has a detection range of 0.8 to 4.2 m and can track up to six skeletons simultaneously. As shown in Fig. 2, each skeleton consists of 25 points, with Kinect v2 providing spatial coordinate data for each point 30 times per second.[26]

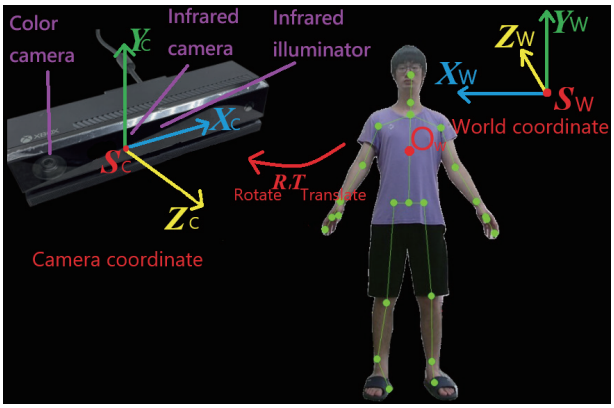


Fig. 1. (Color online) Conversion relationship between world coordinates and camera coordinates.

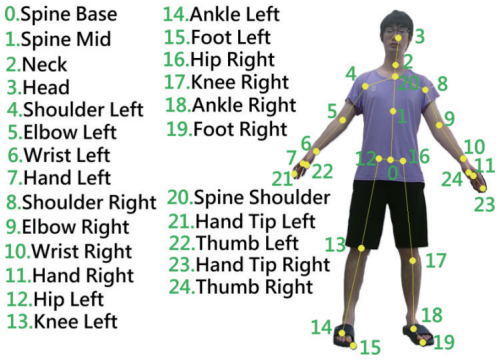


Fig. 2. (Color online) Diagram of the human body and the 25 points on the skeleton.

2.2 Principle of iToF measurement

The Kinect v2 sensor applies the continuous-wave (CW-iToF) method. The system emits infrared light at a fixed frequency *f*. The sensor modulates this light using a sine wave or a 10–100 MHz square wave.[27] It measures the phase shift (φ) of the reflected photons. The system utilizes four control signals. These signals feature precise 90-degree phase delays. The sensor collects four corresponding charge values (Q_1, Q_2, Q_3, and Q_4) and uses them to calculate the exact phase difference, as illustrated in Fig. 3. The mathematical formula is as follows.

$$\varphi = \arctan\left(\frac{Q_2 - Q_4}{Q_1 - Q_3}\right) \tag{1}$$

The system calculates the absolute distance (d) between the target object and the camera. This calculation incorporates the constant speed of light (c) and the light modulation frequency (f). The final exact distance formula is as follows.

$$d = \frac{c\varphi}{4\pi f} \tag{2}$$

2.3 Kinect v2 camera calibration

Distortion occurs when a perspective projection generates an image and can be classified as either radial or tangential distortion. Radial distortion arises when light passing through the edges of the lens bends more than at the center, while tangential distortion is due to the camera not being perfectly parallel to the image sensor. Camera calibration parameters, including internal, external, and distortion parameters, are used to correct the rotation and translation relationships between world coordinates and camera coordinates.

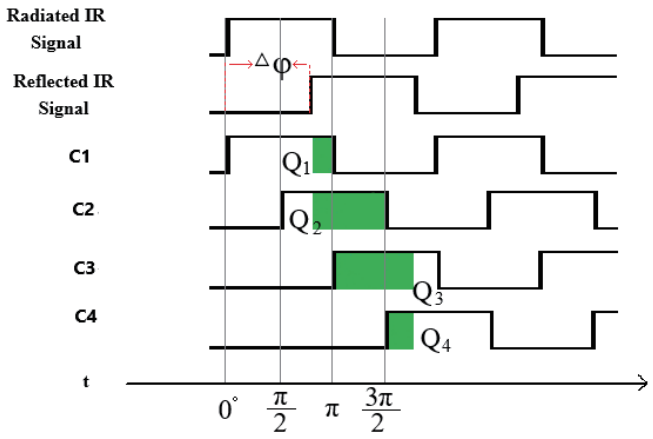


Fig. 3. (Color online) CW-iToF method.

2.3.1 Internal and external parameters of pinhole camera model

In the Kinect v2 skeleton, the spine mid is taken as the real-world coordinate $\boldsymbol{S}_W(x, y, z)$. This coordinate is projected through the camera's lens center onto the image plane, resulting in point $\boldsymbol{s}(u, v)$. The camera's focal length is represented by f, with deviations along the X-axis and Y-axis given by f_x and f_y, respectively. The object's center is imaged at the same point on the plane as through the lens center. The parameters u_0 and v_0 denote the image plane's center, derived using the geometric principles of similar triangles.[28]

$$u = f_x \times \frac{X}{Z} + u_0 \tag{3}$$

$$v = f_y \times \frac{Y}{Z} + v_0 \tag{4}$$

The alignment coordinates show that the real-world spine mid point is projected from the center of the lens onto the camera plane.

$$\boldsymbol{s} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & S_0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_C \\ Y_C \\ Z_C \end{bmatrix} = \boldsymbol{P}\boldsymbol{S}_C \tag{5}$$

The homogeneous coordinate $\boldsymbol{s}$ is found on the image plane, with $\boldsymbol{S}_0$ indicating the skew rate, which is typically zero. A nonzero skew rate means that the image plane parameters do not form a right angle with the X-axis or the Y-axis. $\boldsymbol{P}$ stands for the internal parameter matrix, and $\boldsymbol{S}_C$ represents the camera's coordinate $[X_C, Y_C, Z_C]^T$.

The external parameters of the camera define the relationship between real-world coordinates and camera coordinates, facilitating the conversion between the two coordinate systems. This transformation is achieved using the rotation matrix R and translation matrix t, which align the coordinates, as illustrated in Fig. 4.

The conversion process between camera coordinates and real-world coordinates is described as follows.[29]

$$\boldsymbol{s}_c = \begin{bmatrix} X_C \\ Y_C \\ Z_C \end{bmatrix} = R \begin{bmatrix} X_W \\ Y_W \\ Z_W \end{bmatrix} + t = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} X_W \\ Y_W \\ Z_W \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} = \boldsymbol{RS} + t \tag{6}$$

In this context, $\boldsymbol{S}_C$ represents the camera coordinates, expressed as $[X_C, Y_C, Z_C]^T$. $\boldsymbol{R}$ denotes the rotation matrix, t is the displacement vector, and $\boldsymbol{S}$ represents the real-world coordinates, expressed as $[X_W, Y_W, Z_W]^T$.

2.3.2 Distortion model of camera lens

Since the camera utilizes a lens, distortion correction must be considered. During the imaging process, two types of distortion can occur in the camera lens: radial distortion and tangential distortion. Therefore, the camera model must incorporate corrections for both radial and tangential distortion. The following formula will be employed for this correction.[29,30]

$$\boldsymbol{X}_g = \begin{bmatrix} 2k_3 x_n y_n + k_4 \left(r_2 + 2x_n^2 \right) \\ k_3 \left(r^2 + 2y_n^2 \right) + 2k_4 x_n y_n \end{bmatrix} \tag{7}$$

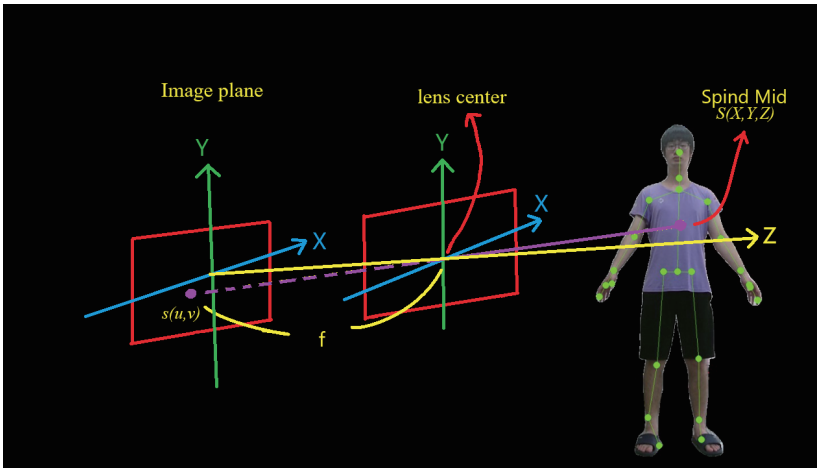


Fig. 4. (Color online) Pinhole camera model for coordinate conversion.

