Robotica (2025), **43**, pp. 4088–4115
doi:10.1017/S0263574725102737

CAMBRIDGE UNIVERSITY PRESS

RESEARCH ARTICLE

Real-time mouth posture estimation for meal-assisting robots

Yuhe Fan[1], Lixun Zhang[1], Canxing Zheng[2], Zhenhan Wang[1], Zekun Yang[1], Feng Xue[1], Huaiyu Che[1] and Xingyuan Wang[1]

[1]College of Mechanical and Electrical Engineering, Harbin Engineering University, Building No.61, Nantong Street No. 145, Harbin, 150001, China
[2]Department of Anorectal Surgery, Weifang People's Hospital, Weifangy, Shandong, China
Corresponding author: Lixun Zhang; Email: zhanglixun@hrbeu.edu.cn

Received: 3 March 2025; **Revised:** 25 September 2025; **Accepted:** 27 September 2025;
First published online: 24 October 2025

Keywords: human–robot interaction; safety; fitting algorithms; posture estimation; meal-assisting robotics

Abstract

In the fields of meal-assisting robotics and human–robot interaction (HRI), real-time and accurate mouth pose estimation is critical for ensuring interaction safety and improving user experience. The complexity arises from the diverse opening degrees of mouths, variations in orientation, and external factors such as lighting conditions and occlusions, which pose significant challenges for real-time and accurate posture estimation of mouths. In response to the above-mentioned issues, this paper proposes a novel method for point cloud fitting and posture estimation of mouth opening degrees (FP-MODs). The proposed method leverages both RGB and depth images captured from a single viewpoint, integrating geometric modeling with advanced point cloud processing techniques to achieve robust and accurate mouth posture estimation. The innovation of this work lies in the hypothesis that different states of mouth openings can be effectively described by distinct geometric shapes: closed mouths are modeled by spatial quadratic surfaces, half-open mouths by spatial ellipses, and fully open mouths by spatial circles. Then, based on these hypotheses, we developed algorithms for fitting geometric models to point clouds obtained from mouth regions, respectively. Specifically, for the closed mouth state, we employ an algorithm based on least squares optimization to fit a spatial quadratic surface to the point cloud data. For the half-open or fully open mouth states, we combine inverse projection methods with least squares fitting to model the contour as a spatial ellipse and circle, respectively. Finally, to evaluate the effectiveness of the proposed FP-MODs method, extensive actual experiments were conducted under varying conditions, including different orientations and various types of mouths. The results demonstrate that the proposed FP-MODs method achieves high accuracy and robustness. This study can provide a theoretical foundation and technical support for improving HRI and food delivery safety in the field of robotics.

1. Introduction

Food constitutes a fundamental necessity for human survival and health maintenance, supplying essential nutrients, energy, and functional components that support immune responses and tissue integrity. However, the number of individuals with upper-limb impairments has been increasing annually due to escalating life pressures, frequent traffic accidents, and inherent functional deficiencies in body tissues and organs. To improve the dietary quality of such populations and reduce the labor and energy expenditure of medical staff or family members during assisted feeding, numerous researchers have started developing feeding assistance devices, thus driving growing attention to the research of meal-assisting robot technology [1–3]. Meal-assisting robots, through autonomous perception and intelligent decision-making technologies, are dedicated to providing fine-grained food picking and safe delivery services for individuals with upper-limb dysfunctions. According to references [4–16] and long-term research experience of teams, intelligent meal-assisting robots mainly consist of a meal mechanics module, mechanical structure module, servo control module, robot vision module, robot force sensing

© The Author(s), 2025. Published by Cambridge University Press.

https://doi.org/10.1017/S0263574725102737 Published online by Cambridge University Press

module, and motion planning and decision-making module. First, the robot identifies and localizes the types, positions, and postures of foods and mouths through the vision module. Then, based on the food types recognized by the vision module, it calls the rheological parameter database of the food mechanics module, calculating the desired gripping force and scooping wrist force using parameters such as viscoelasticity, friction coefficient, dynamic viscosity, and damping coefficient. Next, the force sensing module assists the end-effector in picking solid food or scooping liquid food. Subsequently, the motion trajectories for food picking and delivery are planned based on the positions and postures of foods and mouths obtained by the vision module. Finally, by integrating robot vision perception, force sensing, and kinematic signals, the servo control and mechanical structure modules are guided to accomplish food picking and delivery tasks. This work focuses on the research of mouth point cloud fitting and posture estimation for meal-assisting robots.

Estimating mouth posture in real time and with high accuracy presents significant technical challenges. A major factor contributing to this difficulty is the considerable variation in mouth morphology across individuals. Different users may possess distinct mouth structures and sizes, and even within the same individual, morphological changes can occur under varying articulatory and expressive conditions [17–19]. Second, the complexity of posture estimation is further increased by the multi-degree-of-freedom nature of mouth motions, which involve rotation and movement along multiple axes, thereby making accurate estimation more challenging. In addition, variations in external lighting conditions and the presence of occluding objects (such as glasses and tablespoons) cause some interference in the acquisition of facial images, which in turn affects the accuracy of mouth pose estimation [20–22]. Although numerous challenges exist, achieving real-time and accurate posture estimation of mouth opening degrees (closed, half-open, and open states) is essential for enhancing food delivery efficiency and ensuring the safety and comfort of human–robot interaction (HRI) in meal-assisting robots [23–25].

In the fields of HRI and intelligent robotics, research on pose estimation technology has garnered significant attention [26–31]. Whitehill et al. [32] proposed a pose estimation method based on facial key points by detecting the centroids of both eyes, the nose tip, and the mouth and inputting these key points into a classifier, which finally performs the pose estimation using linear regression techniques. Fanellie et al. [33] utilized depth data and trained a random regression forest approach to estimate the 3D coordinates of the nose tip and head pose, which significantly improved the accuracy of the posture estimation. Luo et al. [34] proposed a multi-tree collaborative voting regression approach to enhance the accuracy of head pose estimation. Saeed et al. [35] concatenated the RGB and HOG features of facial images with depth information and integrated them with a linear SVM [36] for head pose estimation, thereby demonstrating the potential of multimodal data fusion. Ahn et al. [37] employed a convolutional neural network (CNN)-based regression model for efficient head pose estimation. Furthermore, Ahn et al. [38] proposed a multi-task CNN that integrates face detection, bounding box regression, and head pose estimation, thereby demonstrating the advantages of multi-task learning. Ruiz et al. [39] transformed the head pose regression problem into a classification task and estimated the head pose using an expectation function. Borghi et al. [40] proposed a deterministic GAN framework that converts depth images into grayscale maps and then fuses the segmented depth data with motion images to estimate head pose. Martin et al. [41] used the iterative closest point (ICP) algorithm to register the head model and consequently estimate the head pose. Ghiass et al. [42] achieved accurate head pose estimation by segmenting and refining face depth data from RGB images, followed by 3D facial model fitting using the ICP algorithm. Li et al. [43] reconstructed a facial model by merging multi-frame images and employed a face detection algorithm for model registration, thereby enhancing the accuracy of head pose estimation. Yuanquan et al. [44] proposed a head pose estimation method that integrates a deep neural network with 3D point cloud data, combining 3D angle classification with a graph convolutional neural network (GCNN). This approach achieved higher accuracy in pose estimation on the Biwi Kinect dataset. Mo et al. [45] proposed a hybrid architecture that combines CNN with GCNN and evaluated the model on the UBHPD database for head pose estimation using the Euler angles as regression targets. Wu et al. [46] proposed the Synergy method for comprehensive face detection and head pose estimation. Xin et al. [47] considered head pose estimation as a regression task and proposed the EVA-GCN method, which

https://doi.org/10.1017/S0263574725102737 Published online by Cambridge University Press

leverages GCNN to model the complex nonlinear relationship between head angles and graph structures, thereby achieving a substantial improvement in pose estimation accuracy. Chai et al. [48] employed bilinear pooling for head pose estimation, which effectively fused multi-modal features and demonstrated the benefits of feature fusion in improving estimation performance. Kao et al. [49] proposed a method that utilizes facial feature points obtained from monocular images to estimate facial pose. Cao et al. [50] introduced a TriNet network that incorporates the ResNet50 model and utilizes three orthogonal vectors in the rotation matrix to describe the head pose. Hempel et al. [51] proposed a six-degree-of-freedom RepNet network architecture to comprehensively estimate the facial posture, demonstrating the potential of multi-degree-of-freedom models. Xu et al. [52] achieved efficient head pose estimation by generating a 3D point cloud based on the YOLOv4 model and depth maps. Alternatively, Zhou et al. [53] proposed a novel head pose estimation architecture leveraging the YOLOv5 backbone and integrating multiple loss components to improve both accuracy and robustness. Redhwan et al. [54] introduced an unsupervised, real-time head pose estimation framework capable of estimating six degrees of freedom in an omnidirectional manner. The advantages and limitations of the methods used by the above researchers are shown in Table I. Although significant progress has been made in facial and head pose estimation through previous research efforts, the robustness of existing methods in challenging or uncontrolled environments still requires improvement. In addition, to the best of our knowledge, research on mouth pose estimation in facial images remains limited, particularly in terms of estimating mouth opening angles. Robust mouth pose estimation is particularly crucial for the food delivery process of meal-assisting robots, as it directly impacts the safety and user experience in HRI.

In this paper, we propose a new method of point cloud fitting and posture estimation of mouth opening degrees. The method takes the RGB-depth images of facial regions under a single view as the input and estimates the mouth postures by combining geometric methods and robust point cloud fitting algorithms. To validate the performance of the proposed algorithm, we conducted actual posture measurement experiments in multi-directional orientation and mouth shapes. Experimental results demonstrate that the proposed mouth pose estimation algorithm exhibits a small deviation from the actual posture and shows strong robustness and efficient computational performance in tests with different mouth shapes and multi-directional orientations. The main contributions of this study are summarized as follows:

1. A new hypothesis is proposed: the closed mouth is described by a closed spatial quadratic surface, the half-open mouth by a spatial ellipse, and the open mouth by a spatial circle.
2. Fitting algorithms for point clouds of mouths are proposed: the quadratic surfaces based on the least squares are used for fitting closed mouth; the algorithm of the inverse projection combined with least squares is used to fit spatial ellipse and spatial circle for the contour of half-open and open mouths.
3. A novel method integrating spatial geometric modeling and robust point cloud processing techniques is proposed to enable mouth pose estimation from single-view RGB-D images.
4. Actual posture measurement experiments are conducted under various orientations and different types of mouths.
5. This study can lay a theoretical foundation for trajectory planning in food delivery and can provide technical support for HRI safety in fields of robotics.

2. Proposed methods

This paper proposes a novel method for point cloud fitting and posture estimation of mouth opening degrees. The method takes RGB and depth images of facial regions from a single view as input and estimates the mouth posture by integrating geometric modeling techniques with robust point cloud fitting algorithms. The architecture of the FP-MODs method is illustrated in Figure 1. First, the RGB image is aligned with the depth image from the RealSense D405 depth camera [55]. Then, the state-of-the-art instance segmentation model (DCGW-YOLOv8n-seg model [16]) of mouth opening degrees is applied

https://doi.org/10.1017/S0263574725102737 Published online by Cambridge University Press

https://doi.org/10.1017/S0263574725102737 Published online by Cambridge University Press

Table I. *Summary of research methods from previous work.*

Method type	Representative work	Core technique	Advantages	Limitations	Applicable scenarios
Key point-based	Whitehill et al. [32]	Facial key points (eyes/nose/mouth) + linear regression	Lightweight computation, simple pipeline	Poor performance in complex expressions	Static facial pose estimation
	Fanelli et al. [33]	Depth data + random regression forest for 3D nose tip coordinates	Improved accuracy with depth information	Requires high-quality depth sensor	Structured indoor environments
Deep learning	Ahn et al. [37]	CNN regression model for real-time head pose estimation	End-to-end learning, adaptive to variations	High computational cost, large training data	Dynamic HRI
	Ahn et al. [38]	Multi-task CNN integrating face detection, bounding box regression, and pose estimation	Multi-task optimization, high accuracy	Complex architecture, slow inference	Multi-scene facial analysis
Geometry-based	Martin et al. [41]	Iterative closest point (ICP) algorithm for 3D head model registration	High precision in model alignment	Sensitive to initial pose and partial occlusions	Controlled medical imaging scenarios
	Ghiass et al. [42]	Depth image segmentation + 3D facial model fitting with ICP	Robustness to noisy depth data	Complex preprocessing pipeline	High-fidelity facial reconstruction
Multimodal fusion	Saeed et al. [35]	RGB features (HoG) + depth data + linear SVM for head pose estimation	Effective integration of visual and depth cues	Limited to rigid head movements	Static multi-sensor setups
	Borghi et al. [40]	Deterministic GAN for depth-to-grayscale conversion + motion image fusion	Robustness to lighting variations	Requirement for paired depth-motion datasets	Dynamic environment monitoring

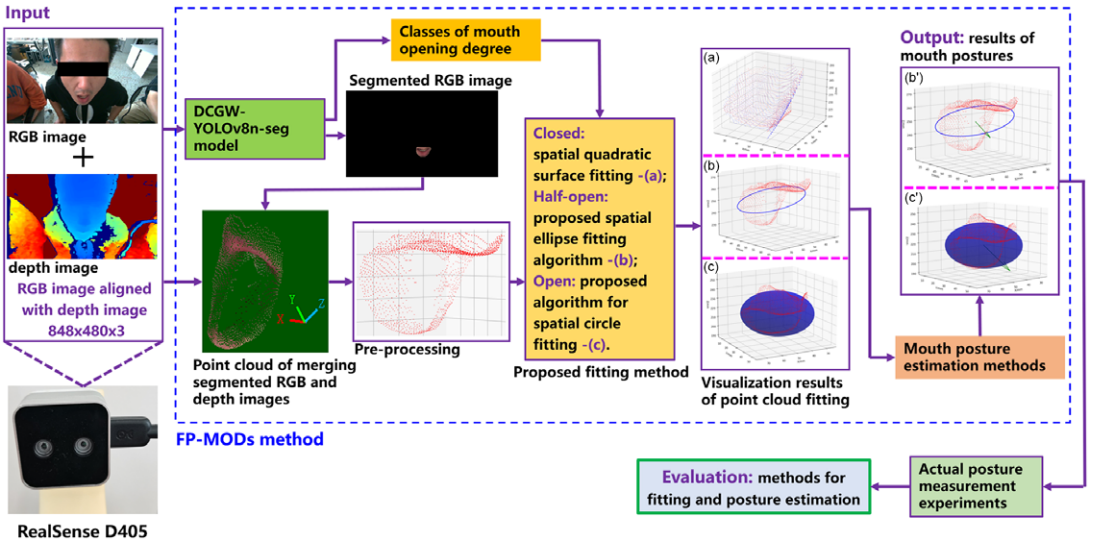


Figure 1. *The proposed method of point cloud fitting and posture estimation of mouth opening degrees.*

to classify and segment the mouth region in the face RGB image. Subsequently, the segmented RGB image is mapped to the depth image using the depth alignment method [56]. Afterward, the segmented RGB image and the depth image were merged to obtain the 3D point cloud data of mouths. Next, the point cloud data is preprocessed to reduce the computational memory and complexity. Then, the proposed algorithm of point cloud fitting is used to fit the mouth contour based on the category of mouth opening degrees. Finally, the Euler angles are applied to estimate the mouth posture by combining the geometric methods. We next describe in detail the point cloud preprocessing, the proposed hypothesis, mouth fitting algorithm, and the proposed method for predicting mouth posture.

