Touchless Head-Control (THC): Head Gesture Recognition for Cursor and Orientation Control

Wahyu Rahmaniar, Alfian Ma'arif, *Member, IEEE*, and Ting-Lan Lin, *Member, IEEE*

***Abstract*— The touchless techniques in human-computer interaction (HCI) can effectively expand computer access capabilities for disabled people. This paper presents Touchless Head-Control (THC), an assistive system method for computer cursor control based on head pose captured with an RGB camera. Our work aimed to replace the standard cursor control using a device on the user's head. The convolutional neural networks with predicted fine-grained feature maps and binned classification were applied to estimate the head pose angles. The mouse pointer or cursor is moved to actual locations on the screen based on head movement (yaw and pitch) and the center position of the face. Head tilt to the right or left (roll) to control the mouse button. In addition, the proposed method can be used to simulate the movement of the robot or joystick using the head to control objects within three degrees of freedom (DOF). Various participants were involved in the interaction design evaluation, in which target selection accuracy, travel time, and path efficiency were measured. This technology allows people with limited motor skills to easily control a PC cursor and 3D object orientation without the use of additional equipment or sensors.**

***Index Terms*— Assistive technology, head pose, human-computer interaction, mouse control, orientation control.**

I. Introduction

HUMAN-COMPUTER interaction (HCI) has advanced recently, making computers more accessible to persons with the restricted motor ability [1], [2]. Previous studies have developed various devices to assist disabled people using gestures and other non-contact techniques [3], [4]. Most HCI activities involve user interaction without utilizing an assistant or assistive device for persons with limited exercise skills, such as spinal infringement and limb paralysis. HCI provides opportunities for disabled persons to produce computer work by facilitating access to computers, such as cursor control. An alternative computer mouse with a gyroscope as a motion sensor has been developed for people with movement disorders [5]. In addition, hands-free interaction with HCI can help people with impairments integrate into the workforce, such as through orientation controls that allow impaired people to participate in daily activities [6]. A head-mounted inertial interface was employed in [7], for patients with cerebral palsy. However, these HCI studies use sensors and devices worn by the user, making the system less flexible and costly.

Recently, a head motion-based interface for HCI control applications has been proposed. Head motion was captured using inertial measurement units (IMUs) [8] and vision-based [9] for wheelchair control. The user can operate more complicated systems with an interaction design that can manage more than one degree of freedom (DOF) [10], and the user can operate more complex systems. A dedicated helmet with a head-mount controller has been designed in place of a joystick [11]. Head controllers in other studies have been used to assist surgeons in controlling robots during surgery [12]. However, HCI control using the head still struggled to verify the accuracy of the head pose for more accurate control.

This paper proposes a Touchless Head-Control (THC) that uses head posture estimation and facial position to control the cursor on a PC and 3D objects in three DOFs (x-, y-, and z-axes). Our contribution is a new method to improve the performance of head pose predictions using deep convolutional neural networks (CNNs). We built an efficient CNN architecture that requires less pre-processing without keypoints and landmarks. The mouse cursor could be moved to a precise target location by tracking the position of the head/face, which moves up-down (nodding) and left-right (rotation). The mouse buttons were controlled by bending the head left or right. Furthermore, the interactive user interface can be used with minimal response lag time. The proposed system is designed to be simple and convenient for persons with disabilities to use computers and operate objects.

II. Related Works

A. Mouse Head-Control

One of the most critical aspects of the HCI system is the accuracy of head posture detection. Changes in lighting

Manuscript received 4 April 2022; revised 16 June 2022; accepted 26 June 2022. Date of publication 30 June 2022; date of current version 14 July 2022. This work was supported by the Ministry of Science and Technology, Taiwan, under Grant MOST 110-2221-E-027-044-MY3. *(Corresponding author: Ting-Lan Lin.)*

Wahyu Rahmaniar is with the Department of Electronic Engineering, National Taipei University of Technology, Taipei 10608, Taiwan (e-mail: wahyu@ntut.edu.tw).

Alfian Ma'arif is with the Department of Electrical Engineering, Universitas Ahmad Dahlan, Yogyakarta 55191, Indonesia (e-mail: alfianmaarif@ee.uad.ac.id).

Ting-Lan Lin is with the Department of Electronic Engineering, National Taipei University of Technology, Taipei 10608, Taiwan, and also adjunctly with the Department of Electronic Engineering, Chung Yuan Christian University, Taoyuan 320314, Taiwan (e-mail: tinglan@ntut.edu.tw).

Digital Object Identifier 10.1109/TNSRE.2022.3187472

conditions, varying facial forms, and eye sizes create significant issues that must be addressed in real-world applications when employing computer vision for head tracking and eye condition monitoring. A method in [13] captures a reference point in the center of the user's head using the camera. The computer cursor coordinates are converted from the processed image of the head position. Two cameras matched to the glasses are used to analyze the position of the screen cursor in [14]. The mouse cursor advances to the desired position, determined by the user's facial orientation. Head tilt and electromyography (EMG) integration were used to move the cursor control [1]. Computer vision speeds up target selection with head tilt, while EMG improves mean free path efficiency and target selection accuracy. On the other hand, ambient conditions can affect the accuracy of head motion detection, and the use of additional sensors reduces the flexibility of the system. A head mouse control system for disabled people with spinal cord injury was proposed in [15]. CNN identifies head movements based on eye, mouth, and nose detection. However, wide-angle head movements can cause these areas of the face to be undetectable, resulting in impaired and inaccurate cursor control. Moreover, this study does not demonstrate the precision of the real-time application approach for flexible cursor control.

Previous works have used computer vision or physiological signals to detect head movements for cursor control. Optical sensors have a limited resolution, expensive, and are not suitable for long-term usage in sensor-based solutions. Furthermore, these sensors only provide limited data to deal with unanticipated circumstances in the field. Camera-based solutions are less expensive and more adaptable to future data modifications. Our goal is to create a cursor control interface that uses accurate and reliable head movements with a single camera. We proposed a method to estimate head pose using CNN to achieve this goal accurately. Our algorithms are designed to manage various field situations, such as changes in light illumination and the user's head position. The system is more flexible and cost-effective because no additional sensors or other devices are required. The proposed method is capable of fast-performing mouse-like button clicks. System performance was assessed by measuring target selection accuracy, travel time, and path efficiency.