$$X_k = \left(1 + k_1 r_2 + k_2 r_4 + k_5 r_6\right) x_n + x_g, \tag{8}$$

The camera's projection point in the coordinate system is given by $\boldsymbol{S} = [X, Y, Z]^T$ while the corresponding point on the image plane is $\boldsymbol{s} = [u, v]^T$. This point is then normalized to $\boldsymbol{S}_n = [x_n, y_n]^T = [x_n/z_n, y_n/z_n]^T$, where $r^2 = x^2 + y^2$. Here, (x, y) are the coordinates corrected for radial distortion. The factors affecting radial distortion are k_1, k_2, and k_5, while k_3 and k_4 influence tangential distortion. These corrections allow for the final calculation of the image coordinates as

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} f_x & 0 \\ 0 & f_y \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} + \begin{bmatrix} u_0 \\ v_0 \end{bmatrix}. \tag{9}$$

2.4 Skeleton feature point marking

Shotton *et al.*[8] presented a technique for pixel-level body part categorization and joint location estimation. The technique involves calculations of the probability distribution of each body part at the pixel level, and then using mean shift pattern detection. It assigns 31 labels to body parts, covering the entire body. Some labels focus on directly locating skeletal joints, while others help fill in the gaps. These labels can be used together to predict other joints.

2.4.1 Depth image features

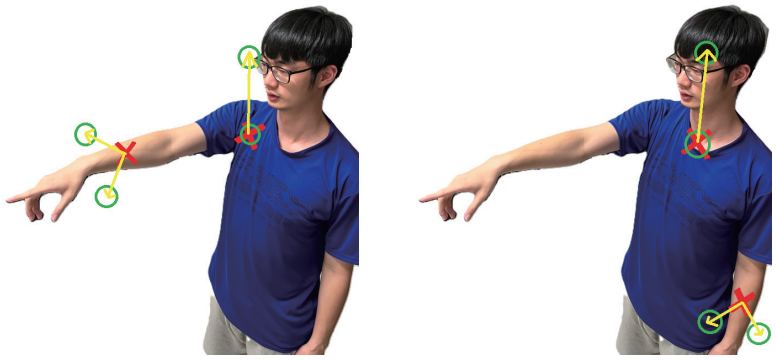
At a specified pixel location x, the feature computation is carried out as

$$f_\theta(I, x) = d_I\left(x + \frac{u}{d_I(x)}\right) - d_I\left(x + \frac{v}{d_I(x)}\right), \tag{10}$$

where $d_I(x)$ refers to the depth of pixel x in image I, and the parameter $\theta = (u, v)$ represents the offsets for u and v, as shown in Fig. 5. The depth of pixel x in the image defines the depth of the feature. Normalizing the offsets using $1/d_I(x)$ guarantees depth invariance for the feature, ensuring that a consistent world-space offset has a uniform effect regardless of the pixel's distance from the camera, as long as it is located at a stable position on the body. This achieves translational invariance in three-dimensional space and reduces perspective distortion. When the offset is pointing to a pixel outside the background or image boundary, the depth probe $d_I(x')$ is set to a large positive constant.

2.4.2 Randomized decision forests

A random forest comprises T decision trees, each including the root, branch, and leaf nodes. Each branch node uses a feature f_θ and a threshold τ for decision-making. To classify pixel x in



(a) (b)

Fig. 5. (Color online) The red X mark indicates the pixel x that is being classified. The green circle indicates the pixel that is offset, as defined in Eq. (9). (a) The two exemplar features produce a significant depth difference response. (b) At the new image location, the response of these same two features is significantly smaller.

image I, Eq. (11) is evaluated starting from the root node, directing the path left or right on the basis of comparisons with τ. In tree t, the classification result is stored as the probability distribution $p_t(c \mid I, x)$ in the leaf node, representing body part labels c. The final prediction is obtained by combining the outputs of all decision trees.

$$P(c \mid I,x) = \frac{1}{T}\sum_{t=1}^{T} P_t(c \mid I,x) \tag{11}$$

2.4.3 Training

To train the trees, distinct sets of randomly synthesized images are used, with 2000 example pixels randomly selected from each image. This ensures a balanced representation of body parts. The training process for each tree follows a uniform algorithm.

(1) Randomly propose a set of segmentation candidates $\varphi_0 = (\theta, \tau)$, where θ represents the feature parameter, and τ is the threshold.

(2) Use each φ_0 to divide the sample set $Q = \{(I, x)\}$ into a left subset and a right subset.

$$Q_l(\varphi_0) = \left\{(I,x) \middle| f_\theta(I,x) < \tau\right\} \tag{12}$$

$$Q_r(\varphi_0) = Q, \quad Q_l(\varphi_0) \tag{13}$$

(3) Calculate the φ_0 that yields the maximum information gain.

$$\varphi^* = \arg\max_{\varphi_0} G(\varphi_0) \tag{14}$$

$$G(\varphi_0) = H(Q) - \sum_{s\in\{1,r\}} \frac{|Q_s(\varphi_0)|}{|Q|} H(Q_s(\varphi_0)) \tag{15}$$

Shannon entropy $H(Q)$ is derived using the normalized histogram, which reflects the distribution of body part labels $l_I(x)$ *for* each $(I, x) \in Q$.

(4) If the maximum gain $G(\varphi^*)$ is sufficiently large and the depth of the tree is below the maximum limit, then regression is performed on the left subset $Q_l(\varphi^*)$ and the right subset $Q_r(\varphi^*)$.

2.4.4 Joint position proposals

Recognizing body parts involves collecting data from individual pixels and combining it to accurately infer the 3D positions of skeletal joints. The method applies a local pattern detection technique that uses mean shift and a weighted Gaussian kernel. We establish the density estimator for each body part as

$$f_c\left(\hat{x}\right) \propto \sum_{i=1}^{N} \omega_{ic} \exp\left(-\| \frac{\hat{x} - \hat{x}_i}{b_c} \|^2\right), \tag{16}$$

where $\hat{x}$ denotes the 3D coordinates, N represents the total number of image pixels, ω_{ic} is the weight assigned to each pixel, and $\hat{x}_i$ refers to the reprojection of the pixel x_i into the 3D space at a given depth $d_I(x_i)$. The parameter b_C stands for the bandwidth associated with each learned part. The weight ω_{ic} considers the probability that the pixel belongs to a specific body part, alongside the pixel's corresponding surface area in the 3D space.

$$\omega_{ic} = P(c \mid I, x_i) \cdot d_I\left(x_i\right) \tag{17}$$

This approach ensures that the density estimate remains invariant to depth, resulting in a small but significant improvement in joint prediction accuracy. Depending on the definition of the body part, the posterior distribution $P(c \mid I, x)$ can integrate information from neighboring or related body parts for consistent computation or processing.

2.5 Camera space

Camera space refers to the 3D coordinate system used by Kinect v2, which is defined as the center of the infrared reception on Kinect v2 as the origin, X increases to the left of the infrared reception, Y increases upwards of the infrared reception, and Z increases in the opposite direction of the infrared reception, all in m, as shown in Fig. 1.

According to Fankhauser *et al.*,[31] there is a 22 mm distance from the IR receiver to the Kinect v2 housing, meaning the starting point of the Z-axis is actually located 22 mm inside the camera housing. Additionally, Wasenmüller and Stricker[32] stated that Kinect v2 provides more accurate data after a 25-min warm-up period.

2.6 Camera space conversion angle

Table 1 summarizes the selected skeletal joints and their corresponding 3D angular features. This sensor-based extraction mechanism filters out redundant skeletal data. It effectively reduces computational complexity and preserves essential kinematic vectors for robust gesture recognition.