2.1. Point cloud preprocessing

First, the RealSense D405 depth camera was employed to capture RGB and depth images of facial regions. Given that the resolutions of the RGB and depth images captured by the depth camera are 1280×720 and 848×480, respectively, they are uniformly scaled to 848×480 to ensure alignment between the RGB and depth images. Subsequently, the DCGW-YOLOv8n-seg model [16] is employed to perform segmentation on the RGB image, thereby obtaining the categories of mouth opening degrees (closed, half-open, and open) and masks. Next, the depth alignment method [56] is applied to project the mouth mask region from the RGB image onto the corresponding depth image, thereby obtaining a depth image that contains only the mouth region. Before reconstructing the 3D point clouds of mouth regions, the pixel coordinates of the mask in RGB image need to be converted to the 3D coordinates of point clouds. It is assumed that the pixel coordinates of mask regions can be represented by the set $w = \{(u_i, v_i)\}_{i=1}^{N}$, where N is the total number of pixels in mask region. The following equations [57, 58] are used to convert the pixel coordinates of w to the 3D coordinates $\{(x_i, y_i, z_i)\}_{i=1}^{N}$ of the point cloud X:

$$\begin{cases} z_i = I_{\text{depth}}\,(u_i, v_i) \\ x_i = \frac{z_i(u_i - C_x)}{F_x} \\ y_i = \frac{z_i(v_i - C_y)}{F_y} \end{cases} \tag{1}$$

where I_{depth} denotes the depth image and (C_x, C_y, F_x, F_y) denote the internal parameters of the RealSense D405 depth camera, encompassing focal lengths and principal point coordinates. Owing to the fixed optical architecture of the depth camera, these internal parameters remain invariant throughout the experimental paradigm. It is imperative to emphasize that the calibration protocol detailed herein is

https://doi.org/10.1017/S0263574725102737 Published online by Cambridge University Press

Table II. *The average point number of point cloud after down-sampling.*

Classes of mouth states	Before down-sampling	Voxel size = 1	Voxel size = 2	Voxel size = 3
Closed	2777	1307 (↓52.9%)	443 (↓84.0%)	223 (↓92.0%)
Half-open	3595	1659 (↓53.8%)	623 (↓82.7%)	319 (↓91.1%)
Open	4330	1877 (↓56.6%)	676 (↓84.4%)	352 (↓91.9%)
Average of three classes	3568	1615 (↓54.5%)	581 (↓83.7%)	298 (↓91.6%)

Note: The data in the table are averages of the point counts of 30 point clouds consisting of 10 participants in each category included in mouth state, and (↓a%) indicates that the point counts of the current point cloud have decreased by a% relative to the point counts of the point cloud that was before down-sampling.

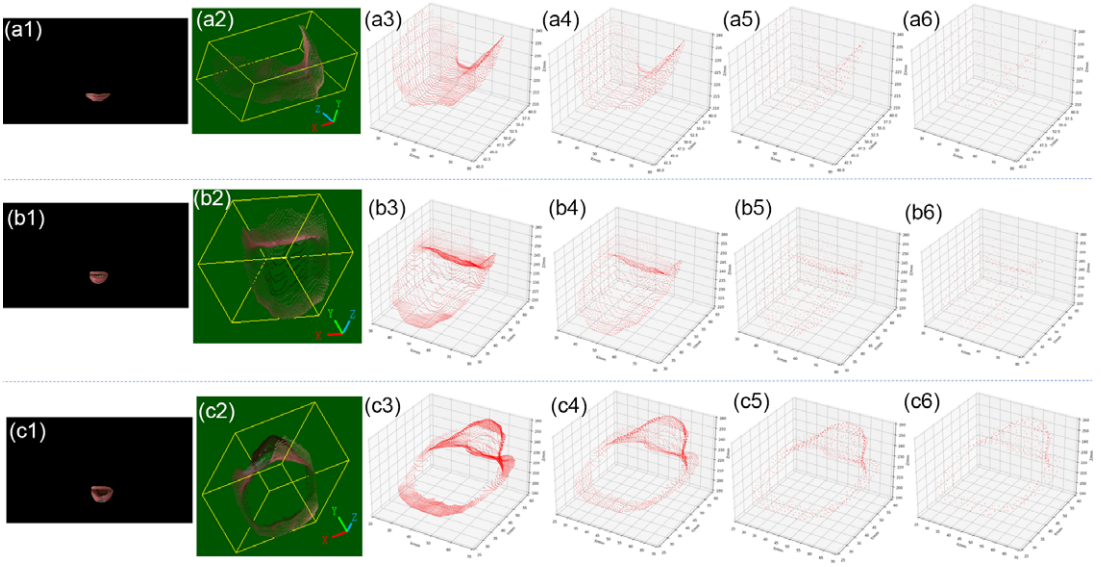


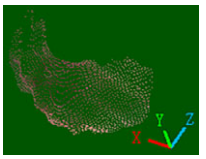
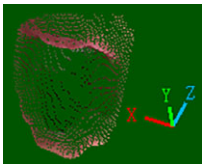
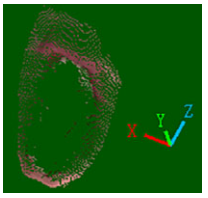
Figure 2. *The visualization results of point cloud down-sampling: letters a, b, and c denote mouth being closed, half-open, and open, respectively. Numbers 1, 2, 3, 4, 5, and 6 denote segmented RGB image of the mouth region, the CloudCompare v2.12.4 (Kyiv) visualizes the source point cloud, matplotlib visualizes the source point cloud data, the down-sampled point cloud (voxel size = 1), down-sampled point cloud (voxel size = 2), and down-sampled point cloud (voxel size = 3), respectively.*

exclusively concerned with internal parameter estimation, as the present study performs pose estimation within the camera's coordinate frame. Thereby rendering external parameters irrelevant to the relative orientation analysis of mouths. In this paper, 40 images of the checkerboard correction board are used to obtain the internal parameters of the camera and calibrate the camera by employing the *findChessboardCorners* and *calibrateCamera* functions in OpenCV.

Merging the segmented RGB image and depth image allows for the generation of a 3D point cloud of mouth regions in a single view. However, the point cloud obtained in this way contains a relatively large number of points (such as 2,777∼4,330 points before down-sampling, as shown in Table II), which increases the complexity and processing time of point cloud computation [59]. To reduce the number of points in the mouth point cloud, the *voxel_down_sample* function from the Open3D open-source library (http://www.open3d.org/) was applied, with the voxel size determined based on the desired number of voxels. In this study, voxel sizes of 1, 2, and 3 were used to reduce computational complexity, memory usage, and processing time. The result of down-sampling the mouth point cloud is shown in Figure 2 and Table II. As illustrated in Table II, the application of down-sampling significantly reduced the number of points in the mouth point cloud.

https://doi.org/10.1017/S0263574725102737 Published online by Cambridge University Press

Table III. *Description of proposed hypothesis and fitting methods.*

RGB image	Point cloud image	Classes of mouth states	Proposed hypothesis	Fitting methods
		Closed	Closed spatial surface	Spatial quadratic surface fitting method based on least squares optimization
		Half-open	Spatial ellipse	Fitting spatial ellipses based on a combination of inverse projection and least squares methods
		Open	Space circle	Fitting spatial circles based on the combination of inverse projection and least squares methods

2.2. Proposed hypothesis

For the morphological characteristics of the contours corresponding to different mouth opening degrees, this paper proposes a new hypothesis: the closed mouth region is described by a spatial quadratic surface, the contour of the half-open mouth is described by a spatial ellipse, and the contour of the open mouth is described by a spatial circle. Table III summarizes the proposed hypothesis and the fitting methods used in this paper for each mouth opening degree.

In the current research context of meal-assisting robotics, the choice of geometric models corresponding to different mouth opening states (a spatial quadratic surface for closed mouths, a spatial ellipse for half-open mouths, and a spatial circle for fully open mouths) over data-driven approaches is justified by three key considerations. First, regarding data availability: high-quality annotated 3D mouth datasets covering diverse opening states, orientations, and morphologies are currently limited in public repositories. The data-driven methods typically require thousands of labeled samples for generalization, and collecting such samples for specific scenarios like assistive feeding is time-consuming and labor-intensive. Second, in terms of real-time performance: data-driven models (especially deep learning-based ones) usually need GPU acceleration to achieve fast inference, yet even with that, their latency for 3D pose estimation often exceeds 50 ms, which fails to meet the strict real-time demands of food delivery (where delays may cause food spills). In contrast, the proposed geometric models, combined with lightweight least-squares fitting and voxel down-sampling, can achieve a latency as low as 5 ms on a mid-range CPU-GPU platform, ensuring real-time interaction. Third, concerning interpretability: geometric models have clear physical meanings (such as a spatial circle directly describes the inner lip contour of a fully open mouth), which makes it easier to debug and adjust parameters for specific user groups (such as, elderly individuals with limited mouth mobility). By comparison, the data-driven methods are often "black boxes" and less adaptable to individual variations in mouth morphology. It is worth noting that the geometric model approach does not negate data-driven methods. Instead, it is the

https://doi.org/10.1017/S0263574725102737 Published online by Cambridge University Press

optimal choice in the current research context, where real-time performance and adaptability to scarce data are prioritized.

2.3. Mouth fitting methods

2.3.1. Closed mouth fitting

For the point cloud data of the closed mouth, the least squares-based spatial quadratic surface fitting algorithm is employed. The fitting process is illustrated in Algorithm S1. In Algorithm S1, the RGB and depth images acquired by the depth camera are taken as inputs, while the quadratic surface parameters fitted by the algorithm with the fitting metrics are taken as outputs. The segmentation operation is first applied to both the RGB image and the depth image. Subsequently, the segmented RGB and depth images are merged to obtain a 3D point cloud of the mouth region, which is then preprocessed. Following this, the parameter matrices and arrays of the quadratic surface are calculated using the least squares method. Correlation coefficients are also computed, and the coefficients are updated accordingly. Finally, the quadratic surface parameters and fitting parameters are returned.

2.3.2. Half-open mouth fitting

For the point cloud corresponding to the half-open state of the mouth, an inverse projection method combined with least squares fitting is employed to model it as a spatial ellipse. First, it is assumed that the coordinate data of the point cloud of the half-open mouth can be described by $P_{half-open} = \{(x_i, y_i, z_i)\}_{i=1}^{N}$, where N represents the number of points in the point cloud corresponding to the half-open state of mouths. Then, the coordinates (x_i, y_i, z_i) of each point in the point cloud are added to the rotated data $\boldsymbol{R}$ and matrix $A_i = [x_i, y_i, 1]^T$ and vector $\boldsymbol{B}_i = z_i$ are constructed. All A_i and $\boldsymbol{B}_i$ are combined to form $A = [\boldsymbol{A_1}, \boldsymbol{A_2}, \ldots, \boldsymbol{A_N}]^T \in \mathbb{R}^{n\times 3}$ and the vector $\boldsymbol{B} = [\boldsymbol{B_1}, \boldsymbol{B_2}, \ldots, \boldsymbol{B_N}]^T \in \mathbb{R}^n$, namely:

$$\mathbf{A} = \begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ : & : & : \\ . & . & . \\ x_n & y_n & 1 \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} z_1 \\ z_2 \\ : \\ . \\ z_n \end{bmatrix} \tag{2}$$

Next, the least square method is applied to fit the plane equation $z = \alpha x + \beta y + \gamma$ with the goal of minimizing $\|Ax - \boldsymbol{B}\|^2$. By solving the regular equation

$$A^T A x = A^T \boldsymbol{B} \tag{3}$$

The fitting plane parameters α, β, and γ are obtained, namely $x = [\alpha, \beta, \gamma]^T$.

Computing the projection points (x_i^p, y_i^p, z_i^p) of each point (x_i, y_i, z_i) on the fitting plane, namely

$$\begin{cases} z_i^p = \alpha x_i + \beta y_i + \gamma \\ x_i^p = x_i \\ y_i^p = y_i \end{cases} \tag{4}$$

and combining the elliptic equation $ax^2 + bxy + cy^2 + dx + ey + f = 0$ at the point (x_i^p, y_i^p) in the projection plane, we construct the matrix $\mathbf{M}$ and the vector $\mathbf{0}$ such that $\mathbf{M}\cdot\mathbf{q} = \mathbf{0}$, the vector $\mathbf{q}=[a, b, c, d, e, f]^{\mathrm{T}}$ and the matrix $\mathbf{M}$ is

$$\mathbf{M} = \begin{bmatrix} (x_1^p)^2 & x_1^p y_1^p & (y_1^p)^2 & x_1^p & y_1^p & 1 \\ (x_2^p)^2 & x_2^p y_2^p & (y_2^p)^2 & x_2^p & y_2^p & 1 \\ : & : & : & : & : & : \\ . & . & . & . & . & . \\ (x_N^p)^2 & x_N^p y_N^p & (y_N^p)^2 & x_n^p & y_N^p & 1 \end{bmatrix} \tag{5}$$

https://doi.org/10.1017/S0263574725102737 Published online by Cambridge University Press

The elliptic parameters a, b, c, d, e, and f are obtained by solving $\mathbf{M}\cdot\mathbf{q}=\mathbf{0}$ by the least square method, which minimizes $\|\mathbf{M}\cdot\mathbf{q}\|^2$.