B. Orientation Head-Control

Several studies have been conducted on head posture detection or tracking head movements for robotics and control. Roll (lateral flexion), pitch (flexion/extension), and yaw (rotation) are three degrees of freedom (DOFs) that can be utilized as an orientation controller for 3D objects using head movements. A motion sensor was used in [16] to detect headgears in an analytical approach. The classification of head movements is based on comparing calculated values with defined criteria. A single IMU is secured in a hairband control used by an auxiliary robot in [17]. Incorporating three accelerometers, three gyroscopes, and three magnetometer sensor data in a nine-axis IMUs enables reliable motion measurement. Sensor orientation is determined as output by integrated sensor fusion. However, long-term sensor usage is ineffective regarding user flexibility and comfort. In addition, the cost of the device hinders its development from adapting to changes in the field.

A video-based approach to estimating head pose has been used for orientation control. A method in [18] uses a depth sensor camera to recognize head gestures. The system uses depth data obtained from sensors to detect facial feature points and represent human head movements. Depth cameras only provide accurate distance information for facial areas at close range. In addition, the acquired depth data is affected by the objects around the user's face, which reduces the accuracy of the information. A previous study in [12] proposed a system that allows surgeons to control the endoscopic camera without an assistant. The camera can be controlled by head movement allowing the surgeon to operate the instrument by hand. The system is based on a flexible endoscope which gives the surgeon more freedom to operate the instrument than a rigid endoscope. However, prolonged sensor use on the surgeon's head can be offensive to the user. Sensor performance is also affected by head movement issues. Nevertheless, these studies show that head motion control has proven to be a cutting-edge technological breakthrough in various fields. However, multiple sensors remain the primary option due to accurate head pose detection limitations. This paper combines the CNN head pose estimation with the Kalman filter to perform a precise head movement control. ‘

III. Proposed Method

This section describes the proposed method for THC, as shown in Fig. 1. First, face detection using CNN was applied to obtain an accurate face area as an input image for the next stage. Then, the pixels in the eye area were examined to determine whether the face was moving or stationary. If the detected face was defined as moving, the head poses angle, such as yaw, pitch, and roll, are calculated. Facial movement is determined to avoid unwanted cursor movements due to changes in lighting or head pose estimation errors. As face detection and head pose calculations are performed on each frame, sometimes the face bounding box changes its pixel slightly, or the head pose changes suddenly. This situation can cause the cursor to move according to unexpected changes, disturbing the user's comfort. The estimated head pose and facial area control cursor movement and orientation simulation at the x and y coordinates. The trajectory obtained from the control was smoothed with a Kalman filter.

A. Face Detection

YOLOv4-based object detection framework [19] was used to detect faces. The WIDER-Face dataset used to train this architecture contains 32,203 images and identifies 393,703 faces with large scale, pose, and occlusion heterogeneity. The input image was divided into a grid of cells with the YOLOv4 model. Each of these grids was in charge of describing a different object. The confidence score was

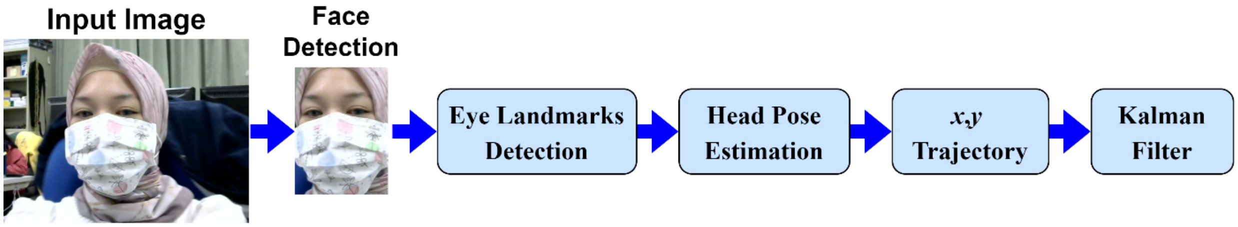


Fig. 1. Touchless head-control framework.

calculated with the bounding box for each grid cell. This approach separates the data into grids and uses the grid to identify object features. The observed features with high reliability in the surrounding cells were combined in one place to achieve model performance. The detected faces were cropped around the bounding box to lower computational costs. Since the facial area was the primary input for determining head pose, accurate face detection was required. Low face detection accuracy indicated that the model was unsure of the sort of item being spotted. It could be mistaken for a false positive, which reduces system performance. As a result, the correct margins on the face bounding box could improve the accuracy of calculating the angle of the head pose. Based on the head pose estimation in [20], YOLOv4 could detect face bounding boxes more precisely than other methods, such as Haar-cascade and SSD-MobileNetV2. YOLOv4 could still detect facial areas precisely for various difficult positions even though the face is covered. Determining the exact area of the face affects the accurate calculation of the angle of the head pose.

B. Eyes Detection

An approach that utilizes a cascade of regressors was used to locate the eyes landmark [21]. The regressor generated predictions based on variables like pixel intensity values generated from the index relative to the current shape estimation as the cascade keypoint. As the cascade advances, this adds some geometric invariance to the process, making it more assured that the exact semantic location on the face has been indexed. The initial shape can be an averaged shape of the training data, centered and scaled using the bounding box output of a generic face detector. Each regressor was learned using the gradient boosting tree method. The residuals, which correspond to the gradient of the squared error loss function assessed for each training sample, were computed in the innermost loop. At each node, the thresholding of the difference in intensity values between two pixels is employed to determine the decision. The closer pair of pixels were selected. Exponential prior was applied to the distance between the pixels in the split. Ocular landmarks number 27 (center of the eye), 36 (left edge of the left eye), and 45 (right edge of the right eye) were identified, and their center points were tracked to quantify facial movement, as shown in Fig. 2.

C. Head Pose Estimation

The input images are first employed in the backbone network based on ResNet-101. A bottleneck block with a layer extension was used in this backbone. With an additional block, the 101-layers construction enhances precision. The face presented in the input image is converted to grayscale with a size of 64 × 64 pixels. For feature map classification, fine-grained structures were mapped to obtain a representative feature set. The network was trained on 300W-LP [22], a synthetically extended dataset, as well as a re-annotated in-the-wild 2D landmark dataset. The poses in the dataset were precisely labeled according to head rotation to generate head pose annotations. The proposed head pose estimation used RGB images rather than depth information for individual color frames to obtain pixel-level intensities. This head pose detection is based on the method in [20] by simplifying the backbone and feature maps.

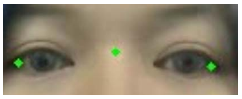


Fig. 2. Selected points (green dots) for eyes detection.