The depth sensor acquires 3D spatial coordinates of the skeletal joints. Table 1 defines the specific joint combinations for each angular feature. The system designates the three joints in each row as Point A, Point B, and Point C. Point B serves as the central vertex. These three spatial points form two specific vectors ($\overrightarrow{BA}$ and $\overrightarrow{CB}$). The system calculates the final gesture angles (φ_R, φ_L, θ_R, and θ_L) using the vector inner product formula.

$$\cos\theta = \left(\frac{\overrightarrow{BA} \cdot \overrightarrow{CB}}{\left|\overrightarrow{BA}\right| \cdot \left|\overrightarrow{CB}\right|} \right) \tag{18}$$

After finding cosθ, multiply by arccos to get the inverse cosine of θ_{Rad}.

$$\theta_{Rad} = \arccos\left(\cos\theta\right) \tag{19}$$

Convert radian to θ.

$$\theta = \theta_{Rad} \times \left(\frac{180^\circ}{\pi} \right) \tag{20}$$

2.7 HIWIN RA605-710-GB mechanical arm

The RA605-710-GB robotic arm is a jointed robotic arm with a single motion tandem connection from the base to the end of the arm, a motor with a speed reducer tandem connection, motor power adjustable from high to low, a total of six degrees of freedom, and a maximum load capacity of 7 kg.

Table 1
Definition of extracted 3D angular features for gesture recognition.

Feature parameter	Kinematic description	Extracted skeletal joints
φ_R	Right arm horizontal yaw angle	Right wrist, right shoulder, spine shoulder
φ_L	Left arm horizontal yaw angle	Left wrist, left shoulder, spine shoulder
θ_R	Right arm vertical pitch angle	Right wrist, spine shoulder, spine mid
θ_L	Left arm vertical pitch angle	Left wrist, spine shoulder, spine mid

2.8 Experimental system architecture

The Kinect v2 depth sensor activates when a customer stands within the detection range. When the customer raises a hand to initiate an order, the wrist feature point must elevate higher than 10 cm below the spine mid feature point. The system detects this specific skeletal elevation and recognizes the ordering intention.

The system calculates the real-time gesture coordinates and then evaluates these coordinates against the designated ordering panel ranges. A left-hand gesture targets panels R1 to R4. The system verifies the accurate hand position and identifies the corresponding meal number.

The gesture must remain stable for exactly 1.5 s. The system records angular data every 0.05 s during this duration. The variable n represents each discrete sampling instance. The system collects exactly 30 samples. The system defines a general angular variable V. This variable V represents any of the four gesture angles (φ_R, φ_L, θ_R, or θ_L). The variable V_n represents the specific angle at sampling instance n. The system calculates the final average angle (V_{avg}) to confirm the meal selection. The mathematical averaging formula is as follows.

$$V_{avg} = \frac{1}{30}\sum_{n=1}^{30} V_n \tag{21}$$

The system transmits a TCP/IP command signal to the RA605-710-GB robotic arm. Upon receiving the command, the robotic arm executes the preprogrammed food preparation trajectory. A selection of meal number 1 triggers preparation path 1. This overall system workflow is illustrated in Fig. 6.

The system architecture utilizes the Kinect v2 depth sensor, which integrates an RGB camera and a ToF module. The ToF module determines depth by measuring the light travel time. The sensor continuously emits CW-iToF–modulated infrared signals. It detects the offset phase of the returning light to calculate spatial depth.[32,33]

The depth sensor processes this data to perform per-pixel body part classification. The tracking algorithm locates body joints by finding the probability-density center of gravity. This algorithm maps these joints to a human skeleton model using temporal continuity and skeletal training data.[34] The skeleton tracking module extracts all 3D joint parameters in real time.

This tracking module transfers the 3D coordinate data to a LabVIEW software program. The LabVIEW program directly converts these coordinates into the required gesture feature angles. These calculated angles determine the final meal selection. A TCP/IP network transmits the designated order number to the robotic arm. The robotic arm executes the programmed movement path for automated food preparation. This complete system architecture is illustrated in Fig. 7.

2.9 Principle of meal angle calculation

The Kinect v2 depth sensor utilizes skeleton tracking technology to calculate critical spatial distances. First, the system measures the direct distance between the tester's shoulder and the

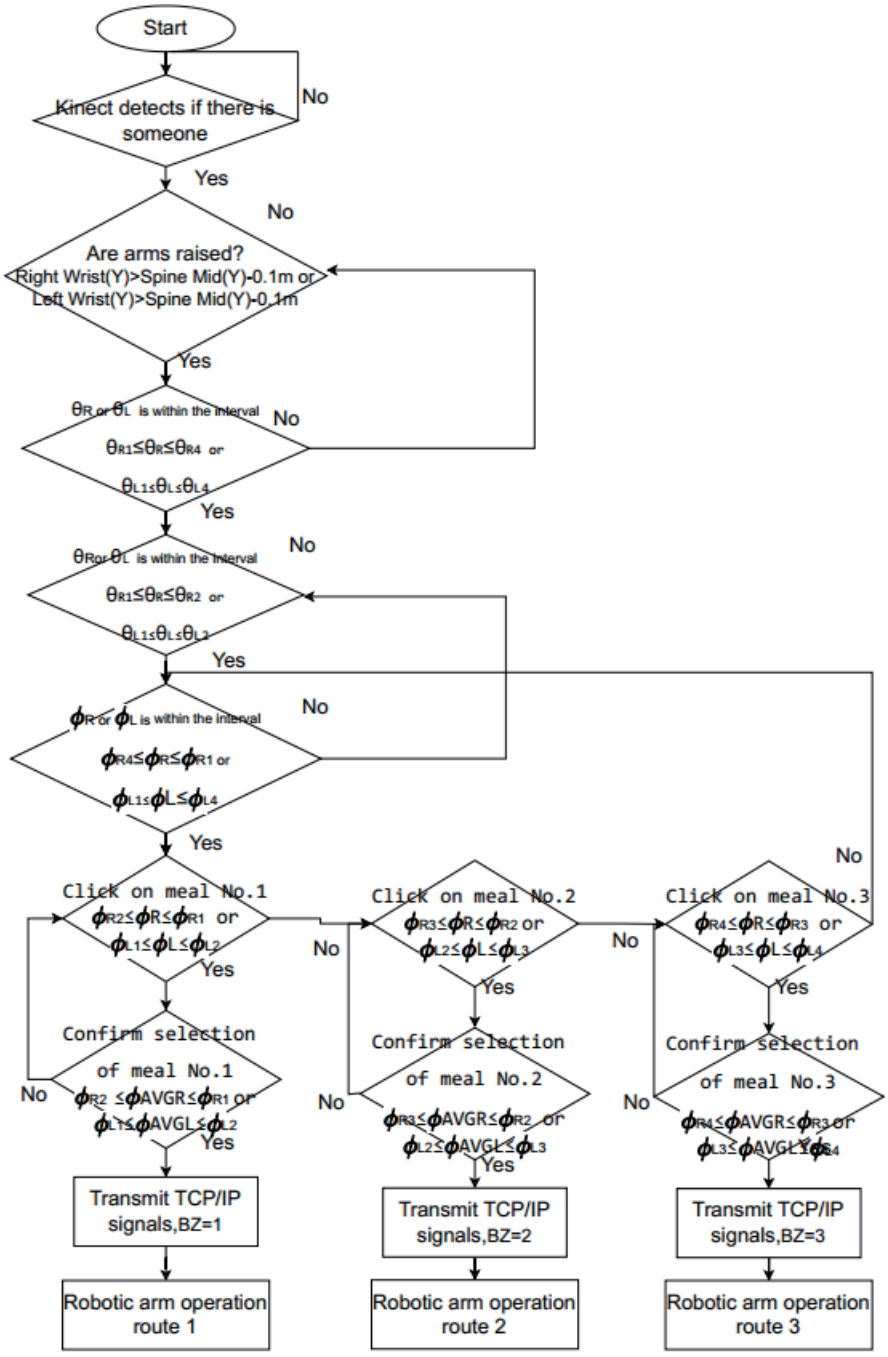


Fig. 6. System architecture diagram of the gesture recognition ordering system.