Next, calculate the long semi-axis ($a_{ellipse}$), the short semi-axis ($b_{ellipse}$), and the rotation angle ($\theta_{ellipse}$) of the ellipse, which is expressed as:

$$\begin{cases} a_{ellipse} = \sqrt{-\dfrac{f}{\lambda_2}} \\ b_{ellipse} = \sqrt{-\dfrac{f}{\lambda_1}} \\ \theta_{ellipse} = \dfrac{1}{2}\arctan\left(\dfrac{b}{a-c}\right) \end{cases} \tag{6}$$

where λ_1 and λ_2 are the eigenvalues of the matrix C and $\lambda_1 > \lambda_2$, and the matrix C is

$$C = \begin{bmatrix} a & \dfrac{b}{2} \\ \dfrac{b}{2} & c \end{bmatrix} \tag{7}$$

Then, the rotation matrix $\boldsymbol{R}_z(\theta_{ellipse})$ of the rotation angle $\theta_{ellipse}$ around the z-axis is computed with the expression:

$$\boldsymbol{R}_z\left(\theta_{ellipse}\right) = \begin{bmatrix} \cos\left(\theta_{ellipse}\right) & -\sin\left(\theta_{ellipse}\right) & 0 \\ \sin\left(\theta_{ellipse}\right) & \cos\left(\theta_{ellipse}\right) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{8}$$

For each projection point (x_i^p, y_i^p, z_i^p), coordinate rotation is performed by Eq. (9) and the point cloud data is updated as $\{(x_i^r, y_i^r, z_i^r)\}_{i=1}^N$, and the rotated data is updated $\boldsymbol{R}$:

$$\begin{bmatrix} x_i^r \\ y_i^r \\ z_i^r \end{bmatrix} = \boldsymbol{R}_z\left(\theta_{ellipse}\right)\begin{bmatrix} x_i^p \\ y_i^p \\ z_i^p \end{bmatrix} \tag{9}$$

Extract the two-dimensional data $\{(x_j^2, y_j^2)\}_{j=1}^M (M \le N)$ of the $x-y$ plane from the rotated data $\boldsymbol{R}$, and then fit the elliptic equation $a'x^2 + b'xy + c'y^2 + d'x + e'y + f' = 0$ to the two-dimensional data $\{(x_j^2, y_j^2)\}_{j=1}^M$ using the least square method, and then calculate the center (x_c, y_c), the long semi-axis (a_2), the short semi-axis (b_2), and the short semi-axis (θ_2) according to the above.

Next, for the x and y coordinates on the ellipse, using Eq. (10) to describe

$$\begin{cases} x = a_2\cos(t) + x_c \\ y = b_2\sin(t) + y_c \end{cases} \tag{10}$$

where t is a uniformly distributed parameter of $[0, 2\pi]$ and add (x, y) to the point set $\boldsymbol{P} = \{(x_k, y_k)\}_{k=1}^{N_2}, N_2 \le N$.

Then, calculate the average z-coordinate in the rotated data $z_{mean} = \frac{1}{N}\sum_{i=1}^N z_i^r$. Next, the z_{mean} is added to each point in the point set $\boldsymbol{P}$ and to the center (x_c, y_c) of the ellipse to obtain a new point set $\boldsymbol{P'} = \{(x_k, y_k, z_{mean})\}_{k=1}^{N_2}$ and the center point (x_c, y_c, z_{mean}). Using the rotation matrix $\boldsymbol{R}_z(-\theta_2) = \begin{bmatrix} cos(\theta_2) & sin(\theta_2) & 0 \\ sin(\theta_2) & cos(\theta_2) & 0 \\ 0 & 0 & 1 \end{bmatrix}$, which is in the opposite direction of the previous rotations, perform coordinate back-rotation on $\boldsymbol{P'}$ and the center point (x_c, y_c, z_{mean}). Get the rotated point set $\boldsymbol{P''} = \{(x_k'', y_k'', z_k'')\}_{k=1}^{N_2}$ and the center point (x_c'', y_c'', z_c''). Arbitrarily choose three points that are not covariant $P_1 = (x_{k_1}'', y_{k_1}'', z_{k_1}'')$, $P_2 = (x_{k_2}'', y_{k_2}'', z_{k_2}'')$, and $P_3 = (x_{k_3}'', y_{k_3}'', z_{k_3}'')$ in the point set $\boldsymbol{P''}$, where k_1, k_2, and$k_3 \epsilon \{1, 2, \ldots, N_2\}$, and compute the normal vector $n = (P_2 - P_1) \times (P_3 - P_1)$.

https://doi.org/10.1017/S0263574725102737 Published online by Cambridge University Press

Algorithm S2 describes a spatial ellipse fitting algorithm that combines the inverse projection method described above with the least-squares method. This algorithm was used to fit the point cloud data of the half-open mouth to a spatial ellipse.

2.3.3. Open mouth fitting

When the mouth is in an open state, the inner contour of the lips exhibits a morphology similar to that of a spatial circle. Based on this observation, we employ a combination of the inverse projection method and the least squares method to fit the point cloud data of the open mouth to a spatial circle. Algorithm S3 describes the corresponding fitting process.

First, the processing is performed on each pair of RGB and depth images in the image list. The input RGB and depth images contain information related to the open mouth scene. A segmentation operation is applied to the RGB images to obtain the corresponding segmented RGB images. This initiative helps to extract the region of interest and reduces the complexity of subsequent processing. Based on the depth image, the segmented RGB image, and the depth alignment method, depth information is extracted, thereby obtaining the segmented depth image. The segmented RGB image is merged with the depth image to generate point cloud data, which characterizes the three-dimensional structure of the open mouth. Subsequently, the generated point cloud is preprocessed to remove noise and anomalies, thereby improving data quality. Finally, the coordinate information of the preprocessed point cloud is extracted. A matrix is constructed based on the point cloud coordinates, which will be used in the subsequent circular parameter fitting. Subsequently, the least squares method is applied to fit this matrix to the circular parameters, thereby obtaining the preliminary fitting parameters. The rotation angle and rotation axis are extracted from the preliminary fitting parameters. This rotational information is then used to reorient the point cloud, thereby enabling more efficient circular parameter fitting. Subsequently, a rotation matrix is computed based on the extracted rotation angle and axis. The original point cloud is rotated using this matrix to obtain a rotated point cloud. Finally, the rotated point cloud is fitted again with circular parameters to achieve more accurate results. The center and radius of the circle are subsequently extracted from the final fitted parameters.

The absolute value of the radius is taken to ensure positivity. The parameters of the fitted circle are then generated based on this radius. Using the center, radius, rotation angle, and the generated parameters, the coordinates of the circle's points are calculated. Each point is augmented with its corresponding Z-coordinate, and the average Z-value is computed. This average is added to the center, updating its coordinates. The updated center and rotation matrices are used to rotate the points and center back to their original orientation. Three points are selected from the rotated set, and the normal vector is computed from them. Finally, the algorithm outputs the center, radius, rotation angle, and normal vector of the spatial circle, which fully describe the fitted geometry.

2.4. Mouth posture predicting method

For the half-open mouth and open mouth states, we fit their point clouds to obtain a spatial ellipse and a spatial circle, respectively. Based on the description of the caregiver and the feedback from the users, the end-effector posture of meal-assisting robots needs to be adjusted in real time during the food delivery process to keep it parallel to the normal direction of the plane on which the inner contour of the users' lips is located. Given this, we define the predicted mouth pose as the direction of the normal vector of the ellipse or circle obtained from the fitting, which is the direction in which the fitting surface points away from faces. In addition, the starting point of the normal vector is defined as the center of the plane in which the fitted ellipse or circle is located. Therefore, the mouth pose vector presupposed can be described as pointing in the direction of the outer normal from the center of the fitting plane, as shown in Figure 3.

The object has six degrees of freedom without any constraints, covering movement along three directions (x, y, z) and rotation around three directions (θ, φ, and ω, around the Z-axis, Y-axis, and X-axis,

https://doi.org/10.1017/S0263574725102737 Published online by Cambridge University Press

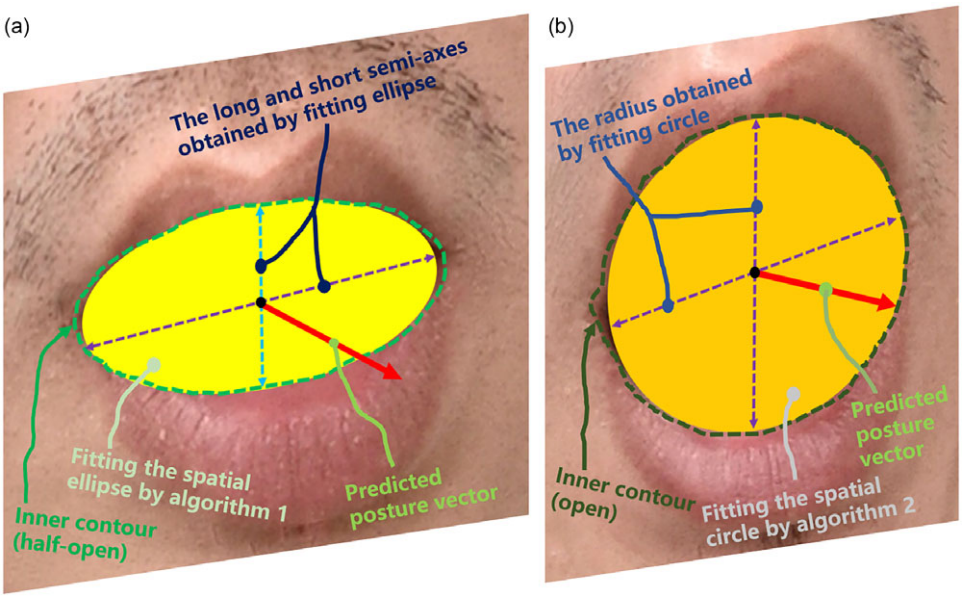


Figure 3. *Schematic diagrams of the proposed method for estimating the predicted posture vectors: (a) the inner contour showing an approximate shape of a spatial ellipse when the mouth is in a half-open state by the Algorithm S2, and (b) the inner contour showing an approximate shape of a spatial circle when the mouth is in an open state by the Algorithm S3. Where, the yellow region and the orange region are the spatial ellipse and the spatial circle obtained by the fitting of Algorithms S2 and S3, respectively, and the red arrow line is the predicted posture vector.*

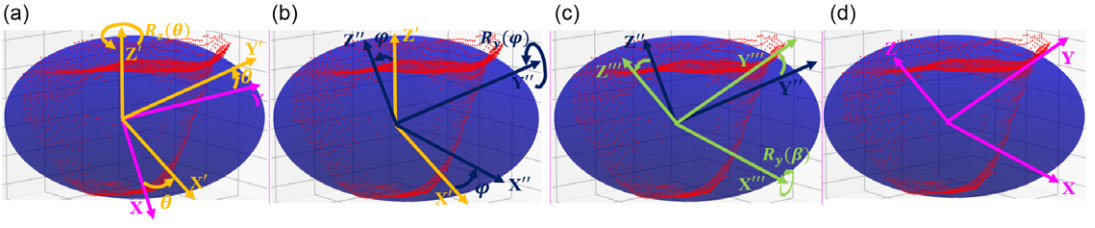


Figure 4. *Schematic representation of the Euler-ZYX angle applied to mouth regions of meal-assisting robotics during meal delivering: (a) rotation around the Z-axis, (b) rotation around the Y-axis, (c) rotation around the X-axis, and (d) posture orientation.*

respectively). The mouth will change accordingly with the movement of the head. Given this, the Euler-ZYX angle is used to characterize the orientation of the mouth posture during food delivery, as shown in Figure 4. Therefore, we define the position and orientation of the mouth during food delivery, represented by the transformation matrix T_{mp}:

$$T_{mp} = \begin{bmatrix} c\theta c\varphi & c\theta s\varphi s\omega - c\omega s\theta & s\theta s\omega + c\theta c\omega s\varphi & x \\ c\varphi s\theta & c\theta c\omega + s\theta s\varphi s\omega & c\omega s\theta s\varphi - c\theta s\omega & y \\ -s\varphi & c\varphi s\omega & c\varphi c\omega & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{11}$$

where "*s*" and "*c*" denote the sine function "*sin*" and the cosine function "*cos*", respectively.

https://doi.org/10.1017/S0263574725102737 Published online by Cambridge University Press