A set of training face images is given as $X = \{x_i\}$ where $i = 1, 2, \ldots, n$, y_i is the pose vector for each image x_i, n is the total number of training images, andx_i contains 3D vectors corresponding to the Euler angles, *i.e.*, yaw, pitch, roll. The function f desired is used to find a predicted head pose $\hat{y} = f\,\{x\}$ that as closely as feasible fits the expected head pose y for a given image x. The features were processed through 2048 × 1 fully-connected layers (FC-layers) that transferred the result from the feature maps selection to a single continuous shape, as shown in Fig. 3. Each orientation angle was classified using a softmax classifier and cross-entropy loss. Between the ground-truth label and the predicted value of the softmax output, a mean squared error (MSE) regression loss was applied. The final training objective weights for each angle are calculated by adding the two losses (cross-entropy and MSE). As a result, the training process could learn more precise position angles.

The classification was divided into several scales to ensure accurate predictions at different classification scales. Then, the classification was refined to improve the accuracy of the regression. Each FC-layer represented a distinct classification scale and measured cross-entropy loss independently. The smoothest classification result was utilized in regression integration to calculate expectations and losses. Then, the Euler angles output was regressed with a 3 × 1 matrix and normalized to the range of $[-1, 1]$.

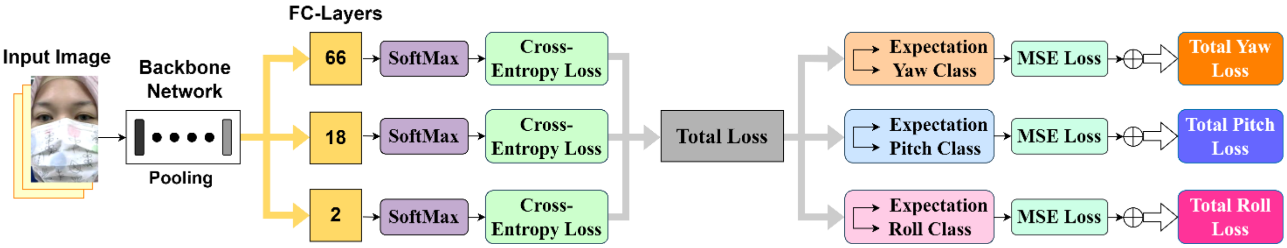


Fig. 3. Head pose estimation architecture.

Tilting the head to the side or wearing a face-covering makes facial features detection more challenging. A fine-grained attention module was applied to solve this problem, attempting to classify each pixel in the feature map. The obtained losses were calculated in several stages and were added up. The fine-grained attention module subsequently classified important face features into different intensity levels. All feature maps were flattened into a 2D matrix that contains all pixels from all feature maps at all stages. Then, the size of feature maps was reduced using average pooling. Convolution was used to transform the combined feature maps at each stage with one stride. The fine-grained structure mapping was incorporated with the attention maps.

Three parallelograms (2, 18, and 66) were used to bin the output angles, then processed by softmax layers to generate bin probabilities for each angle. Each Euler angle had a combined loss and used the previous convolution layer. A total loss consisted of regression loss and multiple classification loss. Each loss includes classification and regression of binned poses for yaw, pitch, and roll separately. Cross-entropy loss function and MSE were used to estimate the error. Since each Euler angle has one cross-entropy loss, the three propagated signals were sent back to the initial stage of the network to improve model training. MSE loss was applied to the estimated output of each angle. Each Euler angle covered a pose range of $[-90°, 90°]$, dividing the prediction class into 181. The image would be rejected if the observed angle was out of these ranges. Over the training samples $i = 1, 2, \ldots, n$, the regression loss weights were diverse to make the predicted angle y_i near the expected angle possible. Then, the mean absolute error (MAE) was minimized by the MSE loss, which can be calculated as follows

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{1}$$

The loss function used to optimize the fine-grained attention maps was cross-entropy. The weighted sum of the intensity map can be used to calculate the final loss function as follows

$$Loss = \alpha MSE(y, \hat{y}) + \sum_{i=1}^{N} CE(y_i, \hat{y}_i) \tag{2}$$

where α is the weight balancing of the MSE loss set to 2 and CE is the cross-entropy loss functions set to $N = 5$.

D. Kalman Filter

The obtained head pose angles (yaw, pitch, and roll) and face bounding box were used to calculate the trajectory on the x-axis T_x, y-axis T_y, and the path angle τ. The trajectory was corrected using the Kalman filter to provide a new set of transformations for each head movement. The Kalman filter has two main components: prediction and measurement correction.

At the prediction step, the Kalman gain can be calculated by dividing the error covariance ε by the process noise covariance β as follows

$$K(t) = \frac{\varepsilon(t)}{\varepsilon(t) + \beta} \tag{3}$$

where $s(t) = [T_x(t), T_y(t), \tau(t)]$ is the trajectory at the prediction step and the initial state denoted by $s(0) = [0, 0, 0]$. The updated error covariance can be determined by $\varepsilon(t) = \varepsilon(t-1) + \beta$ where $\varepsilon(0) = [1, 1, 1]$ is the initial error covariance.

At the measurement step, the Kalman gain can be computed by

$$\hat{K}(t) = \frac{\hat{\varepsilon}(t)}{\hat{\varepsilon}(t) + \hat{\beta}} \tag{4}$$

where $\hat{\beta}$ is measurement noise covariance. The error covariance is adjusted by $\hat{\varepsilon}(t) = (1 - K(t))\,\varepsilon(t)$. The trajectory is compensated by the accumulated measurement $\delta(t)$ as follows

$$\hat{s}(t) = s(t) + \hat{K}(t)\,(\delta(t) - s(t)) \tag{5}$$

The new state from the obtained trajectory can be defined as $\hat{s}(t) = \left[\hat{T}_x(t), \hat{T}_y(t), \hat{\tau}(t)\right]$. The new trajectory can be obtained through the difference between the current state with the accumulation of the measurement as follows

$$\begin{aligned}&\left[\bar{T}_x(t), \bar{T}_y(t), \bar{\tau}(t)\right]\\ &\quad = \left[T_x(t), T_y(t), \tau(t)\right]\left[\sigma_x(t), \sigma_y(t), \sigma_\tau(t)\right]\end{aligned} \tag{6}$$

where $\sigma_x(t) = \hat{T}_x(t) - \delta_x(t)$, $\sigma_y(t) = \hat{T}_y(t) - \delta_y(t)$, and $\sigma_\tau(t) = \hat{\tau}(t) - \delta_\tau(t)$. The trajectory gained from each head movement can be accumulated by

$$\begin{aligned}\delta(t) &= \sum_{n=1}^{t}\left[\left(\bar{T}_x(n) + T_x(t)\right), \left(\bar{T}_y(n) + T_y(t)\right), \left(\bar{\tau}(n) + \tau(t)\right)\right]\\ &= \left[\delta_x(t), \delta_y(t), \delta_\tau(t)\right]\end{aligned} \tag{7}$$