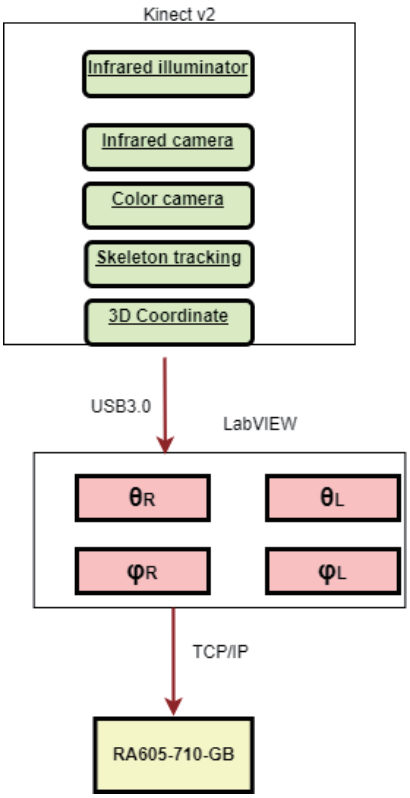


Fig. 7. (Color online) System architecture of the interface parameters.

sensor. Then the system calculates the vertical height difference between the tester's shoulder and foot. The system also calculates the lateral distance from the body midline to the shoulder. The distance between the wall and the sensor is permanently fixed. The system indirectly extrapolates the distance between the tester and the wall using the shoulder-to-sensor measurements.

The system determines three spatial metrics measurements using specific skeletal features. The calculation of the shoulder-to-sensor distance relies on the Right Shoulder or Left Shoulder feature points. The system subtracts the Right Foot feature point from the Right Shoulder feature point to derive the height difference. Then the lateral shoulder width is calculated using the horizontal distance between the shoulder and the spine shoulder feature point.

We examine a specific tester scenario to demonstrate this angular calculation. The tester is pointing to the beverage panel region 1. Table 2 summarizes the system's spatial measurements for this specific tester.

The system first calculates the horizontal yaw angle (φ_L). Figure 8 illustrates the conversion of 3D spatial points to the 2D plane for this calculation. The system defines three geometric points (A, B, and C) and calculates the lengths of line segments $\overline{AB}$ and $\overline{BC}$.

Table 2
Spatial measurement data for the angle calculation.

Parameter	Value (cm)
Tester height	185
Z-axis distance (Kinect v2 to wall)	179
Z-axis distance (Shoulder Left to Kinect v2)	150
Total *Z*-axis distance (Shoulder Left to wall)	329
Measured shoulder width	19
Panel 1 horizontal distance (*X*-axis)	121.5
Panel 1 vertical height (*Y*-axis)	245
Spine mid height (*Y*-axis)	145

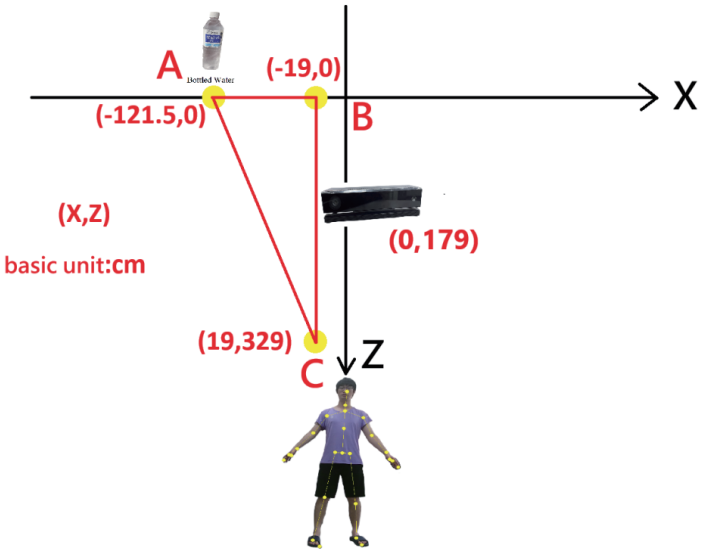


Fig. 8. (Color online) Calculation of 3D to 2D plane conversion φ_L by ordering system.

$$\overline{AB} = \sqrt{(121.5-19)^2 + (0-0)^2} = 102.5 \text{ cm} \tag{22}$$

$$\overline{BC} = \sqrt{(-19-(-19))^2 + (0-329)^2} = 329 \text{ cm} \tag{23}$$

The system calculates the intermediate radian value (*Rad*1) using the arctangent function, based on the actual measurements as shown in Fig. 9, and converts this radian value to the base angle θ_1.

$$Rad1 = \arctan \times (102.5 / 329) \cong 0.3 \tag{24}$$

$$\theta_1 = Rad1 \times (180° / \pi) \cong 17° \tag{25}$$

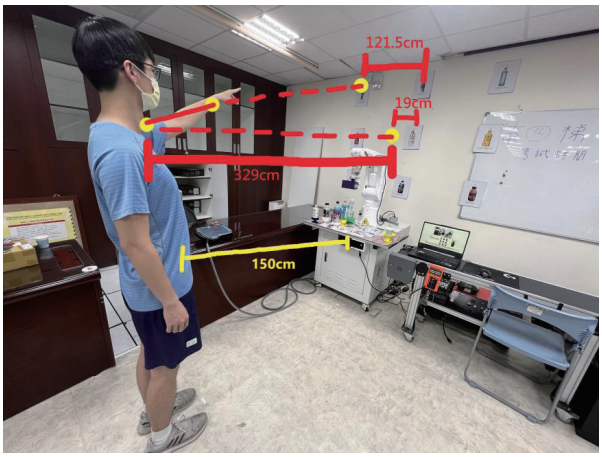


Fig. 9. (Color online) Actual ordering measurement by ordering system.

The ideal straight-forward pointing angle is defined as 90°. The system calculates the final φ_L angle by subtracting θ_1 from 90°.

$$\varphi_L = 90° - 17° = 73° \tag{26}$$

When the tester is pointing to the center point of order number 1, φ_L is 73°.

The system then calculates the vertical pitch angle (θ_L). Figure 10 illustrates this specific 2D plane conversion. The system defines three new geometric points (D, E, and F). The system calculates the effective vertical height difference for point E (245 cm − 145 cm = 100 cm). The system calculates the lengths of line segments $\overline{DF}$ and $\overline{EF}$.

$$\overline{DF} = \sqrt{(0-0)^2 + (329-0)^2} = 329 \text{ cm} \tag{27}$$

$$\overline{EF} = \sqrt{(100-0)^2 + (0-0)^2} = 100 \text{ cm} \tag{28}$$

The system calculates the intermediate radian value ($Rad2$) using the arctangent function. Then it converts this radian value to the base angle $\boldsymbol{\theta}_2$.

$$Rad2 = \arctan \times (100 / 329) \cong 0.3 \tag{29}$$

$$\theta_2 = 0.3 \times (180° / \pi) \cong 17° \tag{30}$$

The system applies the 90° ideal forward angle baseline to calculate the final $\boldsymbol{\theta}_L$ angle by subtracting $\boldsymbol{\theta}_2$ from 90°.

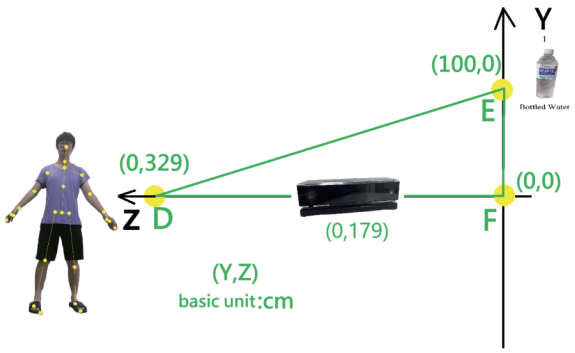


Fig. 10. (Color online) Calculation of 3D to 2D plane conversion of θ_L by the ordering system.

$$\varphi_L = 90° - 17° = 73° \tag{31}$$

When the tester is pointing to the center point of order number 1, θ_L is 73°, as shown in the actual measurements in Fig. 11.