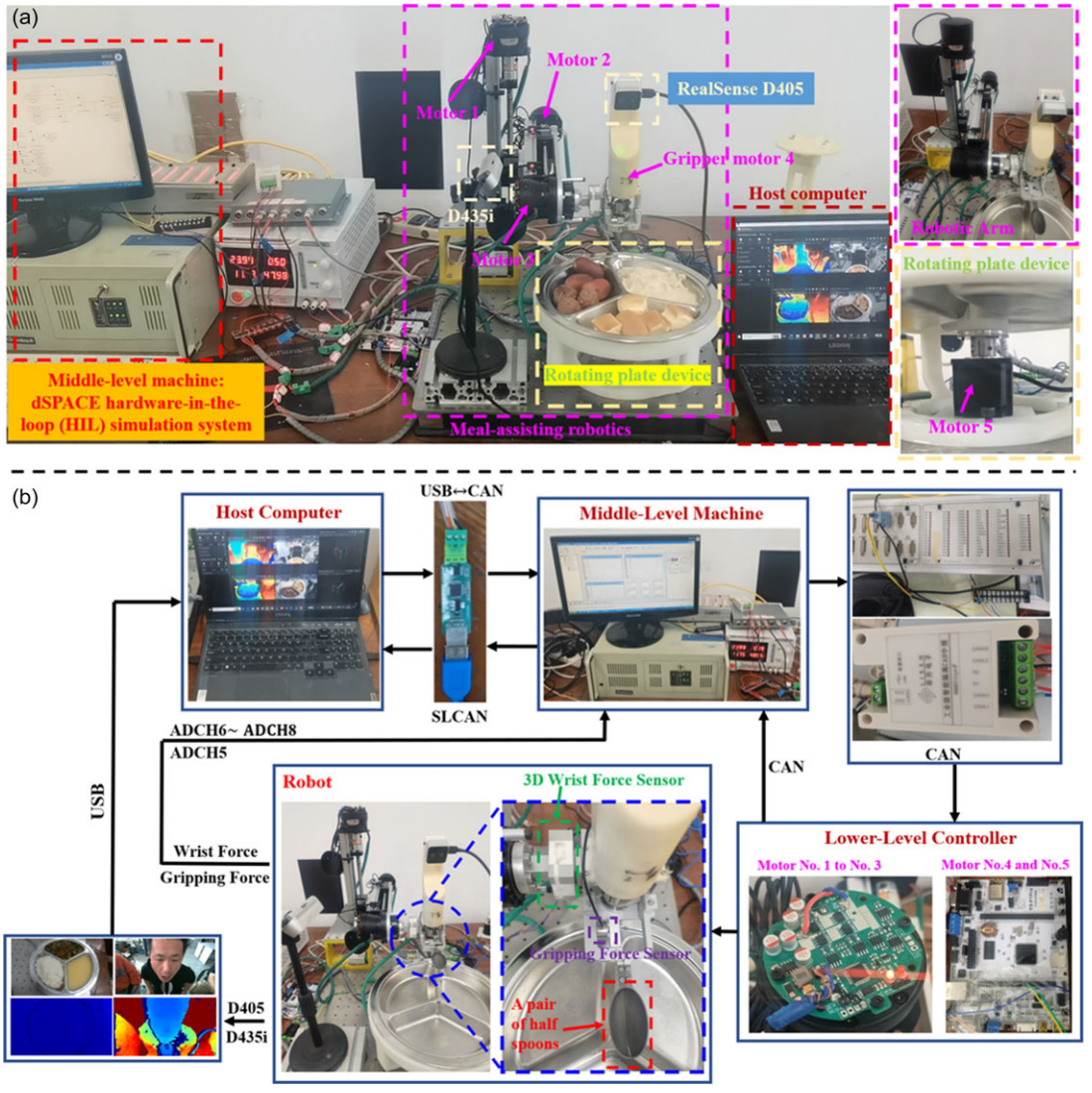


Figure 5. *(a) Meal-assisting robotic testbed system and (b) hardware and communication of the testbed.*

3. Experiments and assessment metrics

3.1. Experimental platform

The hardware architecture of meal-assisting robotic testbed system is shown in Figure 5. The system is divided into four functional units: upper computer, middle computer, lower computer, and the meal-assisting robot body. The upper computer is a laptop running Windows 10 (Intel R Core TM i5-10300H CPU and NVIDIA GeForce GTX 1650 GPU), primarily executing food target recognition and positioning algorithms in robot vision module, feeding vision algorithms such as food posture estimation, as well as food delivery vision algorithms including mouth recognition and positioning, posture estimation, and facial abnormal expression detection. The middle computer is a dSPACE semi-physical simulation system (dSPACE GmbH, Germany), with core functions in the meal-assisting experiment including: real-time reception of visual feedback commands from the upper computer, as well as force sensor and encoder feedback from the lower computer and robot; execution of path planning algorithms in the meal-assisting robot motion planning module, trajectory generation algorithms, and joint trajectory tracking algorithms in servo control module. The lower computer is a customized motor driver

https://doi.org/10.1017/S0263574725102737 Published online by Cambridge University Press

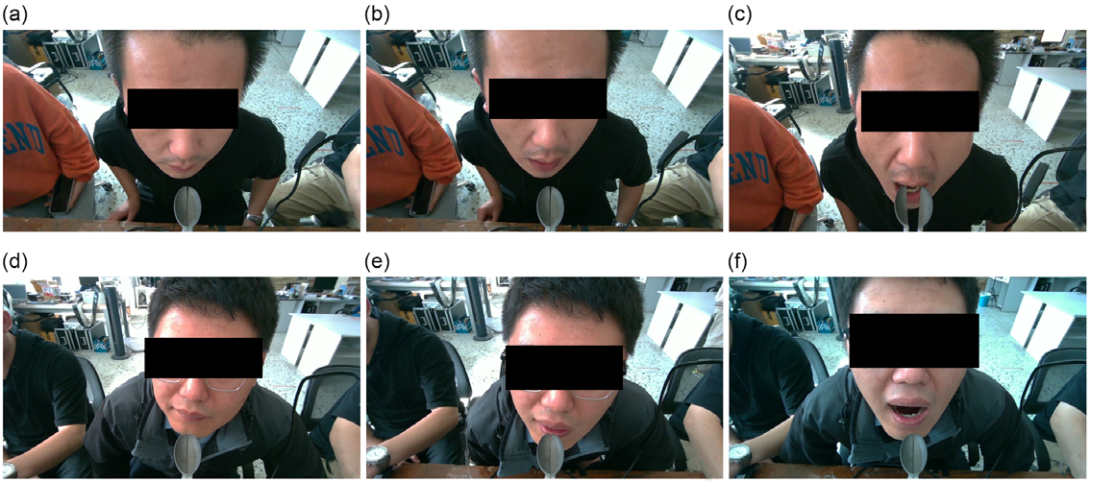


Figure 6. *Selected RGB frames captured using the RealSense D405 depth camera: (a) and (d) correspond to the closed mouth state; (b) and (e) the half-open mouth state; (c) and (f) the open mouth state.*

board and STM32F103 development board, responsible for parsing control commands sent by the middle computer and driving DC servo motors of each robotic arm joint for motion. The system uses a CAN bus as the data communication link to enable real-time information interaction among the upper, middle, and lower computers.

As shown in Figure 5, the meal-assisting robotics testbed consists of a robotic arm, end-effector mechanism, plate rotation mechanism, two depth cameras, and two force sensors. The depth cameras include the RealSense D435i (RGB resolution 1280 × 720, depth resolution 848 × 480, Intel, USA), and the RealSense D405 (RGB resolution 1280 × 720, depth resolution 848 × 480, Intel, USA). The RealSense D435i primarily captures images and point cloud data of meals, while the RealSense D405 acquires facial images and point cloud data of users. The rotating plate mechanism uses a common commercial plate. The robotic arm comprises DC torque motors No.1–3 (rare-earth permanent magnet DC torque 45LYX04 motor), a RealSense D405 depth camera, force sensors, a lead screw module (SGX43-1204, outer diameter 12 mm, pitch 4 mm, produced by Duoshu Code Technology, China), and two half-spoons [60].

3.2. Data acquisition and experiments

The localized camera (RealSense D405, in Figure 5) is used to capture RGB and depth images of faces. The RGB image from the RealSense D405 camera has a resolution of 1280 × 720, while the depth image has a resolution of 848 × 480. To ensure spatial alignment, both images are resized to 848 × 480.

The classification of mouth opening degrees was based on user feedback and the physical dimensions of spoons. Following prior work [15, 16], we defined three categories: closed mouth, half-open mouth, and open mouth. We conducted RGB and depth image acquisition of facial regions of the participants. Specifically, 39 teachers and students were invited to participate in this facial image acquisition experiment and a total of 1,398 RGB-D image pairs were acquired, including 429 RGB-D image pairs with closed mouths, 492 RGB-D image pairs with half-open mouths, and 477 RGB-D image pairs with open mouths. These facial images were acquired from participants in different orientations and under different external light intensities. Specifically, the experiment began on July 20, 2023, and ended on October 30, 2023. The collection site was located in the laboratory of Harbin Engineering University, No. 145 Nantong Street, Harbin City, Heilongjiang Province, China. Each collection was conducted during three time periods: 8–10 am, 11 am–2 pm, and 6-8 pm, which was designed to fit the meal time of most people. Figure 6 presents an example of an RGB sample acquired with the RealSense D405 depth camera.

https://doi.org/10.1017/S0263574725102737 Published online by Cambridge University Press

The DCGW-YOLOv8n-seg [16] instance segmentation model for faces and mouth openings has been trained and tested in previous research work. For more information about the DCGW-YOLOv8n-seg model, the reader is referred to the literature [16]. Therefore, in this work, the DCGW-YOLOv8n-seg model is used to perform instance segmentation on the captured RGB images to obtain the categories of mouth opening degrees and mask regions. Additionally, 40 images of a checkerboard calibration pattern are used to estimate the intrinsic parameters of the camera and perform calibration using the *findChessboardCorners* and *calibrateCamera* functions in OpenCV. This calibration enables the accurate conversion of pixel coordinates from RGB image masks into corresponding 3D point cloud coordinates. The algorithms in this paper are implemented on the Windows 10 operating system, with the PyTorch framework leveraged to load and execute the pretrained DCGW-YOLOv8n-seg model for instance segmentation inference. The test workstation configuration used is Intel (R) Core (TM) i5-10300H CPU and NVIDIA GeForce GTX 1650 GPU. In addition, we have installed the following software and open-source libraries: conda 22.9.0, CUDA 11.7, cuDNN 8.4, PyTorch 1.13.1, Python 3.8.17, PyCharm 2023, and Open3D 0.17.0.

3.3. Actual posture measurement experiments

To validate the accuracy of the proposed method for mouth posture estimation, we conducted mouth actual posture measurement experiments. The number of actual pose measurement experiments was equal to the number of RGB-D image pairs acquired, with one experiment performed for each pair. A total of 1,398 RGB-D image pairs were used in the experiments, consistent with the description provided in Section 3.2. The detailed procedure for the actual pose measurement experiment for each pair of RGB-D images is outlined.

First, the RealSense D405 depth camera mounted on the end-effector is stationary regarding the optical platform (in Figure 5). In this case, the changing factors are the different orientations of the participants and the external light conditions. At the same time, the 3D coordinates of the depth camera relative to the base of the optical platform were measured to ensure the accuracy and traceability of subsequent coordinate changes. Subsequently, the RealSense D405 camera was used to capture and store the RGB-D image of each participant and record the actual 3D coordinates of eight points in the mouth region of the participant relative to the base of the optical platform. The measured spatial coordinates of the eight points were then converted to Open3D point cloud spatial coordinates. Given that the point cloud 3D coordinates obtained from the acquired RGB-D images after segmentation, alignment, and merging are relative to the RealSense D405 camera, they were further transformed into point cloud 3D coordinates relative to the optical platform substrate, facilitating subsequent pose estimation. Subsequently, the widely used randomized sampling consensus (RANSAC) algorithm [61], which exhibits superior performance in terms of noise immunity and robustness, is employed to fit a plane to a set of eight points in the point cloud data processed in Open3D. Finally, the center point of the plane fitted by the RANSAC algorithm is taken as the origin of the actual posture vector, while the direction of the outward normal of the plane is used as the direction of the posture vector, representing the actual mouth posture, as illustrated in Figure 7.

3.4. Assessment metrics

To evaluate the accuracy and computational efficiency of the proposed algorithm for mouth region fitting and pose estimation, we have selected a set of widely adopted evaluation metrics. In the context of mouth region fitting, the root mean square error (RMSE) is commonly utilized to evaluate the global deviation between the reconstructed point cloud and the corresponding real-world point cloud. This metric places more emphasis on larger discrepancies, thus providing an effective measure of the error magnitude during the estimation process. A smaller RMSE value suggests that the algorithm produces results that are more consistent with the actual data, indicating superior performance in terms of accuracy. Suppose

https://doi.org/10.1017/S0263574725102737 Published online by Cambridge University Press

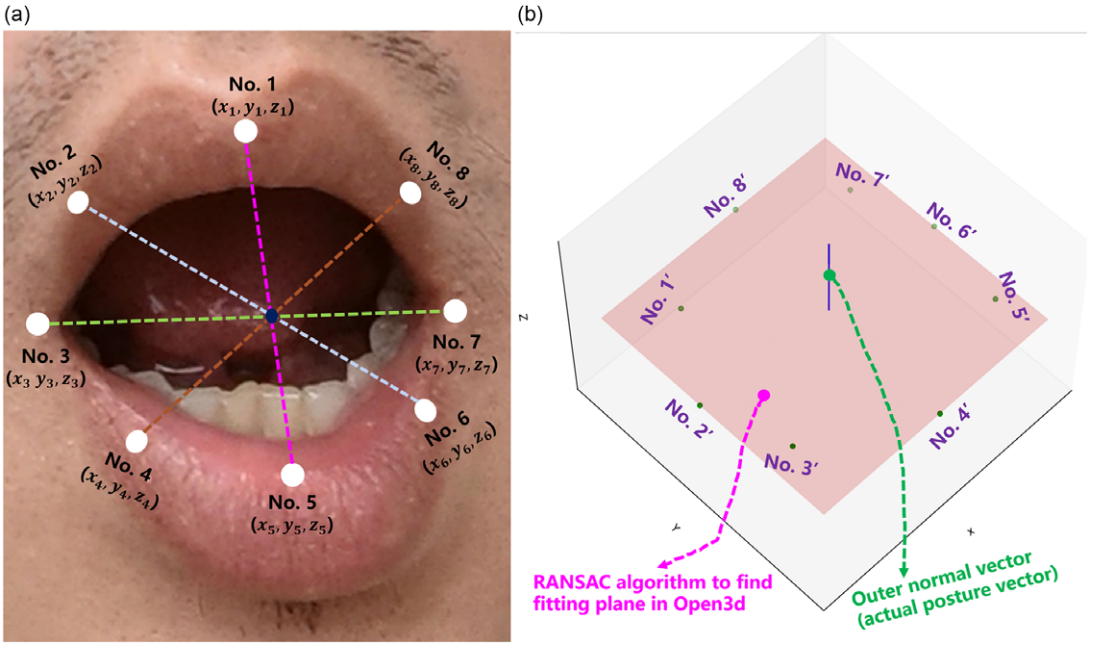


Figure 7. *Schematic representation of the actual posture vector: (a) the schematic representation of the positions of the eight points of the mouth contour (spatial coordinates x, y, and z under the reference coordinate system of the optical platform) measured during the acquisition of the mouth RGB and depth images; (b) the eight points in (a) merged in the eight points in the Open3d point cloud, and then based on the eight points fitted by the RANSAC algorithm into a plane, and the outer normal vector of the plane represents the actual posture vector of mouths.*

the point set of the fitted point cloud of the algorithm is $\{P_f = (x_f^i, y_f^i, z_f^i)\}$, and the point set of the actual point cloud is $\{P_a = (x_a^i, y_a^i, z_a^i)\}$, where i=1,2,. . .,N denotes the serial number of the points in the point cloud, and N is the total number of the points. The RMSE can be described as:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left\|P_f^i - P_a^i\right\|^2} \tag{12}$$

where $\|\cdot\|$ denotes the Euclidean paradigm.