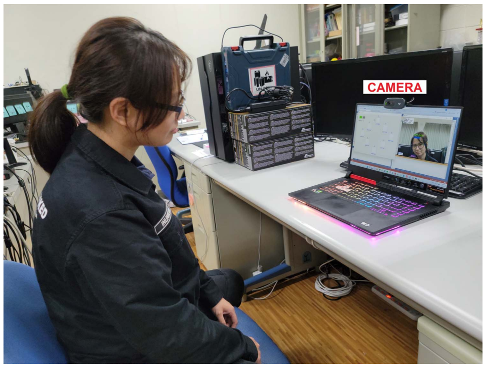


Fig. 4. Experimental design for THC.

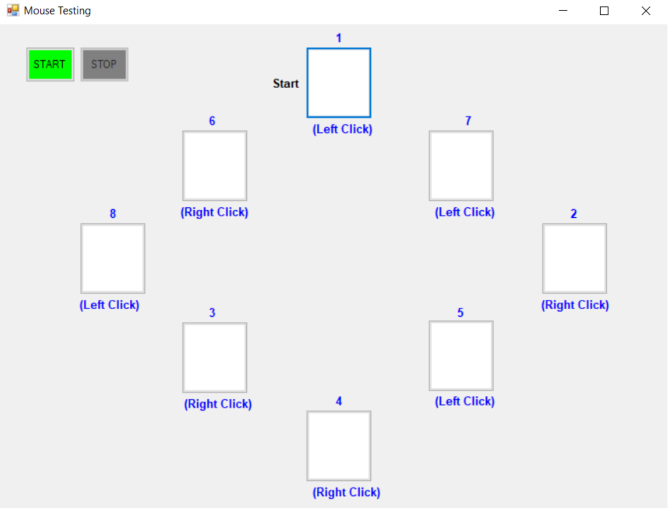


Fig. 5. Cursor control user interface design.

IV. Experimental Design

Face detection and head pose estimation were trained on an Intel Core-i7 processor and an RTX2060 GPU with 16 GB of RAM. The experimental programming software includes Python and Visual Basic Net for User Interface (UI), tested on the CPU. The objectives used for this experiment were cursor and orientation control, which includes the trajectory of continuous head movements to achieve a specific task. Participants conducted the experiment in front of the camera without additional tools or sensors, as shown in Fig. 4. The task accuracy, path efficiency, and completion time during the targeting task were recorded for method performance analysis. We asked the participants to do the task well under the given instructions. The head must be moved in the proper DOF to produce the desired range of head movements. The controls were designed to be as efficient as possible, allowing the user to make the possible minor movement utilizing only the skull and neck muscles while avoiding straining the neck muscles. During the experiment, participants were requested to maintain a fixed body position from the neck.

A. Mouse Head-Control

1) Procedure: Fig. 5 shows the UI to demonstrate the cursor control performance. The cursor can be moved vertically by moving the head up/down (extension/flexion) and horizontally by moving the head to the left/right (rotation). The left/right mouse click function was performed by bending left/right (lateral bending). Participants were instructed to move the cursor from box 1 to box 8, then click left or right according to the command in each box. The UI has a size of 1023 × 726 pixels, with each box having a size of 100 × 100. The distance between boxes is 162 and 128 pixels on the x- and y-axes. If the left or right-click is successful, the box on the UI will be green or red, respectively. Participants could reposition their heads to their regular positions and bend to make mouse clicks more efficient. Participants were requested to move the cursor pointer path displayed in the UI as efficiently as possible. Participants were asked to perform several experiments with variations in the distance with the camera and the initial head-base position. Experiments were also carried out under several different ambient lighting conditions to prove the reliability of our method. The laptop's brightness was selected according to the level of comfort in the room.

2) Algorithm: Participants can move the cursor by tilting their heads toward the desired cursor movement. The midpoint $\left(\Delta_x, \Delta_y\right)$ between the three detected ocular landmarks is calculated to determine whether the head is moving or not. Cursor coordinates (x, y) are calculated based on the center position $\left(f_x, f_y\right)$ of the detected face with the Euler angles of the head pose $[\psi, \theta, \phi]$, as described in Algorithm 1. At the commencement of the system, step 1 is completed for calibration. The midpoint in the bounding box of the user's face is examined for a few seconds before being saved as the face's initial location. In the next step, the center coordinates of the detected ocular landmarks are compared with their initial position to determine the state of the face as moving or stationary. Nodding movements (extension and flexion) are associated with moving the cursor up and down, and the rotating movement is associated with moving the cursor left and right. The bending movement is more efficiently used to perform the mouse click function than detecting the blink of an eye. This combination of gestures allows 180° of movement in a two-dimensional display surface. A head pose and facial movement combination are used to obtain the appropriate cursor position in the UI. Participants can observe the face detection and head pose calculations performed on the image generated by the camera and the calculation results to determine the cursor's location, which is depicted with a black line on the UI.

3) Performance Metrics: Target selection accuracy, travel time between targets, and path efficiency were averaged against each measurement result for each participant. Travel time was the time (seconds) the user took to navigate and select each box in sequence. The target accuracy was determined by the number of times the participants made an error in the experiment while choosing the specified box. The initial accuracy value was 10, which would be reduced by the number of times the participant failed to select the following box. After reaching a box, the participant had to perform a right or left click command according to the instructions on each box. If the participant did not pass the boxes in sequence or move the cursor before executing the click command, accuracy