Initial gesture measurements exhibited systematic errors. We calibrated the straight-ahead angle under ideal conditions. The experiment included 10 measurements per hand for a total of 20 samples. A laser pointer projected the shoulder positions onto the wall to ensure measurement accuracy. The Kinect v2 recorded an average angle of 77.1 ± 4.3°. Figure 12 illustrates these measurement results. The system adjusted the ideal reference angle from 90 to 77.1° on the basis of these results.

$$\varphi_L = 77.1° - \angle C \tag{32}$$

The ordering range is defined as extending 30 cm to the left and right and 17.5 cm above and below the panel, which has the dimensions of 30 cm in length and 21 cm in width, as shown in Fig. 13. Taking a la carte number 5 as an example, the measurer's height was 185 cm and the distance from Kinect v2 was 150 cm. According to the formulas, the range was 73.2 to 87.4° for both φ_L and φ_R, while θ_L ranged from 78.4 to 89.6°, and θ_R from 78.4 to 89.6°. Table 3 provides further details on the angle calculation results for all order codes.

3. Results

The purpose of this experiment is to verify whether the Kinect v2 sensor can be used as an accurate distance measuring instrument, and whether its converted angle is correct, and ultimately to evaluate whether this ordering system can correctly judge the customer's gestures and select the correct meal.

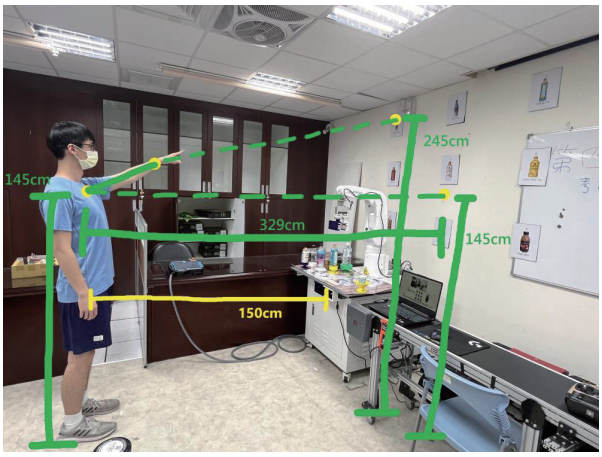


Fig. 11. (Color online) Actual geometry measurement of the ordering experiment.

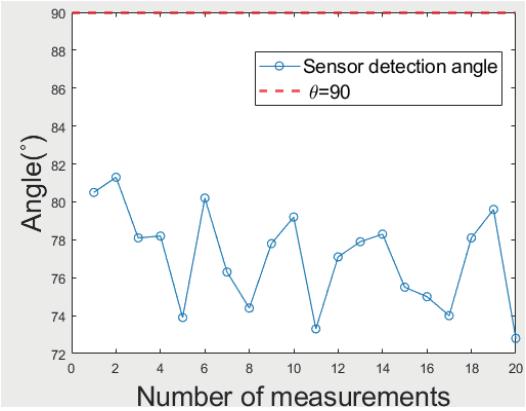


Fig. 12. (Color online) Ideal angle of the finger pointing forward and the error of the Kinect v2's detected angle.

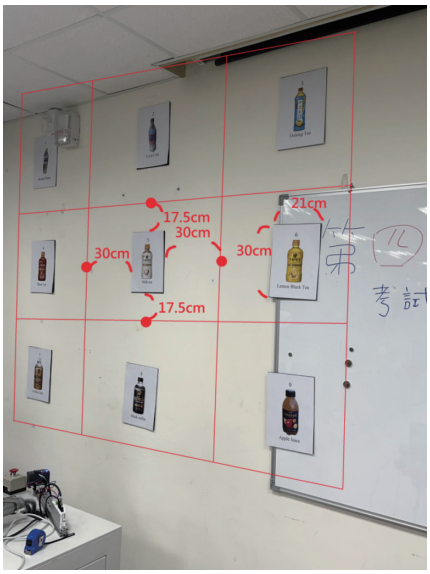


Fig. 13. (Color online) Definition of ordering range in ordering panel.

Table 3
Corresponding angles for ordering codes.

NO.	φ_L	φ_R	θ_L	θ_R
1	59.8–73.2°	87.4–100.2°	68.1–78.4°	68.1–78.4°
2	73.2–87.4°	73.2–87.4°	68.1–78.4°	68.1–78.4°
3	87.4–100.2°	59.8–73.2°	68.1–78.4°	68.1–78.4°
4	59.8–73.2°	87.4–100.2°	78.4–89.6°	78.4–89.6°
5	73.2–87.4°	73.2–87.4°	78.4–89.6°	78.4–89.6°
6	87.4–100.2°	59.8–73.2°	78.4–89.6°	78.4–89.6°
7	59.8–73.2°	87.4–100.2°	89.6–100.8°	89.6–100.8°
8	73.2–87.4°	73.2–87.4°	89.6–100.8°	89.6–100.8°
9	87.4–100.2°	59.8–73.2°	89.6–100.8°	89.6–100.8°

3.1 Verify the accuracy of the Kinect v2 sensor 3D coordinate system

The Right Wrist point was chosen for the distance measurement. First, a physical reference marker was established, and a tape measure was used to measure the 3D coordinate system. The Right Wrist point was then overlaid on the mark and tested. The Kinect v2 sensor outputs 3D coordinate system data for the Right Wrist point in meters. This experiment ensured that the Kinect v2 sensor was warmed up for more than 25 min prior to testing to obtain more accurate data.

The actual value measured using a tape measure was $X = 0.000$ m, which was at a point 5.1 cm to the right of the RGB camera as the infrared receiving position.[35] The measured values are $Y = 0.159$ m and $Z = 1.066$ m. Considering the internal depth offset of 22 mm inside the camera housing, the actual starting point of the Z-axis should be $Z = 1.066$ m + 0.022 m = 1.088 m, which is indicated by a blue diamond.[31]

Fifteen 3D coordinate system samples were collected using the Kinect v2 sensor and are represented by red dots in Fig. 14. The measured mean values were $X = -0.004 \pm 0.003$ m, $Y = 0.159 \pm 0.002$ m, and $Z = 1.090 \pm 0.005$ m.

3.2 Verifying the angular accuracy of Kinect v2 sensors

The experimental measurement of angles is divided into two groups. The first set of angles is the angle formed by the points right wrist, right shoulder, and spine shoulder. For the measurement, a string was used to connect these three points to form the angle. The angle is then measured with a protractor and converted to a supplementary angle to determine the φ_R angle. The angle data output by the Kinect v2 sensor was then recorded. A total of 20 angle samples were collected for this experiment.

The angle measured using a protractor is 84°, as shown in Fig. 15, and the converted supplementary angle is 180° − 84° = 96°. The average value of φ_R of 20 angles measured using the Kinect v2 sensor is 95.4 ± 0.6°, as shown in Fig. 16.

The second set of angles measured experimentally is the clip angle formed by the points left wrist, left shoulder, and spine shoulder. For this measurement, we also used a piece of string to connect these three points and form the pinch angle. Then we measured the pinch angle with a protractor and converted it to a supplementary angle to determine the φ_L angle. The angle data

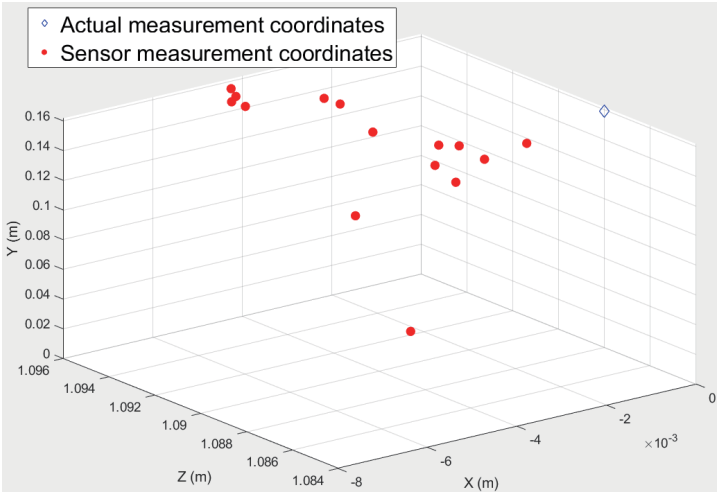


Fig. 14. (Color online) Scatterplot of Kinect v2 measured 3D coordinates.