Additionally, the mean absolute error (MAE) is commonly used to quantify the deviation between the fitted point cloud generated by the algorithm and the corresponding real-world point cloud. In contrast to RMSE, which applies equal weighting to all errors, MAE computes the average of the absolute differences, without fitted point cloud and amplifying the impact of larger deviations. This characteristic makes MAE a more straightforward and interpretable metric for evaluating the overall accuracy of the fitting process. A smaller MAE value means that the point cloud fitting result of the algorithm is closer to the actual point cloud and shows better accuracy and reliability. Mathematically, the MAE is defined as:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left\|P_f^i - P_a^i\right\| \tag{13}$$

Next, in the mouth pose estimation, the coefficient of determination (R^2) is used to evaluate the extent to which the algorithm fits the mouth point cloud. The value of R^2 ranges from 0 to 1. A value closer to 1 indicates that the algorithm fits the mouth point cloud data better, while a value closer to 0 means that the algorithm fits the mouth point cloud data relatively poorly. R^2 can be described as:

https://doi.org/10.1017/S0263574725102737 Published online by Cambridge University Press

$$R^2 = 1 - \frac{\sum_{i=1}^{N} \left\| P_a^i - P_f^i \right\|^2}{\sum_{i=1}^{N} \left\| P_a^i - \overline{P_a} \right\|^2} \tag{14}$$

where $\overline{P_a}$ denotes the mean point of the actual mouth point cloud, that is,

$$\overline{P_a} = \left(\frac{1}{N} \sum_{i=1}^{N} x_a^i, \frac{1}{N} \sum_{i=1}^{N} y_a^i, \frac{1}{N} \sum_{i=1}^{N} z_a^i \right) \tag{15}$$

In addition, the deviation of the mouth posture vector ($\varepsilon\beta$) is introduced, which quantifies the difference degree between the predicted and actual posture vectors. The $\varepsilon\beta$ visualizes the accuracy of the algorithm in predicting the posture. A smaller value of the $\varepsilon\beta$ means that the predicted posture vector is closer to the actual posture vector. Further, to synthesize the accuracy of the algorithm in predicting posture in the overall mouth database, the average of the posture vector deviations ($\overline{\varepsilon\beta}$) is used to characterize the posture estimation performance of multiple samples. The smaller average pose vector deviation indicates that the algorithm can predict the pose more accurately in most cases with high reliability and generalization ability. The mathematical expression for the mean of $\overline{\varepsilon\beta}$ is

$$\overline{\varepsilon\beta} = \frac{1}{N} \sum_{i=1}^{N} \varepsilon\beta_i \tag{16}$$

where $\varepsilon\beta_i$ denotes the posture vector deviation of the ith point cloud sample. In this research work, we describe the pose vector deviation by adopting the Euler-ZYX angle. The $\varepsilon\beta_i$ is expressed as:

$$\varepsilon\beta_i = \sqrt{\varepsilon\beta_{z,i}^2 + \varepsilon\beta_{y,i}^2 + \varepsilon\beta_{x,i}^2} \tag{17}$$

where

$$\begin{cases} \beta_{z,i} = \left| \theta_{a,i} - \theta_{f,i} \right| \\ \beta_{y,i} = \left| \varphi_{a,i} - \varphi_{f,i} \right| \\ \beta_{x,i} = \left| \omega_{a,i} - \omega_{f,i} \right| \end{cases} \tag{18}$$

where $(\theta_{a,i}, \varphi_{a,i}, \omega_{a,i})$ represents the Euler-ZYX angles obtained from the actual posture vectors and $(\theta_{f,i}, \varphi_{f,i}, \omega_{f,i})$ represents the Euler-ZYX angles resulting from the conversion of the estimated posture vectors.

Time efficiency is an essential aspect in the practical implementation of mouth pose estimation. A reduced average time ($\bar{t}$) suggests that the algorithm can execute pose estimation more rapidly, leading to improved system response and real-time performance. Mathematically, the $\bar{t}$ is defined as:

$$\bar{t} = \frac{1}{N} \sum_{i=1}^{N} t_i \tag{19}$$

where t_i denotes the time consumed by the algorithm in processing the ith point cloud sample.

4. Results and discussion

4.1. Impact of different voxel numbers on mouth pose estimation

To improve the efficiency of mouth fitting and pose estimation, we perform down-sampling operations on the point cloud data obtained from the 3D reconstruction of mouths. In this study, we set the voxel numbers to 1, 2, and 3 to maximize the processing speed of the algorithm. Furthermore, to investigate the impact of varying voxel numbers on the fitting performance of the algorithm and the accuracy of posture estimation, we performed practical posture measurement experiments, the results of which are shown in Figure 8 and Table IV.

Figure 8 shows the visualization results of the fitting and pose estimation of the algorithm for different voxel numbers. In Figure 8, the number of points in the point cloud of mouth regions decreases rapidly as the number of down-sampled voxels increases. However, the proposed fitting algorithm shows

https://doi.org/10.1017/S0263574725102737 Published online by Cambridge University Press

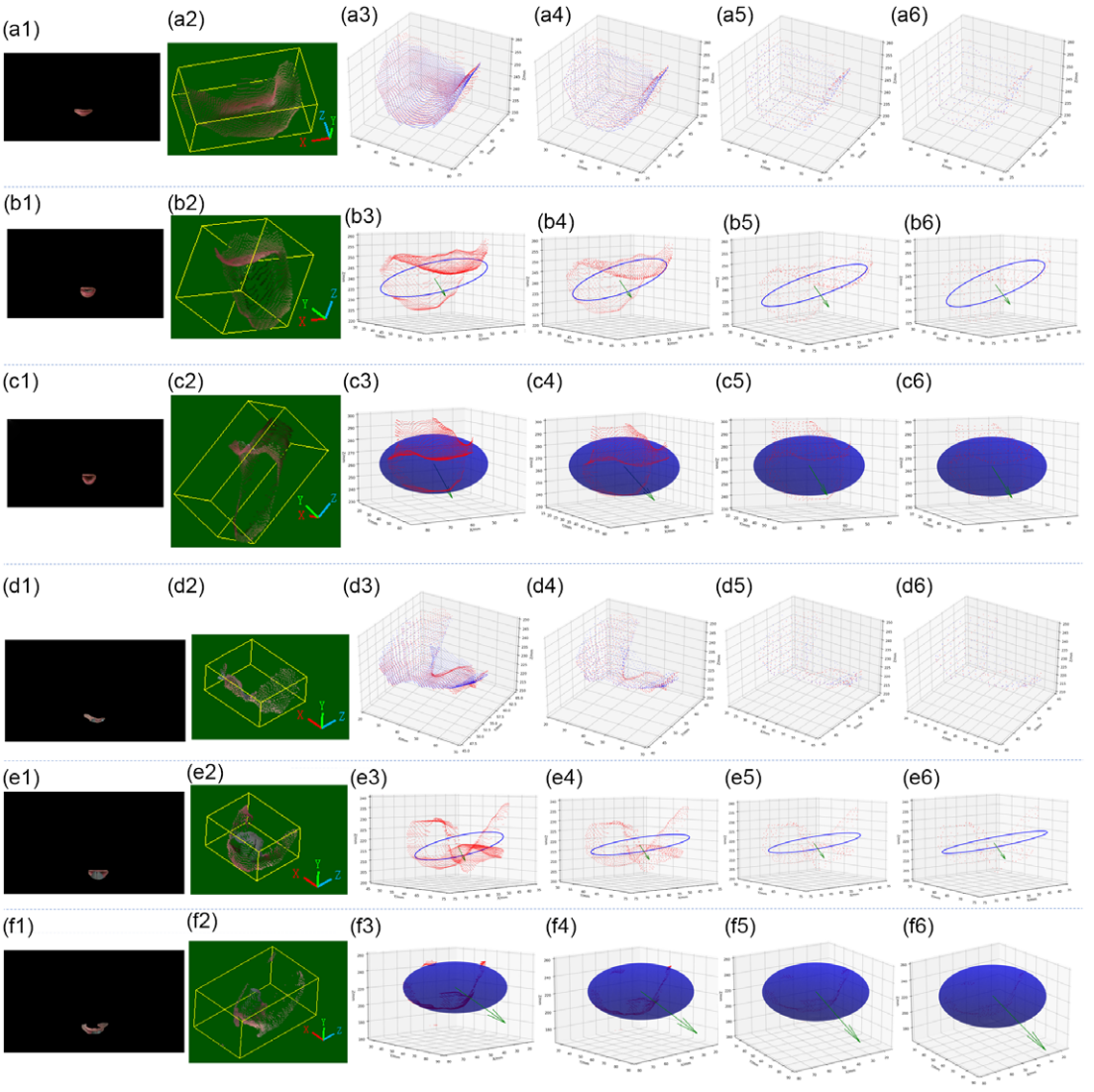


Figure 8. *The results of point cloud fitting and posture estimation for mouth openings with different down-sampling methods: numbers 1, 2, 3, 4, 5, and 6 denote the segmented RGB images of the mouth region, and CloudCompare v2.12.4 (Kyiv) visualizes the source point cloud of the mouth region, the fitted result of the source point cloud, the fitted result of the point cloud with down-sampling (voxel size = 1), the fitted result of the point cloud with down-sampling (voxel size = 2), and the fitted result of the point cloud with down-sampling (voxel size = 3), respectively. (a) and (d), (b) and (e), and (c) and (f) denote the mouth in closed state, the mouth in half-open state, and the mouth in open state, respectively. Where, (a), (b), and (c) denote the fitting and posture results for the normal segmentation case, while (d), (e), and (f) denote the fitting and posture results for the poor segmentation case. The red points in the figure indicate point cloud data. The blue points, blue closed curves, and blue spheres represent the fitted geometry. The green arrows indicate the posture estimation of the mouth in the half-open or open state.*

https://doi.org/10.1017/S0263574725102737 Published online by Cambridge University Press

Table IV. *The results of different down-sampling methods on fitting and posture vectors.*

Mouth state	Down-sampling	Number	RMSE	MAE	R^2	$\bar{t}$/(ms)	$\overline{\varepsilon\beta}$/(°)
Closed	No down-sampling	30	1.231	0.985	0.965	53.79	–
	Voxel size = 1	30	1.248	0.994	0.960	26.48	–
	Voxel size = 2	30	1.254	1.003	0.960	11.92	–
	Voxel size = 3	30	1.246	1.001	0.962	7.82	–
Half-open	No down-sampling	30	–	–	–	5.33	3.59
	Voxel size = 1	30	–	–	–	4.82	3.71
	Voxel size = 2	30	–	–	–	4.15	3.74
	Voxel size = 3	30	–	–	–	3.97	4.27
Open	No down-sampling	30	–	–	–	5.33	2.73
	Voxel size = 1	30	–	–	–	4.53	2.84
	Voxel size = 2	30	–	–	–	4.23	2.90
	Voxel size = 3	30	–	–	–	4.10	3.18

Note: The resultant data in the table are averaged over 30 pairs of point cloud obtained by merging the RGB and depth images, $\bar{t}$ denotes the average value of the time consumed in preprocessing and fitting process, and $\overline{\varepsilon\beta}$ denotes the average of the deviation of the actual posture vector and the predicted posture vector. "-" indicates metrics not applicable to specific mouth opening states. For half-open/open states, there is no correspondence between the fitted data and the original point cloud; for the closed state, mouth pose estimation is not required.

good results and is not affected by the change in voxel number. This indicates that the quadratic surface algorithm, optimized by the least-squares method, exhibits superior fitting capability for the mouth region in the closed mouth state. In addition, the algorithm using the inverse projection method combined with the least-squares method is effective in fitting the spatial ellipse and spatial circle for the half-open and open mouth regions. As shown in Figure 8, the pose estimation of mouth opening degrees by the proposed algorithm remains unaffected by variations in the number of voxels. These results demonstrate that the proposed algorithm exhibits strong robustness in estimating mouth opening degrees.

In Table IV, the quadratic surface optimization algorithm is used to fit the contours of the closed mouth region to achieve better results. The RMSE values range from 1.231 to 1.254, demonstrating that the deviation between the algorithm's fitted point cloud and the actual point cloud is relatively small, which reflects the high accuracy of the proposed method. Given that the RMSE assigns a higher weight to larger deviations, this range indicates that larger deviation scenarios are not significant across the entire data sample. Overall, the errors of the algorithm in predicting point cloud locations are relatively stable and do not show particularly large and unusual deviations. In addition, the MAE ranges from 0.985 to 1.003, which further confirms that the algorithm exhibits a small average deviation in predicting the positions of points in the point cloud. The MAE assigns equal weight to all errors, and this range suggests that the algorithm demonstrates good accuracy and reliability in predicting individual points with a relatively small deviation from the actual point cloud. In addition, the coefficient of determination reaches up to 0.96, indicating that the algorithm demonstrates a strong fit with the mouth point cloud. Furthermore, the impact of varying voxel numbers on the fitting algorithm is minimal, suggesting that the mouth fitting algorithm is highly robust.