Algorithm 1 Touchless Head-Control

```
Input: - Euler angles: Yaw $\psi$, Pitch $\theta$, Roll $\phi$
       - Ocular landmarks center location $(\Delta_x, \Delta_y)$
       - Face center location $(f_x, f_y)$
Output: Trajectory $(x, y)$
Step 1: $n = 0$
    $\bar{f}_x = \text{update}\ (f_x)$
    $\bar{f}_y = \text{update}\ (f_y)$
    if $|\bar{f}_x - f_x| < 5$ and $|\bar{f}_y - f_y| < 5$
    $n = n + 1$
    if $n > 30$
      Cursor control is ready
      $f_{x0} = f_x$
      $f_{y0} = f_y$
      $c_x = 360$
      $c_y = 100$
Step 2: $\bar{\Delta}_x = \text{update}\ (\Delta_x)$
    $\bar{\Delta}_y = \text{update}\ (\Delta_y)$
    if $|\bar{\Delta}_x - \Delta_x| < 5$ and $|\bar{\Delta}_y - \Delta_y| < 5$
      Face is move
      Calculate $\psi, \theta, \phi$
Cursor control:
$x = c_x + (\psi \times 6) + ((\bar{f}_x - f_{x0}) \times 5)$
$y = c_y + (\theta \times 6) + ((\bar{f}_y - f_{y0}) \times 5)$
if $\phi \le -10$
Left click
if $\phi > 10$
Right click
Orientation control:
$x = (\psi + (\bar{f}_x - f_{x0})) \times 2$
$y = (\theta + (\bar{f}_y - f_{y0})) \times 2$
if $\phi \le -20$
Open gripper
if $\phi > 20$
Close gripper
```

points would be deducted by 1. The path efficiency was calculated as the straightness and length measurement of the path between targets. The origin (x_i, y_i) was determined as the efficient coordinate obtained from the results of several experiments. When selecting the current target, the coordinates corresponding to the cursor location were defined as (x_j, y_j). The angle of every 50 points of the cursor coordinates is calculated by

$$\tau_j = \tan^{-1}\left(\frac{y_j}{x_j}\right) \tag{8}$$

Thus, the path accuracy can be calculated as follows

$$\text{Path\_accuracy} = 100\% - \left(\frac{\text{abs}|\tau_i - \tau_j|}{\tau_j} \times 100\%\right) \tag{9}$$

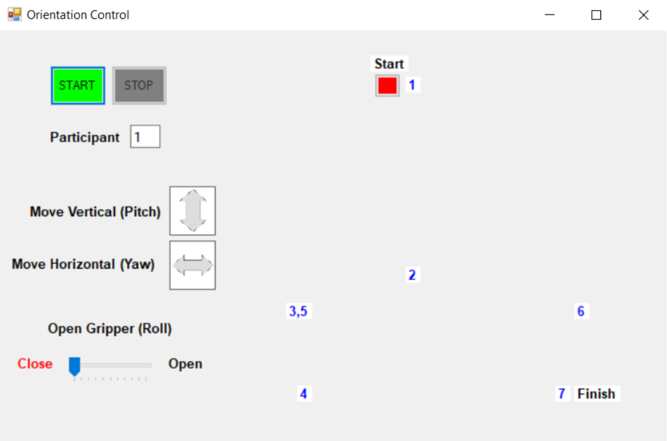


Fig. 6. Orientation control user interface design.

The mean (M) and standard deviation (SD) testing measurements were analyzed for each outcome, *i.e.*, target selection accuracy, travel time, and path efficiency, to assess the impact of various lighting conditions and the distance between participant and camera head-base position.

B. Orientation Head-Control

1) Procedure: The head's orientation was determined by flexion/extension in the sagittal plane, lateral flexion in the coronal plane, and axial rotation in the cross-section. Participants were asked to move the red box on the orientation control UI, as shown in Fig. 6, to simulate the robot's movement. In this case, it acted like a robotic arm. Participants were instructed to move the box sequentially from points 1 to 7. From point 1, participants could perform pitch or flexion movements to reach point 2. Then, they could move their head slightly to the left or rotate their heads to reach point 3. At point 3, the robot was simulated to be at the stopping point to pick up goods at point 4. When it reached point 4, the robot was simulated to make the gripper open (bend to the right) and close (bend to the left). The robot was then simulated back to point 3 (in this case, the fifth position), moving to point 6 on the right. Then the gripper movement was simulated to place objects at point 7. Each participant was asked to perform movements sequentially while paying attention to the movement path of the box in the UI. In the targeting task, the participants were asked to achieve the desired orientation in the shortest possible time.

2) Algorithm: Participants could move the box by tilting their heads toward the desired movement. Nodding movements (extension and flexion) were associated with a simulation of a robotic arm moving up and down. In contrast, rotating movements were associated with a simulation of a robotic arm moving left and right. For gripper simulation, a bending movement was used to open and close. A combination of head poses and facial movements was used to replicate robotic motions accurately, as explained in Algorithm 1. On the user interface, participants could observe the path of movement.

3) Performance Metrics: Similar to the mouse head-control, the value of each measurement result for each participant was averaged against the results of the target selection accuracy, travel time between targets, and path efficiency measurements. Travel time was the time (seconds) the user took to navigate each box to predetermined points in sequence. The initial accuracy value was 10, which would be reduced by the number of times the participant failed to select the following box. Path accuracy compared the straightness of the path taken by the participant (x_j, y_j) with the predetermined efficient path (x_i, y_i), which can be calculated in Eq. 8 and 9.

V. Results

The experiment was conducted using an RGB camera with a resolution of 640 × 480 on an Intel Core-i7 CPU for performance testing. Our method in real applications achieves a computation time of around 11 frames per second (fps). The head pose estimation was performed on benchmark datasets with an average prediction error of 5.09°, 4.35°, and 3.32° for the AFLW2000, AFLW, and BIWI datasets.

The participant's position varied between 50cm and 100cm from the camera. The experiment evaluated the cursor and orientation control using head movements. Lateral bend was selected as right/left mouse click and gripper open/close for orientation control. Participants were 10 healthy adults, consisting of 4 women and 6 men. Participants were requested to remain still for a few seconds after the software had started until the word "Ready" appeared on the image. At this time, the head position was being calibrated. Participants were instructed to slowly move their heads up, down, left, and right to see if the software properly tracked head movement. As lighting and head-base conditions changed, participants were asked to calibrate at the start of each experiment. The distance between the user and the camera used in this experiment is 50 cm, 75 cm, and 100 cm. Lighting variations were carried out in bright, half-bright, and dark conditions. Bright conditions using 2 lamps in the room. Half-bright conditions, only use 1 lamp in the room, and in dark conditions, only use laptop lights. The head-base position variation is performed in the upright position of the user's head with the camera and head tilted to the side (approximately 45 degrees).