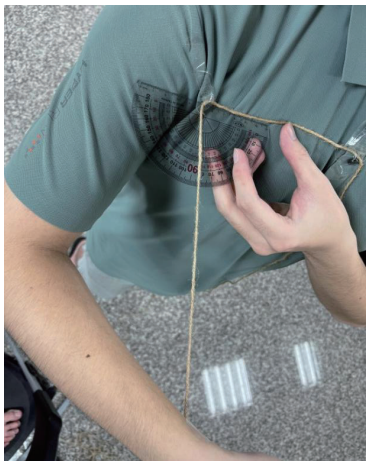


Fig. 15. (Color online) Angle φ_R measured using a protractor is 84°.

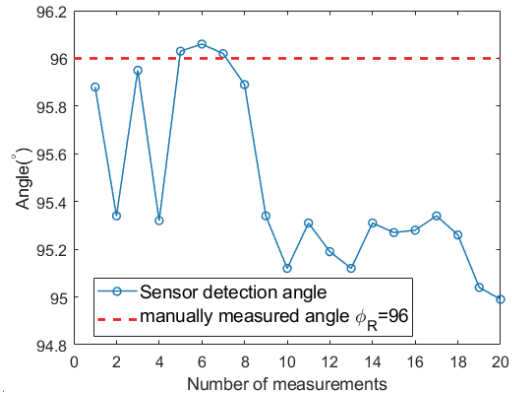


Fig. 16. (Color online) Folding chart of Kinect v2-measured φ_R.

output by the Kinect v2 sensor was then recorded. A total of 20 angle samples were collected for this experiment.

The angle measured using a protractor is 80°, as shown in Fig. 17, and the converted supplementary angle is 180° − 80° = 100°, and the average value of φ_L of 20 measured angles using the Kinect v2 sensor is 101.2 ± 1.6°, as shown in Fig. 18.

3.3 Verifying the reliability of the angles calculated by the ordering system

To test the reliability of the angular range calculated by the ordering system, measurements were taken at a distance of 1.498 ± 0.027 m (referred to as distance A) and 2.510 ± 0.010 m (distance B) from the Kinect v2 sensor. A laser pointer was utilized to confirm that the participant pointed directly at the center of ordering panel 5. The experiment involved 10 trials of measuring angles θ_2, θ_3, φ_{R2}, φ_{R3}, φ_{L2}, and φ_{L3}. Table 4 shows the angle ranges derived from

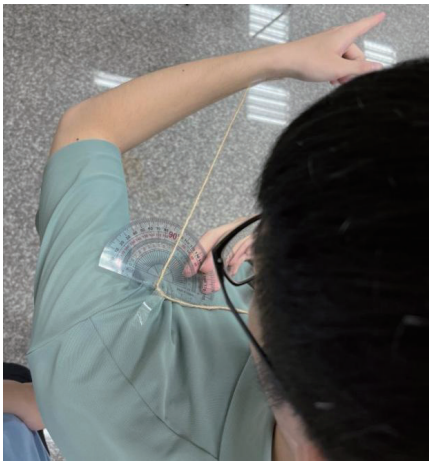


Fig. 17. (Color online) Angle φ_L measured using a protractor is 80°.

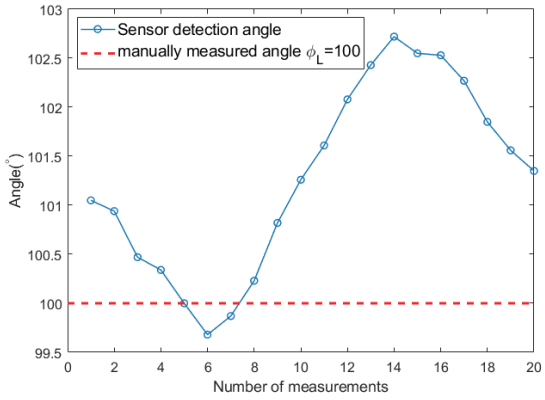


Fig. 18. (Color online) Folding chart of Kinect v2-measured angle φ_L.

Table 4
Meal angle range to obtain measurement results.

		θ_2	θ_3	φ_{R2}	φ_{R3}	φ_{L2}	φ_{L3}
Distant A	mean value	77.09°	88.14°	87.29°	73.30°	73.03°	87.16°
	maximum value	77.42°	88.58°	87.44°	73.45°	73.07°	87.25°
	minimum value	76.33°	87.35°	87.16°	72.16°	72.98°	87.06°
Distant B	mean value	80.53°	89.11°	84.85°	74.11°	74.02°	84.82°
	maximum value	80.58°	89.15°	84.88°	74.13°	74.05°	84.85°
	minimum value	80.50°	89.06°	84.83°	74.10°	74.01°	84.81°

the formula, assuming a height of 185 cm and a camera distance of 1.5 m. The calculated angles are θ_2= 78.4°, θ_3 = 89.6°, φ_{R2} = 87.4°, φ_{R3} = 73.2°, φ_{L2} = 73.2°, and φ_{L3} = 87.4°.

The meal angle measurement outcomes are illustrated in blue in the box plot in Fig. 19, with corresponding values listed in Table 4. A comparison reveals that the difference between the calculated values and the experimental results is minor, confirming that the original formula provides a suitable foundation for setting the angular range limit for meal orders.

3.4 Verify the reliability of the ordering system

To verify the reliability of the ordering function of the system, a test was conducted with eight males, numbered 1 to 8. Their physical data are shown in Table 5. The experiment focused on three aspects: left-handed and right-handed ordering, ordering at a distance, and ordering by different people. The test consisted of nine types of meal, numbered 1 to 9. The test distances were set in two ranges, 1.5 ± 0.05 m for distance C and 2.5 ± 0.05 m for distance D, as shown in Figs. 20(a) and 20(b).

The experiment was conducted as follows. The tester selected meal numbers 1 to 9 in turn, pointing to the position of each meal number, as shown in Fig. 21. The Kinect v2 sensor calculated the corresponding angles on the basis of the tester's gesture characteristics and provided feedback through light signals. The system is set up with nine light signals, each corresponding to one of the nine meal codes.

When the tester's gesture enters the angle range of the corresponding number, the light will light up green to indicate that it has entered the angle range of the meal number. If the gesture remains within the angular range for 1.5 s, the light will turn red, indicating that the tester has confirmed the selection of the meal, as shown in Fig. 22. When the light turns red, the position of the light is recorded. If the location of the light is the same as the number of the meal that the tester wants to order, the ordering is successful; otherwise, the ordering fails.

In addition, if the tester takes more than 5 s to enter the ordering angle range, it is also regarded as a failure. Each tester was required to perform the test from the same distance with the right and left hands, and order each meal number once using each hand, for a total of 36 tests. The test results are shown in Table 6.

A total of 288 ordering tests were conducted to evaluate the system. Experimental results showed that the success rate did not decrease significantly as the distance from the Kinect v2 sensor increased. Distance C achieved a success rate of 96.5%, while that of distance D reached

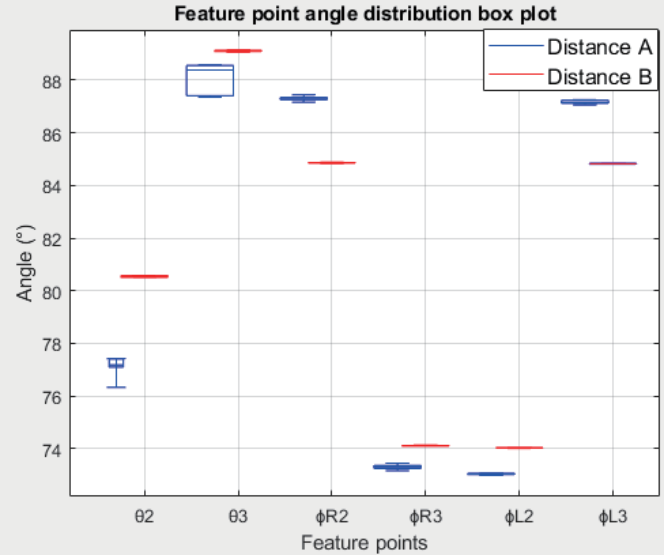


Fig. 19. (Color online) Box plot showing the change in the angle of ordering at distances A and B.