On the other hand, in Table IV, the voxel number size of the down-sampling has an impact on the processing speed of the fitting algorithm, especially regarding the processing of the point cloud data when closing mouths. On a general hardware device (such as Intel (R) Core (TM) i5-10300H CPU and NVIDIA GeForce GTX 1650 GPU), the processing time of the fitting algorithm for the closed mouth state reaches 53.79 milliseconds, which is not enough to achieve real-time effect. However, the down-sampling method can significantly enhance the processing speed of the algorithm, making real-time processing achievable on general hardware devices. In the posture estimation of mouth opening degree, different voxel numbers have less effect on the algorithm combining the inverse projection method with the least square method. This indicates that the average difference between the mouth posture vectors predicted by the algorithm and the actual posture vectors is more stable and does not fluctuate drastically

https://doi.org/10.1017/S0263574725102737 Published online by Cambridge University Press

Table V. *The results of fitting and posture vectors for different orientations.*

Mouth state	Orientation number	RMSE	MAE	R^2	t/(ms)	$\varepsilon\beta$/(°)
Closed	1	1.807	1.438	0.925	24.95	–
	2	1.425	1.182	0.951	23.57	–
	3	1.459	1.161	0.944	23.41	–
	4	1.490	1.201	0.945	27.64	–
	5	1.449	1.189	0.949	24.21	–
	Average	1.526	1.234	0.943	24.76	–
Half-open	1	–	–	–	4.30	3.92
	2	–	–	–	6.32	3.74
	3	–	–	–	6.57	3.53
	4	–	–	–	4.64	2.19
	5	–	–	–	4.27	4.08
	Average	–	–	–	5.22	3.49
Open	1	–	–	–	4.44	2.68
	2	–	–	–	4.61	2.94
	3	–	–	–	4.94	3.05
	4	–	–	–	4.39	2.52
	5	–	–	–	4.45	3.17
	Average	–	–	–	4.57	2.87

Note: The data in the table are the results after down-sampling (voxel size = 1), t denotes the time consumed for point cloud preprocessing and fitting, and $\varepsilon\beta$ denotes the deviation of the predicted posture vector concerning the actual posture vector. "-" indicates metrics not applicable to specific mouth opening states. For half-open/open states, there is no correspondence between the fitted data and the original point cloud; for the closed state, mouth pose estimation is not required.

due to changes in voxel number. It also means that the adopted mouth pose estimation algorithm shows strong stability in the face of different numbers of down-sampled voxels. It is evident that the average value of $\overline{\varepsilon\beta}$ falls within the range of 2.73–4.27, suggesting that the algorithm achieves a certain level of accuracy in estimating mouth posture. Nevertheless, further improvements remain possible. The possible reason for this is that the algorithm has difficulty in accurately capturing the actual pose information when dealing with complex changes in mouth morphology or when subjected to external interference. In addition, another possible reason is the effect of the diversity and complexity of the data. If the mouth poses data has a large range of variation or there are some unpredictable situations, it may be difficult for the algorithm to estimate the pose vectors completely accurately, which results in a large average bias. Nonetheless, the range of pose estimation bias is not entirely accurate or bias-free when the meal-assisting robotic system estimates the user's mouth posture. However, the bias remains within a controllable range. During the food delivery process, this means that the robot can determine the approximate orientation of the mouth of users more accurately, thereby improving the accuracy of food delivery and reducing the occurrence of food spills or excessive deviations in the food delivery posture.

4.2. Impact of different orientations on mouth pose estimation

To further validate the performance of the proposed algorithm for mouth region fitting and posture estimation, we conducted facial image acquisition and actual pose measurement experiments on participants in different orientations. In this work, the number of down-sampled voxels was reduced to 1 in order to enhance the algorithm's processing speed while maintaining the fitting accuracy. Figure 9 and Table V show the results of the fitting and pose estimation of the mouth region of the participants in different orientations.

In Figure 9, the least-squares optimized quadratic surface algorithm effectively fits the point cloud of mouth regions in a closed mouth state under various orientations. This indicates that the algorithm is robust and can fit the closed mouth region excellently. In addition, the algorithm using the inverse

https://doi.org/10.1017/S0263574725102737 Published online by Cambridge University Press

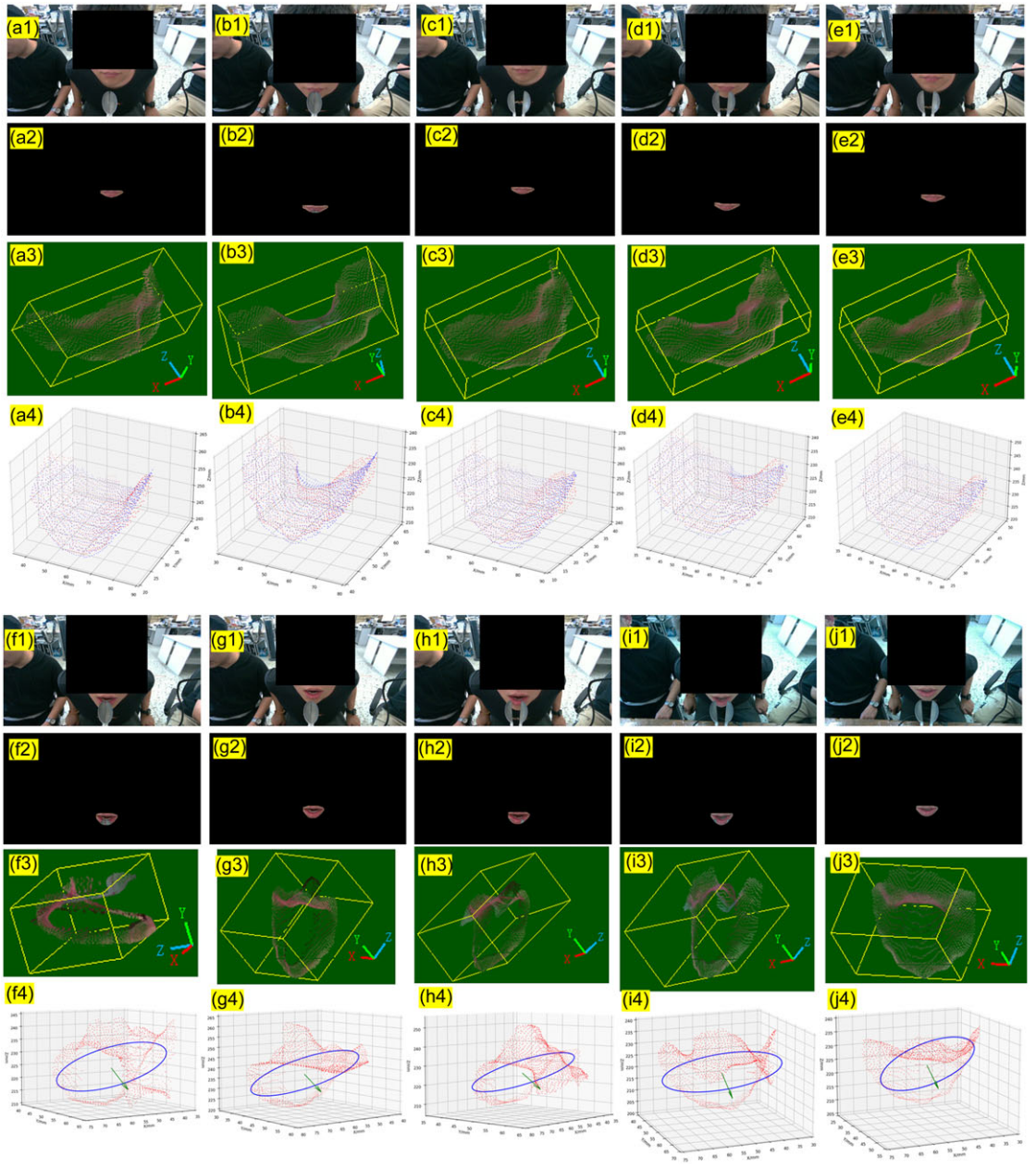


Figure 9. *The results of fitting and posture vectors in different orientations. The results of fitting and posture vectors in different orientations: (a)∼(e) indicate the results when the mouth is located in the closed state in five different orientations; (f)∼(j) indicate the results when the mouth is located in the half-open state in five different orientations; (k)∼(o) indicate the results when the mouth is located in the open state in five different orientations. Number 1 represents the RGB image acquired by the RealSence D405 depth camera; number 2 represents the RGB image obtained after segmentation of the mouth contour region; number 3 represents the CloudCompare v2.12.4 (Kyiv) visualization results; and number 4 represents the fitting and posture results obtained from the source point cloud after down-sampling (voxel size = 1). Red dots indicate the down-sampled point cloud data. Blue dots, blue closed curves, and blue spherical surfaces indicate the fitted geometries. Green arrows indicate the posture vectors for the mouth in the half-open or open state.*

https://doi.org/10.1017/S0263574725102737 Published online by Cambridge University Press

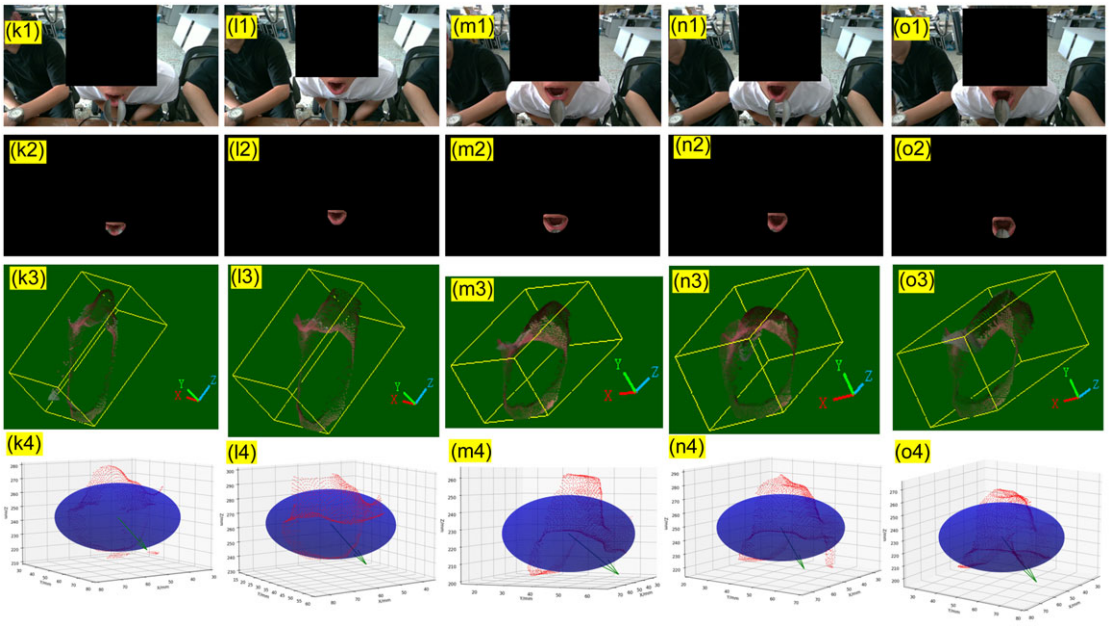


Figure 9. *(Continued)*

projection method combined with the least-squares method can fit a good spatial ellipse and spatial circle to the point cloud of the mouth region that is in half-open and open mouths under different orientations. Meanwhile, the pose estimation method can perform pose estimation of mouth opening degrees under different orientations. The results indicate that the algorithm for fitting and estimating the mouth opening degree exhibits high robustness under varying conditions.

In Table V, the quadratic surface fitting algorithms optimized using the least square method exhibit high accuracy and good fitting results for different orientations of the closed state. This may be due to the fact that the least squares method, a widely used optimization technique, estimates the best-fit curve or surface by minimizing the sum of squared errors. In the current scenario, the least square method is effective in finding the parameters of the quadratic surface so that the fitted point cloud is as close as possible to the actual point cloud. The method is robust to noise and outliers and can reduce the influence of disturbing factors in the data on the fitting results to a certain extent. In addition, the quadratic surface may be more suitable for fitting the point cloud of the mouth region in the closed mouth state. The mouth exhibits a more regular shape when closed, making it more effectively modeled by a quadratic surface compared to other types of surfaces. The quadratic surface has the flexibility to adapt to different mouth shapes and sizes by adjusting the parameters. In addition, when the number of down-sampled voxels is reduced to 1, the processing time required for fitting the point cloud of the closed mouth state is sufficient for real-time applications.

As shown in Table V, the algorithm proposed in this paper, which integrates the inverse projection method with the least-squares method for fitting ellipses and circles, demonstrates shorter processing times for point cloud fitting and pose estimation of half-open and open mouth states across various orientations, compared to the closed mouth state, and is capable of real-time processing on general hardware. In addition, different orientations have less effect on the posture estimation of the mouth opening degree, which indicates that the mouth fitting and posture estimation algorithm has strong robustness.

4.3. Impact of different mouth shapes on mouth pose estimation

The diversity of mouth morphology poses a greater challenge to the fitting and pose estimation of the mouth region. Given this, we conducted image acquisition experiments and actual pose measurement

https://doi.org/10.1017/S0263574725102737 Published online by Cambridge University Press

Table VI. *The results of fitting and posture vectors for different mouth types.*

Mouth state	Numbering	RMSE	MAE	R^2	t/(ms)	$\varepsilon\beta$/(°)
Closed	1	1.664	1.302	0.937	26.69	–
	2	1.763	1.420	0.928	32.37	–
	3	1.203	0.953	0.967	26.01	–
	4	1.456	1.157	0.950	26.81	–
	5	1.361	1.074	0.952	24.19	–
	Average	1.489	1.181	0.947	27.21	–
Half-open	1	–	–	–	4.73	3.96
	2	–	–	–	4.86	3.42
	3	–	–	–	4.70	4.24
	4	–	–	–	4.78	5.07
	5	–	–	–	4.51	3.69
	Average	–	–	–	4.71	4.08
Open	1	–	–	–	5.24	2.95
	2	–	–	–	4.91	2.79
	3	–	–	–	4.72	3.24
	4	–	–	–	5.08	3.08
	5	–	–	–	5.59	2.87
	Average	–	–	–	5.11	2.99

Note: The data in the table are the results after down-sampling (voxel size = 1), t denotes the time consumed for point cloud preprocessing and fitting, and $\varepsilon\beta$ denotes the deviation of the predicted posture vector concerning the actual posture vector. "-" indicates metrics not applicable to specific mouth opening states. For half-open/open states, there is no correspondence between the fitted data and the original point cloud; for the closed state, mouth pose estimation is not required.

experiments on the mouth morphology of different participants to validate the performance of the proposed algorithm in this paper. In this study, the number of down-sampled voxels was fixed at 1 throughout the experiments. Figure 10 and Table VI present the fitting results and mouth pose estimation outcomes for various mouth shapes.

As shown in Figure 10, under the closed mouth state, the impact of different mouth types on the least squares optimization of quadratic surfaces is minimal, indicating that the algorithm can effectively fit various mouth types. In the half-open and open states, the algorithm based on spatial ellipses and circles demonstrates improved performance in fitting different mouth shapes and estimating the degree of mouth opening. These results indicate that the proposed mouth fitting and pose estimation algorithm exhibits high stability and robustness.