A. Mouse Head-Control Results

1) Control Performance: The experimental results showed that the participants could move the cursor well and smoothly. All participants could complete the given task correctly without any problems quickly. The participants' average movement trajectories were excellent and almost identical to the predetermined efficient path, as shown in Fig. 7. Successive boxes on the cursor control UI had varying distances and click function commands. Participants had to perform flexion and rotating movements to the right to move the cursor to box 2 from box 1. For the first trial, the average participant would experience confusion in moving the cursor to the right and left and looking for the correct box position. After the second experiment, participants were more flexible in moving the cursor. Table I summarizes the average time required by participants to move the cursor to each box in sequence. The easiest way was to move the cursor from box 3 to box 4 because it only went down slightly to the right. The average time traveled was also the fastest. The second fastest travel time was cursor movement from box 4 to box 5. The most difficult trajectory with the longest travel time was to move the cursor from box 5 to box 6. Moving the cursor to a higher position was more difficult than to a lower one, depending on how well the participant could control the cursor. The time required by participants to perform the right or left click function on the mouse was similar; this indicated that these two functions could work quite balanced. Overall, THC made it easy to control the cursor using head movement effectively with an efficient path, as shown in Fig. 8.

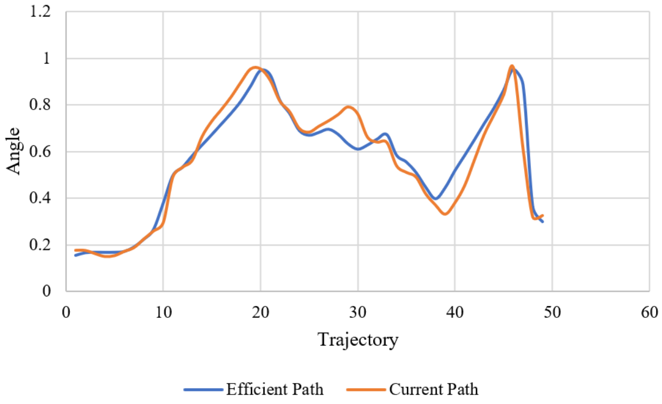


Fig. 7. Path result of the proposed cursor control.

TABLE I
Average Time for Cursor Control (Second)

ID	Box								Mouse Click	
	1	2	3	4	5	6	7	8	L	R
1	0.85	3.86	2.87	1.67	4.10	3.66	2.28	2.12	3.65	2.34
2	0.87	7.32	5.15	1.37	2.49	4.51	3.49	3.10	2.13	1.90
3	0.93	3.84	3.94	1.69	2.29	5.55	4.65	1.64	1.77	1.58
4	0.78	2.67	3.88	0.48	2.82	3.76	2.42	1.19	2.54	1.12
5	1.02	7.78	2.91	2.05	4.11	5.04	3.86	1.58	2.25	1.19
6	0.83	4.81	0.93	1.15	2.06	5.15	2.72	1.22	0.91	2.75
7	1.15	3.56	2.91	1.13	2.34	3.25	2.32	1.17	1.73	0.62
8	0.94	1.75	0.51	0.80	0.90	1.50	1.32	2.85	1.35	0.93
9	0.63	2.99	1.50	1.52	1.52	3.21	1.24	2.48	0.72	0.49
10	0.81	2.59	2.04	0.83	1.42	1.88	1.89	1.33	1.69	1.47
Avg	0.88	4.11	2.66	1.27	2.40	3.75	2.62	1.58	1.87	1.44

Participants were asked to rate from 1 to 5 (the fifth was the highest) of the eight questions on the questionnaire. Table III shows the mean and SD of the scores for each question. Participants were delighted with the feedback from the mouse/cursor movement shown on the UI with the black line. According to the questionnaire results, it was considered easier to move the cursor horizontally than vertically. In addition, the questionnaire also showed that THC could move the cursor with precision. THC was quite reliable as a physical mouse replacement and made it possible to help disabled people.

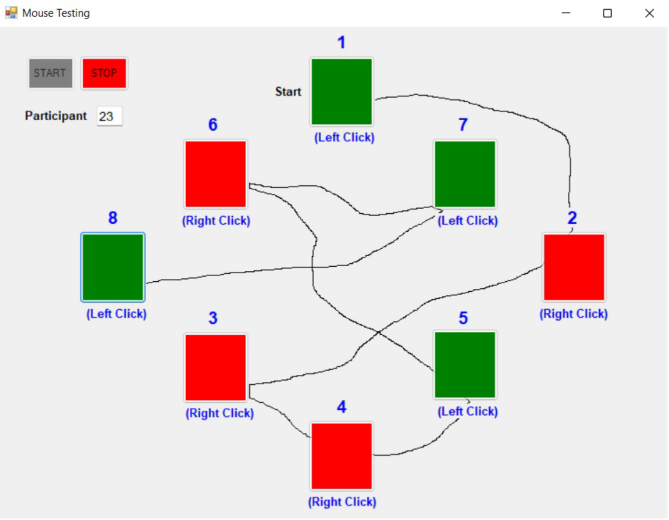


Fig. 8. Result of the proposed cursor control.

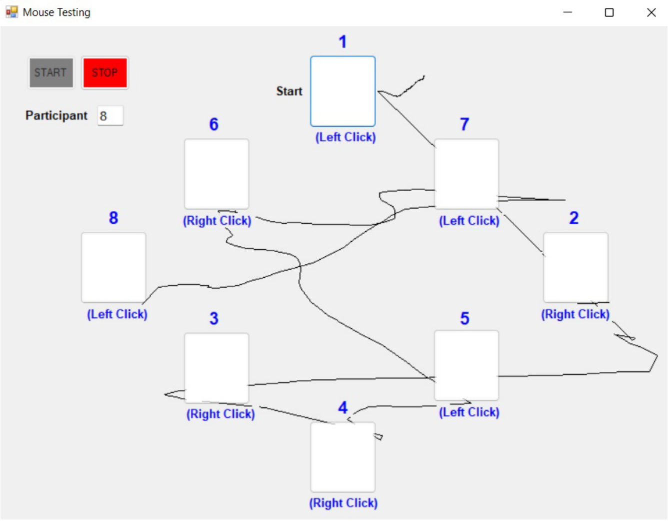


Fig. 9. Camera mouse path result.