Table 5
Testers' physical data.

Tester No.	Age	Height (cm)	Weight (kg)
1	21	185	74
2	26	187	104
3	24	184	98
4	24	179	82
5	24	174	90
6	21	170	65
7	24	169	45
8	21	178	61

(a) (b)

Fig. 20. (Color online) Ordering diagram at tester (a) distance C and (b) distance D .

Fig. 21. (Color online) Experimental setup of the ordering screen.

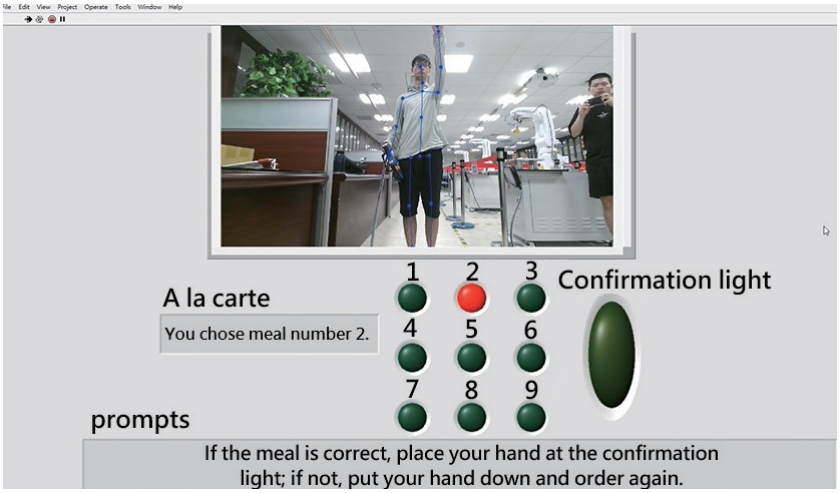


Fig. 22. (Color online) Software operation of the ordering interface.

Table 6
Data captured by the ordering system.

Tester No.	Distance C, right hand success/Total test (times)	Distance C, left hand success/Total test (times)	Distance D, right hand success/Total test (times)	Distance D, left hand success/Total test (times)
1	9/9	9/9	6/9	8/9
2	9/9	7/9	8/9	9/9
3	9/9	9/9	9/9	9/9
4	8/9	9/9	9/9	9/9
5	9/9	9/9	9/9	9/9
6	9/9	9/9	9/9	9/9
7	9/9	7/9	7/9	9/9
8	9/9	9/9	9/9	9/9

95.1%. Participants were also divided into three height groups for statistical validation. The group of ≤ 175 cm achieved a 96.3% success rate. The 175–185 cm group attained a 99.0% success rate. The group of ≥ 185 cm recorded a 90.3% success rate. These findings confirm that the adaptive geometric model maintains high accuracy across different body heights. Individual success rates ranged from 88.8 to 100% for the 36 trials per person. This variation is attributed to differences in individual biomechanical habits. Overall, the system demonstrates high robustness across varying distances and tester statures.

4. Discussion

The primary objective of this study was to establish a highly reliable gesture recognition framework utilizing a Kinect v2 depth sensor for contactless interaction. To rigorously evaluate the system's robustness, experiments were conducted across distinct spatial parameters, for a total of 162 independent tests for each distance category. The system achieved a 96.5% success rate at a close proximity of 1.5 ± 0.05 m. Furthermore, it maintained a 95.1% success rate at an extended distance of 3 ± 0.05 m. The fundamental challenge addressed in this research was maintaining consistent recognition accuracy despite variations in tester height and sensor-to-tester distance, as these factors directly impact the effective ordering range and detection reliability.

4.1 Adaptive geometric calibration and spatial robustness

To overcome distance-induced scaling errors, the proposed methodology employs a 3D-to-2D geometric projection technique. The Kinect v2 depth sensor continuously captures spatial metrics, including the tester's height and precise distance from the camera. By applying trigonometric principles—specifically, the Pythagorean theorem—and inverse kinematic deduction, the system dynamically calibrates the required angular boundaries for the operational range.

The minimal 1.4% discrepancy in success rates (96.5% vs 95.1%) between the near and far test groups quantitatively confirms that this dynamic angular calibration algorithm effectively compensates for spatial variations. Experimental results demonstrate that this approach achieves stable gesture recognition under various conditions, successfully maintaining accuracy during both left-handed and right-handed operations across different testers.

4.2 System limitations and future directions

Despite the demonstrated high reliability, inherent hardware limitations restrict the current system's operational envelope. The Kinect v2 sensor's depth sensing resolution exponentially degrades, ultimately limiting accurate gesture recognition beyond a maximum range of 4.2 m. Future research and development will focus on deploying multiple Kinect v2 sensors to execute trilateral spatial measurements. This multisensor integration is anticipated to mitigate current

hardware constraints, significantly improving overall system stability and extending coordinate accuracy over a broader operational range.

5. Conclusions

We successfully developed a contactless automated ordering system driven by Kinect v2 depth sensing and iToF technology. The system utilizes infrared phase-shift detection to capture 3D skeleton tracking data. We proposed an adaptive geometric model to transform these raw sensing signals into precise gesture interaction regions. This sensor-driven approach effectively compensates for variations in tester height and standing distance.

To ensure the reliability of the spatial data processing, we rigorously verified the Kinect v2 3D coordinate system. Experimental measurements of 15 spatial coordinates demonstrated high precision. The average values were $X = -0.004 \pm 0.003$ m, $Y = 0.159 \pm 0.002$ m, and $Z = 1.090 \pm 0.005$ m. These results align closely with the actual baseline positions of $X = 0.000$ m, $Y = 0.159$ m, and $Z = 1.088$ m. Validation tests confirmed that the depth sensor accurately captures the kinematic vectors required for gesture recognition.

System evaluations comprising 288 independent tests confirmed the robustness of the adaptive calibration. The system achieved a 96.5% success rate at 1.5 ± 0.05 m and maintained a 95.1% success rate at 3 ± 0.05 m. The algorithm recorded a 96.5% success rate for left-handed operations and 95.1% for right-handed operations. Individual testing yielded success rates between 88.8 and 100%. These findings indicate that while the system compensates for spatial variations, minor discrepancies persist due to individual biomechanical habits.

The system performance remains constrained by the physical limitations of the depth sensor. Kinect v2 exhibits measurement instability for subjects shorter than 1 m. Depth resolution also degrades significantly beyond 4.2 m. To overcome these hardware sensing constraints, future research will focus on integrating multiple Kinect v2 sensors for trilateral measurements. As indicated by Yang *et al.*,[35] trilateral integration can reduce coordinate errors from 51.22 to 25.61 mm. This multisensor integration approach will further extend the reliable operational range and environmental adaptability of contactless sensing systems.

Acknowledgments

This research work is a basic study on the integration of robotic arms controlled via a remote instructor interface and is supported by the National Science and Technology Council (NSTC) under the Special Project Grant (NSTC 112-2221-E-167-028 and NSTC 114-2637-E-167-010), which is gratefully acknowledged.