As shown in Table VI, the low RMSE and MAE in the closed mouth state indicate that the algorithm for optimizing the quadratic surface using the least squares method is less sensitive to variations in mouth shape. This suggests that the deviation between the predicted and actual point clouds is relatively small. This is probably because quadratic surfaces are adaptive in fitting the mouth region in the closed mouth state, which can better capture the common features of different mouth shapes in the closed mouth state, thereby resulting in a smaller error. On the other hand, the shape of the mouth in the closed mouth state is relatively stable with a limited range of variation. By optimizing the parameters of the quadratic surface, the least square method can find a better fitting surface, enabling more accurate fitting results for different mouth shapes when the mouth is closed. In addition, the high R^2 value indicates that the algorithm effectively fits the point cloud data of the mouth region under various mouth shapes in the closed state. This suggests that the quadratic surface is capable of capturing the essential features of different mouth shapes, demonstrating the algorithm's high accuracy and reliability. This is probably because the least square method can effectively minimize the sum of squares of the errors and find the optimal parameters of the quadratic surface, which minimizes the difference between the fitting results and the actual data. Meanwhile, there may be some common geometric features of different mouth

https://doi.org/10.1017/S0263574725102737 Published online by Cambridge University Press

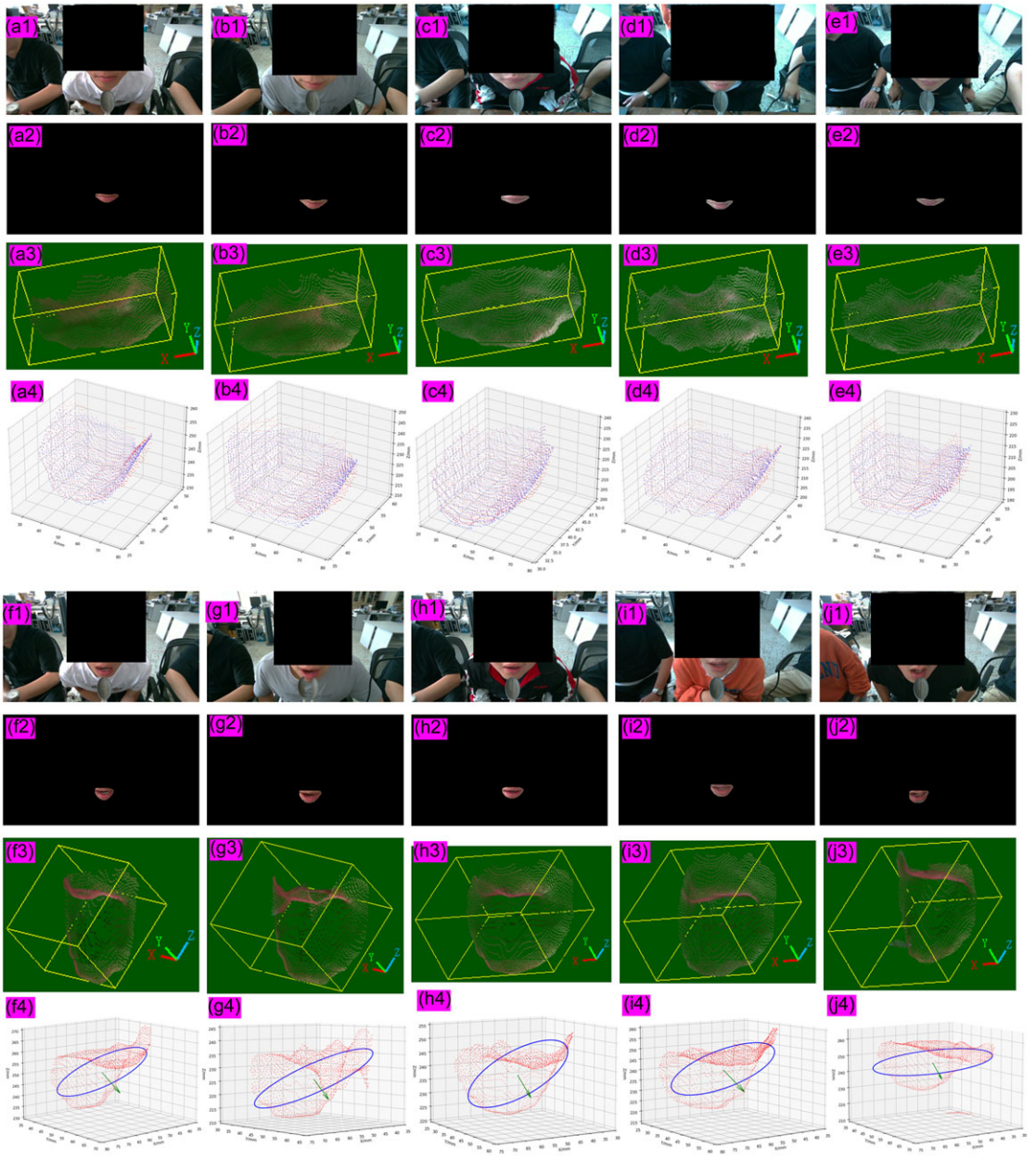


Figure 10. *The results of fitting the mouth openings and posture vectors in different mouth shapes. The results of fitting the mouth openings and posture vectors in different mouth shapes: (a)∼(e) the results when the mouths of participants with different mouth shapes were in the closed state, (f)∼(j) the results when the mouths of participants with different mouth shapes were in the half-open state, and (k)∼(o) the results when the mouths of participants with different mouth shapes were in the open state. Number 1 represents the RGB image acquired using the RealSence D405 depth camera, number 2 represents the RGB image after segmentation of the mouth contour region, number 3 represents the visualization results in CloudCompare v2.12.4 (Kyiv), and number 4 represents the fitting and posture results obtained after down-sampling (voxel size = 1) of the source point cloud. Red points indicate point cloud data after down-sampling. Blue points, blue closed curves, and blue spheres indicate the fitted geometry. Green arrows indicate the posture vectors for the mouth half-open or open state.*

https://doi.org/10.1017/S0263574725102737 Published online by Cambridge University Press

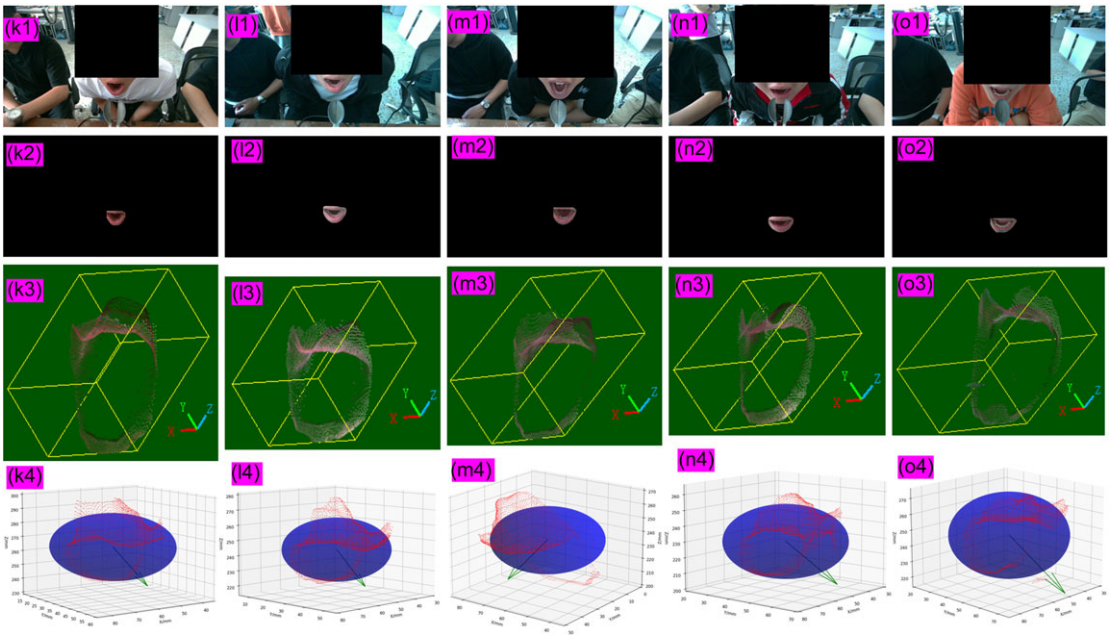


Figure 10. *(Continued)*

shapes when closing mouths, which are well captured by the quadratic surface, thereby enhancing the fitting results. Finally, the shorter processing time indicates that the algorithm has some advantages in computational efficiency. This is important for practical applications, especially in scenarios that require real-time processing. Regarding the half-open and open states, the effects of different mouth shapes on the algorithms for fitting and pose estimation are smaller in processing time and in pose estimation. This indicates that the proposed algorithm for fitting and pose estimation, which combines the inverse projection method with the least square method, has strong robustness, stability, and fast computational efficiency.

To summarize, this paper conducts comprehensive experiments and detailed analysis on mouth fitting and pose estimation with respect to various voxel numbers, orientations, and mouth types. The results demonstrate that the proposed algorithms are more accurate, robust, and computationally efficient. Furthermore, to better contextualize the contribution of the proposed FP-MODs method, we briefly compare it with representative existing pose estimation approaches (summarized in Table I) within the scope of meal-assisting robotics. For data-driven methods, while they achieve high accuracy in head/face pose estimation, they require large-scale annotated 3D mouth datasets (which are scarce in current research) and suffer from high inference latency (>20 ms on similar hardware), making them less suitable for real-time food delivery. For geometry-based methods targeting facial regions, they focus on global head pose rather than local mouth opening states, failing to capture the dynamic contour changes of closed/half-open/open mouths. In contrast, the FP-MODs method leverages task-specific geometric priors (quadratic surface/ellipse/circle) to match mouth states, avoiding reliance on large datasets and achieving a lightweight inference time (3.97–26.48 ms) that meets real-time requirements for meal-assisting robots. A comprehensive quantitative comparison with state-of-the-art methods (including data-driven and geometry-based approaches) will be conducted in future work, as noted earlier.

4.4. Limitation

The proposed algorithms have certain limitations. First, the geometric models for half-open and open mouths may deviate from actual mouth contours, leading to pose estimation errors in extreme opening

https://doi.org/10.1017/S0263574725102737 Published online by Cambridge University Press

states. Second, the algorithm is validated primarily for meal-assisting robot scenarios, and its generalizability across diverse HRI domains remains untested. Furthermore, partial occlusions, which are common in practical meal-assisting scenarios such as small spoons blocking the mouth contour or complex facial expressions like smiling while opening the mouth, reduce the effective point cloud density of the mouth region to a certain extent, increasing the average posture vector deviation compared to unobstructed cases. The current framework lacks dedicated modules (such as point cloud completion or occlusion segmentation) to mitigate such interference, which may restrict its performance in real-world feeding applications.

5. Conclusions

The objective of this study is to advance the safety and effectiveness of HRI in meal-assisting robotics by developing a sophisticated method for mouth fitting and pose estimation. Due to variations in mouth shape, head orientations, and the influence of external lighting conditions and occlusions, achieving accurate and real-time posture estimation of mouths poses significant technical challenges. Therefore, this paper proposes a new method of point cloud fitting and posture estimation applied to the mouth opening degrees. The method takes the RGB and depth images of facial regions from a single view as inputs and estimates mouth posture by combining geometric methods and robust point cloud fitting algorithms. First, this paper proposed the hypothesis that different states of mouth openings can be effectively described by distinct geometric shapes: closed mouths are modeled by spatial quadratic surfaces, half-open mouths by spatial ellipses, and fully open mouths by spatial circles. Then, based on these hypotheses, the algorithm based on least squares optimization of spatial quadratic surfaces is used to fit the point cloud data in closed mouth state. For the mouth in half-open and open states, the spatial ellipse algorithm and the spatial circle optimization algorithm that combine the inverse projection method with the least square method are used to estimate posture of mouth opening degrees, respectively. Finally, to validate the performance of the proposed algorithm, we conducted actual posture measurement experiments in multiple directions and with different mouth types. The experimental results demonstrate that the proposed algorithm for mouth posture estimation achieves higher accuracy, exhibits strong robustness, and demonstrates efficient computational performance.

This study can provide a theoretical foundation and technical support for enhancing HRI and safety in fields of intelligent robotics. It also provides important guidance and practical application value for food delivery planning of meal-assisting robotics. In future work, we will continue to explore the use of geometric models that are closer to the mouth contour to fit the point clouds of mouth regions, so as to minimize the deviation of the algorithm in estimating the mouth posture. Second, we will compare the proposed pose estimation algorithm of mouths with state-of-the-art methods in robotics to further enhance its performance. Additionally, we plan to further explore the reliability of the method in the presence of partial occlusions (including cutlery, objects, and complex facial expressions). Finally, the proposed algorithm will be applied to the task of mouth pose estimation in various domains, including healthcare robotics, to further validate its generalizability and practical applicability.

Supplementary material. The supplementary material for this article can be found at https://doi.org/10.1017/S0263574725102737.

Acknowledgments. We sincerely thank the teachers and students who participated in the facial image acquisition experiment for their support and cooperation.

Author contribution. Yuhe Fan: analysis, experiments, drafting, and revision. Lixun Zhang: funding, methods, review, and revision. Canxing Zheng: acquisition of images and experiments. Zhenhan Wang: analysis, discussion, and revision. Zekun Yang: image acquisition and experiments. Feng Xue: experiments and validation. Huaiyu Che: image acquisition and measurement. Xingyuan Wang: review and revision. All authors agree to be accountable for all aspects of the work.

https://doi.org/10.1017/S0263574725102737 Published online by Cambridge University Press

Financial Support. The research work is supported by National Key R&D Program of China under grant 2020YFC2007700 and Fundamental Research Funds for the Central Universities of China under grant 3072022CF0703.

Conflicts of Interest. The authors declare no conflicts of interest exist.

Ethical Approval. Not applicable.