TABLE II
CURSOR CONTROL PERFORMANCE

Metrics	Conditions	Mean	SD
Target selection (score)	Different lighting	8.48	0.54
	Different distance	8.07	0.47
	Head-base position	7.95	0.49
Movement time (sec)	Different lighting	2.29	0.99
	Different distance	3.18	1.09
	Head-base position	4.94	1.13
Path accuracy (%)	Different lighting	93.88	5.71
	Different distance	86.28	12.48
	Head-base position	78.65	15.49

TABLE III
CURSOR CONTROL QUESTIONNAIRE

No	Statement	Mean	SD
1	UI can represent the use of the mouse by the user	4.7	0.46
2	The feedback on the current head orientation is easy to understand and useful	4.1	0.70
3	Feedback from cursor movement is useful	4.8	0.40
4	Right and left mouse click is useful and easy	4.1	0.83
5	It's easy to move the cursor horizontally	4.3	0.64
6	It's easy to move the cursor vertically	3.9	0.70
7	It's easy to move the cursor with precision	4.0	0.63
8	I can well imagine the benefits of using mouse head-control for disabled people	4.7	0.48

Table II summarizes the average performance metrics (Mean and SD) for light, distance, and head-base variations. The analysis results showed that the average accuracy of the target selection score in the THC system was (M = 8.17, SD = 0.5). Target selection score was the highest when the acquisition task was completed in different lighting conditions (M = 8.48, SD = 0.54), while the head-base position variation showed the lowest accuracy (M = 7.95, SD = 0.49). The average path efficiency showed excellent results with varying lighting conditions (M = 86.27%, SD = 11.23). On average, the travel time in the experiment was considered fast (M = 3.47 sec, SD = 1.07). The proposed model showed no significant difference in the head pose estimation accuracy with variations in-room lighting and head-base position.

2) Comparison With Camera Mouse: Camera Mouse is free software that allows persons with limited motor movements to utilize a computer [23]. The software's cursor movement and selection settings are set to default settings. The radius is set to "normal" with the idle time of 1 second, and the horizontal and vertical sensitivity is set to "medium". The software automatically selects features in the user's eye to track. A green box appears around the selected feature so that the user can observe the cursor movement based on its position. The visual tracking algorithm evaluates the feature's shift to determine the current cursor position as the user moves his head. The cursor's coordinates on the computer screen are directly mapped according to the location of the feature being tracked. The sensitivity and smoothness of the software setting affect the coordinates' position and the trajectory of the cursor. Cursor sensitivity affects how eye movement is translated into cursor movement. Slight eye movement is converted into many cursor movements when the sensitivity is high. THC used artificial intelligence, making the system not require manual settings like Camera Mouse. Experimental results showed that cursor control could still be carried out accurately and smoothly under various challenging conditions, as shown in Fig. 9.

Fig. 10 compares the performance results of the THC method with a camera mouse with variations in lighting conditions, distance from the camera, and head-base position. Task performance was assessed using target selection scores, travel time between targets, and path accuracy. Because the Camera Mouse relied solely on the user's eye feature, its performance was degraded when conducted in a dark room. The Camera Mouse also did not perform effectively at long distances because the user's eye features were not

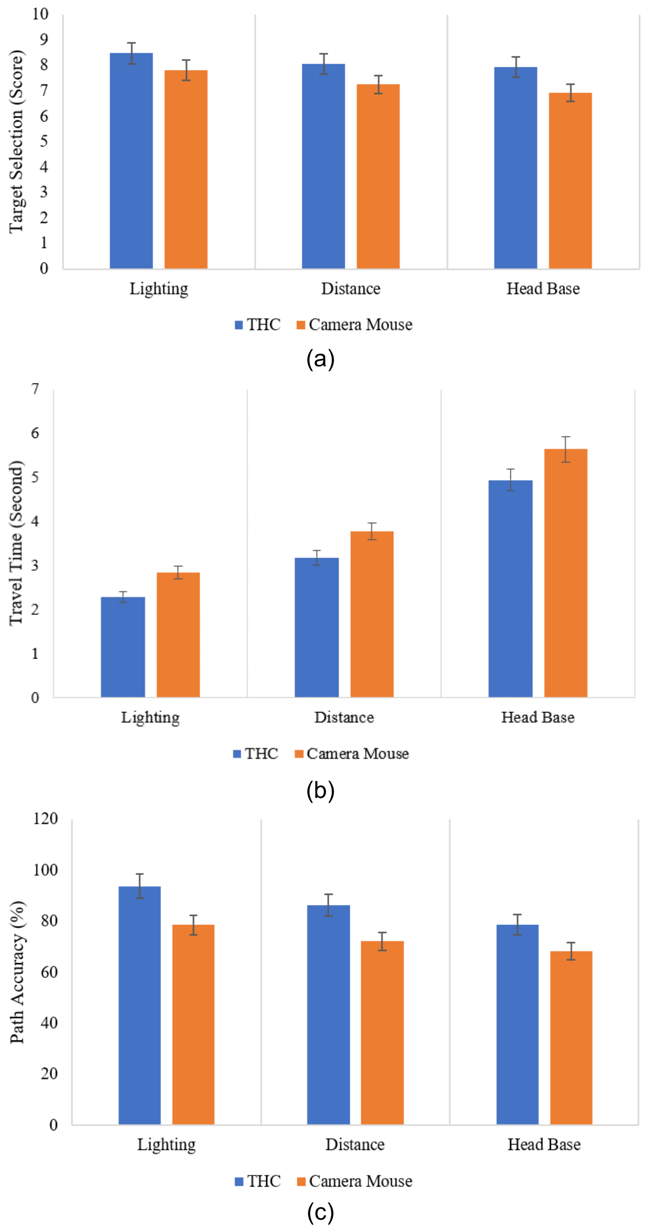
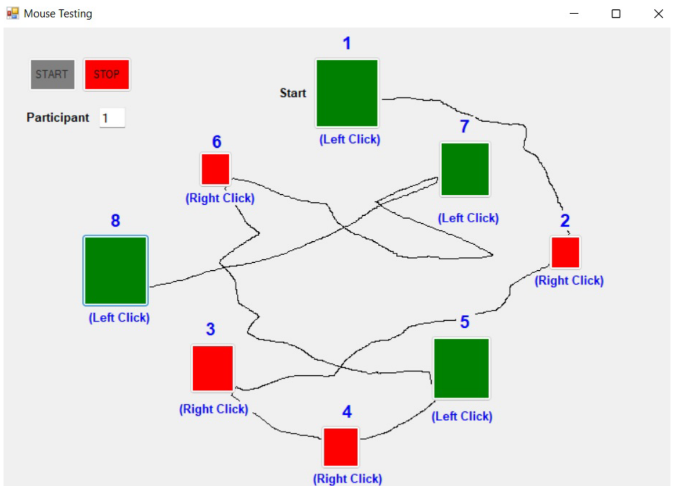


Fig. 10. Performance comparison results of THC and Camera Mouse. (A) Target selection score, (B) Travel time, and (C) path accuracy.

clearly identified. In THC performance results, the system's accuracy was not affected by darkroom lighting conditions and significant distance from the camera. In certain positions, users had difficulty controlling the cursor and had to be in an upright head position to improve performance. Thus, cursor control with Camera Mouse was not good enough when the head-base position was quite difficult. Camera Mouse allowed users to move the cursor freely with just a slight head movement. However, the cursor movement was unstable. Moving the cursor horizontally was considered more difficult than vertically. Thus, the cursor would move rather far in the horizontal position when a little movement was made, causing the path to be inefficient. In addition, performing the mouse clicking function was not easy.