References

1 Y. Lee, Y. Fanjiang, C. Hung, and H. Huang: Proc. 2019 IEEE Int. Conf. Consumer Electronics - Taiwan (IEEE, 2019) 1–2. https://doi.org/10.1109/ICCE-TW46550.2019.8991692

2 S. Goto, T. Sugi, and M. Nakamura: Proc. 2006 SICE-ICASE Int. Joint Conf. (IEEE, 2006) 227–232. https://doi.org/10.1109/SIC E.2006.315612

3 S. Chen, H. Yu, and S. Yang: Proc. 2018 Int. Symp. Computer, Consumer and Control (IEEE, 2018) 318–321. https://doi.org/10.1109/IS3C.2018.00087
4 F. Shen, F. Tsai, H. Lin, and H. Zheng: Proc. 2015 IEEE Int. Conf. Consumer Electronics (IEEE, 2015) 356–357. https://doi.org/10.1109/ICCE-TW.2015.7216941
5 M. Kavitha, J. Venkatesh, S. K. Salma, and S. R. Moosa: Proc. 2022 7th Int. Conf. Communication and Electronics Systems (IEEE, 2022) 449–454. https://doi.org/10.1109/ICCES54183.2022.9835976
6 A. Sajnani and N. Patel: Proc. Asian Journal for Convergence in Technology (AJCT, 2020) 56–62. http://asianssr.org/index.php/ajct/article/view/915
7 Z. Wang, G. Liu, and G. Tian: Proc. 2017 Chinese Automation Congress (IEEE, 2017) 4640–4644. https://doi.org/10.1109/CAC.2017.8243598
8 J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake: Proc. Conf. Computer Vision and Pattern Recognition (IEEE, 2011) 1297–1304. https://doi.org/10.1109/CVPR.2011.5995316
9 M. M. F. M. Fareed, Q. I. Akram, S. B. A. Anees, and A. H. Fakih: Proc. 2015 Fifth Int. Conf. Communication Systems and Network Technologies (IEEE, 2015) 1210–1215. https://doi.org/10.1109/CSNT.2015.86
10 C. Li, C. Yang, P. Liang, A. Cangelosi, and J. Wan: Proc. 2016 Int. Conf. Advanced Robotics and Mechatronics (IEEE, 2016) 133–138. https://doi.org/10.1109/ICARM.2016.7606908
11 J. R. Balbin, J. Y. P. Damilig, and M. G. Pelayo: Proc. 2021 IEEE Int. Conf. Automatic Control and Intelligent Systems (IEEE, 2021) 163–168. https://doi.org/10.1109/I2CACIS52118.2021.9495880
12 M. Limin and Z. Peiyi: Proc. 2017 IEEE Int. Conf. Intelligent Techniques in Control, Optimization and Signal Processing (IEEE, 2017) 1–7. https://doi.org/10.1109/ITCOSP.2017.8303077
13 J. Wang, Z. Liu, Y. Wu, and J. Yuan: IEEE Trans. Pattern Anal. Mach. Intell. **36** (2014) 914. https://doi.org/10.1109/TPAMI.2013.198
14 H. Alabbasi, A. Gradinaru, F. Moldoveanu, and A. Moldoveanu: Proc. 2015 E-Health and Bioengineering Conf. (IEEE, 2015) 1–4. https://doi.org/10.1109/EHB.2015.7391465
15 Y. A. Hutabarat, A. Rizal, H. Mukthar, and S. Ziani: Proc. 2023 3rd Int. Conf. Intelligent Cybernetics Technology & Applications (IEEE, 2023) 12–16. https://doi.org/10.1109/ICICyTA60173.2023.10428906
16 S. G. Heng, R. Samad, M. Mustafa, N. R. H. Abdullah, and D. Pebrianti: Proc. 2019 IEEE 9th Int. Conf. System Engineering and Technology (IEEE, 2019) 17–22. https://doi.org/10.1109/ICSEngT.2019.8906419
17 Q. Weiming, Z. Xiaomei, H. Jiwei, and L. Ping: Proc. 2020 5th Int. Conf. Mechanical, Control and Computer Engineering (IEEE, 2020) 387–391. https://doi.org/10.1109/ICMCCE51767.2020.00092
18 Y. Bouteraa and I. B. Abdallah: Proc. 2019 Int. Conf. Signal, Control and Communication (IEEE, 2019) 337–343. https://doi.org/10.1109/SCC47175.2019.9116099
19 N. Klievtsova and M. Fuschlberger: it - Inf. Technol. **65** (2023) 91. https://doi.org/10.1515/itit-2023-0006
20 R. R., R. R. P., S. C., and S. K. B.: Proc. 2023 2nd Int. Conf. Advancements in Electrical, Electronics, Communication, Computing and Automation (IEEE, 2023) 1–5. https://doi.org/10.1109/ICAECA56562.2023.10199301
21 M. M. Mepani, K. B. Gala, T. A. Mishra, K. S. Bhole, J. Gholave, and S. Daingade: Mater. Today Proc. **68** (2022) 1930. https://doi.org/10.1016/j.matpr.2022.08.140
22 Y. Elhousry, O. Yasser, S. M. Ismail, and H. H. Ammar: Proc. 2024 Int. Conf. Machine Intelligence and Smart Innovation (IEEE, 2024) 210–213. https://doi.org/10.1109/ICMISI61517.2024.10580859
23 J. M. Garcia-Haro: Electron. **10** (2021) 47. https://doi.org/10.3390/electronics10010047
24 C. Nuzzi, S. Pasinetti, M. Lancini, F. Docchio, and G. Sansoni: IEEE Instrum. Meas. Mag. **22** (2019) 44. https://doi.org/10.1109/MIM.2019.8674634
25 J. H. Kim, G. S. Hong, B. G. Kim, and D. P. Dogra: Displays **55** (2018) 38. https://doi.org/10.1016/j.displa.2018.08.001
26 J. Sell and P. O'Connor: IEEE Micro **34** (2014) 44. https://doi.org/10.1109/MM.2014.9
27 A. Corti, S. Giancola, G. Mainetti, and R. Sala: Rob. Auton. Syst. **75** (2016) 584. https://doi.org/10.1016/j.robot.2015.09.024
28 T. Watanabe and Y. Saito: Proc. 2013 Seventh Int. Conf. on Sensing Technology (ICST, 2013) 434–439. https://doi.org/10.1109/ICSensT.2013.6727690
29 J. Jiao, L. Yuan, W. Tang, Z. Deng, and Q. Wu: ISPRS Int. J. Geo-Inf. **6** (2017) 349. https://doi.org/10.3390/ijgi6110349
30 D. Herrera C., J. Kannala, and J. Heikkilä: IEEE Trans. Pattern Anal. Mach. Intell. **34** (2012) 2058. https://doi.org/10.1109/TPAMI.2012.125
31 P. Fankhauser, M. Bloesch, D. Rodriguez, R. Kaestner, M. Hutter, and R. Siegwart: Proc. 2015 Int. Conf. Advanced Robotics (IEEE, 2015) 388–394. https://doi.org/10.1109/ICAR.2015.7251485

32 O. Wasenmüller and D. Stricker: Proc. Asian Conf. Computer Vision Workshop (Springer Link, 2017) 34–45. https://doi.org/10.1007/978-3-319-54427-4_3
33 M. J. Rosenstrauch, T. J. Pannen, and J. Krüger: Procedia CIRP **76** (2018) 183. https://doi.org/10.1016/j.procir.2018.01.026
34 Z. Zhang: IEEE MultiMedia **19** (2012) 4. https://doi.org/10.1109/MMUL.2012.24
35 L. Yang, L. Zhang, H. Dong, A. Alelaiwi, and A. E. Saddik: IEEE Sens. J. **15** (2015) 4275. https://doi.org/10.1109/JSEN.2015.2416651

About the Authors

Bing-Yan Chen received his B.S. degree from National Chin-Yi University of Technology, Taiwan, in 2025. His research interests are in automation integration, robotics, machine stereo vision, and sensors. (3B013208@gm.student.ncut.edu.tw)

Cheng-Yu Peng received his Ph.D. degree from the Graduate Institute of Mechanical and Electrical Engineering, National Taipei University of Technology, Taiwan, in 2007. He was a senior researcher and department manager at Green Energy and Environment Institute, Industrial Technology Research Institute, from 2007 to 2017. Since 2017, he has been a faculty member in the Department of Electronic Engineering, National Chin-Yi University of Technology. His current research interests primarily involve smart supervisory control, intelligent robotics, automation and mechatronics, power engineering, solar energy, and engineering applications. (peng@ncut.edu.tw)