References

[1] P. Daehyung, H. Yuuna and C. K. Charles, "A multimodal anomaly detector for robot-assisted feeding using an LSTM-based variational autoencoder," *IEEE Robot. Autom. Lett.* **3**(3), 1544–1551 (2018).
[2] H. Jihyeon, P. Sangin, I. Chang-Hwan and K. Laehyun, "A hybrid brain–computer interface for real-life food assist robot control," *Sensors* **21**, 4578 (2021).
[3] E. Nabil and B. Aman, "A learning from demonstration framework for implementation of a feeding task," *Encycl. Semant. Comput. Robot. Intell.* **2**(1), 1850001 (2018).
[4] K. S. Tejas, K. G. Maria and A. Graser, "Application of reinforcement learning to a robotic drinking assistant," *Robotics* **9**(1), 1–15 (2019).
[5] L. Fei, Y. Hongliu, W. Wentao and Q. Changcheng, "I-feed: A robotic platform of an assistive feeding robot for the disabled elderly population," *Technol. Health Care* **2**, 1–5 (2020).
[6] L. Fei, X. Peng and Y. Hongliu, "Robot-assisted feeding: A technical application that combines learning from demonstration and visual interaction," *Technol. Health Care* **1**, 1–6 (2020).
[7] B. Suneel, K. Ethan, C. Yuxiao, S. Siddhartha, B. Tapomayukh and S. Dorsa, "Balancing Efficiency and Comfort in Robot-Assisted Bite Transfer," **In:** *Proc. 2022 IEEE Int. Conf. Robot. Autom., Shenyang, China* (2022). arXiv:2111.11401.
[8] P. Daehyung, K. Hokeun and K. Charles, "Multimodal anomaly detection for assistive robots," *Auton. Robot.* **43**, 611–629 (2019).
[9] F. Yuhe, Z. Lixun, W. Xingyuan, W. Keyi, W. Lan, W. Zhenhan, X. Feng, Z. Jinghui and W. Chao, "Rheological thixotropy and pasting properties of food thickening gums orienting at improving food holding rate," *Appl. Rheol.* **32**, 100–121 (2022).
[10] F. Yuhe, Z. Lixun, Z. Jinghui, Z. Yunqin and W. Xingyuan, "Viscoelasticity and friction of solid foods measurement by simulating meal-assisting robot," *Int. J. Food Prop.* **25**(1), 2301–2319 (2022).
[11] F. Yuhe, Z. Lixun, Z. Canxing, W. Xingyuan, W. Keyi and Z. Jinghui, "Motion behavior of non-Newtonian fluid-solid interaction foods," *J. Food Eng.* **347**, 111448 (2023).
[12] F. Yuhe, Z. Lixun, Z. Canxing, X. Feng, W. Zhenhan, W. Xingyuan and W. Lan, "Contact forces and motion behavior of non-Newtonian fluid–solid food by coupled SPH–FEM method," *J. Food Sci.*, **88**, 2536–2556 (2023).
[13] F. Yuhe, Z. Lixun, Z. Canxing, Z. Yunqin, W. Xingyuan and Z. Jinghui, "Real-time and accurate meal detection for meal-assisting robots," *J. Food Eng.* **371**, 111996 (2024).
[14] F. Yuhe, Z. Lixun, Z. Canxing, Z. Yunqin, W. Keyi and W. Xingyuan, "Real-time and accurate model of instance segmentation of foods," *J. Real-Time Image Process* **21**, 80 (2024).
[15] F. Yuhe, Z. Lixun, Z. Canxing, W. Zhenhan, Z. Jinghui and W. Xingyuan, "Real-time and accurate detection for face and mouth openings in meal-assisting robotics," *Signal, Image Video Process* **18**, 9257–9274 (2024).
[16] F. Yuhe, Z. Lixun, Z. Caixing, W. Xingyuan, Z. Jinghui and W. Lan, "Instance segmentation of faces and mouth-opening degrees based on improved YOLOv8 method," *Multimedia Syst.* **30**, 269 (2024).
[17] M. K. Chowdary, T. N. Nguyen and D. J. Hemanth, "Deep learning-based facial emotion recognition for human–computer interaction applications," *Neural Comput. Appl.* **35**, 23311–23328 (2023).
[18] M. W. Taiba, S. G. Teddy, A. A. Q. Syed, K. Mira and A. Eliathamby, "A comprehensive review of speech emotion recognition systems," *IEEE Access* **9**, 47795–47814 (2021).
[19] W. Xuening, Q. Zhaopeng and Y. Chongchong. Multi-stage Multi-modalities Fusion of Lip, Tongue and Acoustics Information for Speech Recognition. **In:** *Proc. 6th AI Cloud Comput. Conf., Kyoto, Japan* (2023) pp. 16–18.
[20] L. Menghan, H. Bin and T. Guohui, "A comprehensive survey on 3D face recognition methods," *Eng. Appl. Artif. Intell.* **110**, 104669 (2022).
[21] F. Souheil, C. Daqing, G. Kun and X. Perry, "Lip reading sentences using deep learning with only visual cues," *IEEE Access* **8**, 215516–215530 (2020).
[22] X. Wei, C. T. Li and Y. Hu, "Robust Face Recognition Under Varying Illumination and Occlusion Considering Structured Sparsity," **In:** *Proc. 2012 Int. Conf. Digit. Image Comput. Tech. Appl., Fremantle, WA, Australia* (2012) pp. 1–7 .
[23] H. Hongyang, S. Chai, T. Jin, T. Taoling, H. Chen, D. Zhang and G. Danni, "A novel machine lip reading model," *Procedia Comput. Sci.* **199**, 1432–1437 (2022).
[24] G. Pu and H. Wang, "Review on research progress of machine lip reading," *Vis. Comput.* **39**, 3041–3057 (2023).
[25] K. G. Ethan, K. J. Rajat, N. Amal, L. Ziang, S. Tyler and B. Haya, "An Adaptable, Safe, and Portable Robot-Assisted Feeding System. **In:** *Proc. 2024 ACM/IEEE Int. Conf. Hum.-Robot Interact. March, Boulder, CO, USA* (2024) pp. 11 (14.
[26] S. Du, T. Chen, Z. Lou and Y. Wu, "A2D-LiDAR-based localization method for indoor mobile robots using correlative scan matching," *Robotica* **43**, 514–541 (2025).
[27] S. Dinct, F. Fahimit and R. Aygun, "Mirage: An O(n) time analytical solution to 3D camera pose estimation with multi-camera support," *Robotica* **35**, 2278–2296 (2017).

https://doi.org/10.1017/S0263574725102737 Published online by Cambridge University Press

[28] M. Zubair, S. Kansal and S. Mukherjee, "Vision-based pose estimation of craniocervical region: Experimental setup and saw bone-based study," *Robotica* **40**, 2031–2046 (2022).
[29] Q. Liang, M. Zhao, S. Wang and M. Chen, "Asemantic knowledge database-based localization method for UAV inspection in perceptual-degraded underground mine," *Robotica* **42**, 3480–3504 (2024).
[30] M. Zubair, S. Kansal and S. Mukherjee, "Investigating the vision-based intervertebral motion estimation of the Cadaver's craniovertebral junction," *Robotica* **41**, 2907–2914 (2023).
[31] F. Yuhe, Z. Lixun, Z. Canxing, Y. Zekun, C. Huaiyu, W. Zhenhan, X. Feng and W. Xingyuan, "Measuring posture and volume of meals for meal-assisting robotics," *J. Food Eng.* **392**, 112485 (2025).
[32] J. Whitehill and J. R. Movellan. A Discriminative Approach to Frame-By-Frame Head Pose tracking. **In:** *Proc. IEEE Conf. Autom. Face Gesture Recognit.* (2008) pp. 1–7.
[33] G. Fanelli, J. Gall and L. Van. Real time head pose estimation with random regression forests. **In:** *Proc. IEEE Conf. Comput. Vis. Pattern Recognit*, 2011) pp. 617–624.
[34] C. Luo, J. Zhang, J. Yu, C. W. Chen and S. Wang, "Real-time head pose estimation and face modeling from a depth image," *IEEE Trans. Multimed.* **21**(10), 2473–2481 (2019).
[35] A. Saeed and A. Al-Hamadi. Boosted human head pose estimation using kinect camera. **In:** *Proc. IEEE Int. Conf. Image Process* (2015) pp. 1752–1756.
[36] T. Baltrušaitis, P. Robinson and L. P. Morency. 3d Constrained Local Model for Rigid and Non-rigid Facial Tracking. **In:** *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* (2012) pp. 2610–2617.
[37] B. Ahn, J. Park and I. S. Kweon. Real-Time Head Orientation From a Monocular Camera Using Deep Neural Network. **In:** *Proc. Asian Conf. Comput. Vis.* (2014) pp. 82–96.
[38] B. Ahn, D. G. Choi, J. Park and I. S. Kweon, "Real-time head pose estimation using multi-task deep neural network," *Robot. Auton. Syst.* **103**, 1–12 (2018).
[39] N. Ruiz, E. Chong and J. M. Rehg. Fine-Grained Head Pose Estimation Without Keypoints. **In:** *Proc. IEEE Conf. Comput. Vis. Pattern Recognit* (2018) pp. 2074–2083.
[40] G. Borghi, M. Fabbri, R. Vezzani, S. Calderara and R. Cucchiara, "Face-from-depth for head pose estimation on depth images," *IEEE Trans. Pattern Anal. Mach. Intell.* **42**(3), 596–609 (2020).
[41] M. Martin, F. Van and R. Stiefelhagen. Real Time Head Model Creation and Head Pose Estimation on Consumer Depth Cameras. **In:** *Proc. Int. Conf. 3D Vis.* (2014) pp. 641–648.
[42] R. S. Ghiass, O. Arandjelovi and D. Laurendeau. Highly Accurate and Fully Automatic Head Pose Estimation from a Low Quality Consumer-Level RGB-d Sensor. **In:** *Proc. Workshop Comput. Models Soc. Interact. Hum.-Comput.-Media Commun.* (2015) pp. 25–34.
[43] S. Li, K. N. Ngan, R. Paramesran and L. Sheng, "Real-time head pose tracking with online face template reconstruction," *IEEE Trans. Pattern Anal. Mach. Intell.* **38**(9), 1922–1928 (2015).
[44] X. Yuanquan, J. Cheolkon and C. Yakun, "Head pose estimation using deep neural networks and 3D point clouds," *Pattern Recognit.* **121**, 108210 (2022).
[45] S. Mo and X. Miao. Osgg-Net: One-Step Graph Generation Network for Unbiased Head Pose Estimation. **In:** *Proc. 29th ACM Int. Conf. Multimedia* (2021) pp. 2465–2473.
[46] C. Y. Wu, Q. Xu and U. Neumann, "Synergy Between 3dmm and 3d Landmarks for Accurate 3d Facial Geometry," **In:** *Proc. 2021 Int. Conf. 3D Vis.* (2021).453 (463.
[47] M. Xin, S. Mo and Y. Lin. "Eva-gcn: Head Pose Estimation Based on Graph Convolutional Networks," **In:** *Proc. IEEE Conf. Comput. Vis. Pattern Recognit* (2021) pp. 1462–1471.
[48] W. Chai, J. Chen, J. Wang, S. Velipasalar, A. Venkatachalapathy and Y. Adu-Gyamfi, "Driver head pose detection from naturalistic driving data," *IEEE Trans. Intell. Transp. Syst.* **24**(9), 9368–9377 (2023).
[49] Y. Kao, B. Pan, M. Xu, J. Lyu, X. Zhu and Y. Chang, "Towards 3D face reconstruction in perspective projection: Estimating 6DoF face pose from monocular image," *IEEE Trans. Image Process* **32**, 3080–3091 (2023).
[50] Z. Cao, Z. Chu, D. Liu and Y. Chen. "A Vector-Based Representation to Enhance Head Pose Estimation," **In:** *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.* (2021) pp. 1188–1197.
[51] T. Hempel, A. A. Abdelrahman and A. Al-Hamadi, "6D rotation representation for unconstrained head pose estimation," *Proc, IEEE Int. Conf. Image Process* **2022**, 2496–2500 (2022).
[52] Y. Xu, C. Jung and Y. Chang, "Head pose estimation using deep neural networks and 3D point clouds," *Pattern Recognit.* **121**, 108210 (2022).
[53] H. Zhou, F. Jiang and H. Lu, "A simple baseline for direct 2D multi-person head pose estimation with full-range angles (2023), arXiv preprint, arXiv:2302.01110.
[54] A. Redhwan, S. Hyunsoo and L. Sungon, "Real-time 6DoF full-range marker-less head pose estimation," *Expert Syst. Appl.* **239**, 122293 (2024).
[55] Intel (2022) https://www.intelrealsense.com/wp-content/uploads/2020/06/Intel-RealSense-D400-Series-Datasheet-June-2020.pdf (Online accessed October 2022), 2022.
[56] C. Martin, S. Daniel and L. Sungkil, "Automated outdoor depth-map generation and alignment," *Comput. Graph.* **74**, 109–118 (2018).
[57] L. Guichao, T. Yunchao, Z. Xiangjun and W. Chenglin, "Three-dimensional reconstruction of guava fruits and branches using instance segmentation and geometry analysis," *Comput. Electron. Agric.* **184**, 106107 (2021).
[58] G. Lin, Y. Tang, X. Zou, J. Xiong and J. Li, "Guava detection and pose estimation using a low-cost RGB-D sensor in the field," *Sensors* **19**(2), 428 (2019).

https://doi.org/10.1017/S0263574725102737 Published online by Cambridge University Press

[59] C. Zong and H. Wang, "An improved 3D point cloud instance segmentation method for overhead catenary height detection," *Comput. Electr. Eng.* **98**, 107685 (2022).
[60] F. Yuhe, Z. Lixun, Z. Canxing, W. Zhenhan, C. Huaiyu and W. Xingyuan, "Study on adaptive fuzzy force control based on food rheology properties," *J. Food Eng.* **406**, 112818 (2026).
[61] R. Schnabel, R. Wahl and R. Klein, "Efficient RANSAC for point-cloud shape detection," *Comput. Graph. Forum.* **26**(2), 214–226 (2007).

Cite this article: Y. Fan, L. Zhang, C. Zheng, Z. Wang, Z. Yang, F. Xue, H. Che and X. Wang (2025). "Real-time mouth posture estimation for meal-assisting robots", Robotica **43**, 4088–4115. https://doi.org/10.1017/S0263574725102737

https://doi.org/10.1017/S0263574725102737 Published online by Cambridge University Press