Participant 1

Fig. 11. Result of the proposed cursor control with various box sizes.

The average travel time using THC (M = 3.47 sec) was statistically faster than the Camera Mouse (M = 4.09 sec). Different lighting conditions had no significant effect on travel time. Similarly, the difference in distance and head-base position did not affect the target selection score. Cursor controls were recorded at a 1920 × 1080 screen resolution, but system performance tests were adapted to the UI design. Using THC, participants selected all targets with an average path accuracy of 86.27%, 18% better than Camera Mouse. Participants completed the experiment using the Camera Mouse with a mean path accuracy of 73.03%.

3) Control Performance on Various Box Sizes: Fig. 11 shows a UI design with several different box sizes. The average travel time result is not much different from the control cursor performed on the UI with the same box size (M = 3.57 sec). However, cursor control is a bit of an issue in some positions. Since it is more difficult to move the cursor up than down, there is a slight difficulty when the cursor is moved from box 5 to box 6. Box 6 (50 × 50) is smaller than the other boxes, making it slightly difficult to navigate. However, cursor control can still be done well. In addition, from box 6 to box 7, due to their different sizes, it is also more challenging to move the cursor compared to a UI that has the same box size. Overall, slight UI issues with different box sizes did not affect cursor control performance.

B. Orientation Head-Control Results

The experimental results showed that all participants could move the box to simulate the robot arm movement with head movements. All participants were able to complete the assigned task correctly without any problems. Some participants could not control the gripper properly for complex tasks because the head did not move in the correct orientation. In the subsequent trial, all participants could be more flexible and fast in controlling the boxes on the UI according to the order of points. The participants' average movement path results were excellent, as shown in Fig. 12. Sequential points in UI had different distances. Table IV summarizes the time required by participants to move the red box on the UI from point 1 to 7

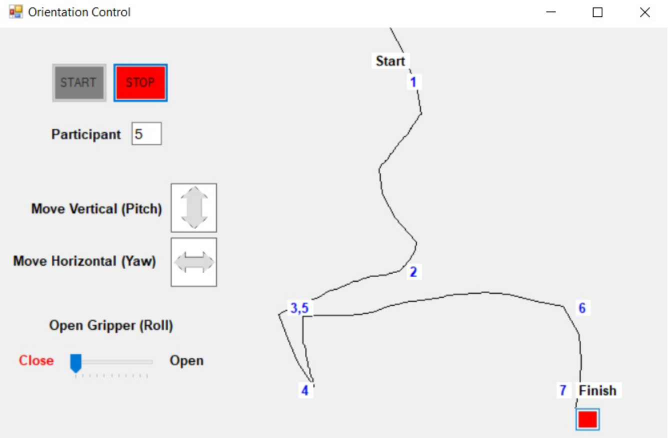


Fig. 12. Path result of the proposed orientation control.

TABLE IV
AVERAGE TIME FOR ORIENTATION CONTROL (SECOND)

ID	Point							Gripper	
	1	2	3	4	5	6	7	Open	Close
1	1.12	4.78	4.74	3.20	3.78	6.98	2.35	4.12	3.67
2	1.36	5.89	5.12	4.31	4.71	7.53	3.41	3.98	3.13
3	1.45	6.12	5.67	4.89	2.98	8.92	2.49	2.70	4.57
4	0.98	6.89	4.97	5.12	3.77	9.13	2.52	3.79	4.21
5	0.85	5.43	4.43	3.95	3.56	9.67	2.67	3.45	3.68
6	1.27	7.81	5.89	4.47	4.11	8.02	3.56	4.56	3.73
7	1.22	8.12	4.12	5.62	2.73	7.65	2.77	5.12	4.12
8	1.32	9.10	5.33	4.11	4.62	6.61	1.81	4.34	3.20
9	1.68	8.34	6.67	3.82	3.58	7.52	1.92	5.08	4.89
10	1.53	6.72	4.25	4.53	3.46	8.09	2.53	5.30	3.95
Avg	1.28	6.92	5.11	4.40	3.73	8.01	2.60	4.24	3.91

TABLE V
ORIENTATION CONTROL PERFORMANCE

Metrics	Conditions	Mean	SD
Target selection (score)	Different lighting	8.62	0.58
	Different distance	8.47	0.62
	Head-base position	8.38	0.65
Movement time (sec)	Different lighting	4.47	1.93
	Different distance	5.01	1.99
	Head-base position	6.02	2.45
Path accuracy (%)	Different lighting	85.33	8.30
	Different distance	80.18	3.44
	Head-base position	70.23	17.21

TABLE VI
ORIENTATION CONTROL QUESTIONNAIRE

No	Statement	Mean	SD
1	UI can represent the orientation control by the user	4.5	0.67
2	The feedback on the current head orientation is easy to understand and useful	4.1	0.70
3	The feedback from trajectory movement is useful	4.5	0.67
4	It's easy to move the box horizontally	4.2	0.74
5	It's easy to move the box vertically	4.1	0.70
6	It's easy to move the gripper in simulation	3.9	0.70
7	It's easy to move the box with precision	4.0	0.63
8	It's easy to estimate the position of the box	4.0	0.77
9	I can well imagine how the robot arms move in real applications	4.1	0.70
10	I can well imagine how the gripper moves in real applications	3.8	0.75
11	I can well imagine the benefits of using orientation head-control for disabled people	4.3	0.64

sequentially, including opening and closing the gripper. The experimental results showed that participants could control well and quickly. The farthest vertical position was between points 1 and 2, and the farthest horizontal position was between points 5 and 6.

The time required by participants to complete a predetermined control task was measured. The travel time based on different exposures ($M = 4.47$ sec, $SD = 1.93$) was significantly higher than the travel time with variations in distance ($M = 5.01$ sec, $SD = 1.99$) and head-base position ($M = 6.02$ sec, $SD = 2.45$), as summarized in Table V. SD in travel time increased with the complexity of the control task. Male participants tended to complete control tasks faster than female subjects. Target selection scores were highest when task acquisition tasks were completed in different lighting conditions ($M = 8.62$, $SD = 0.58$), whereas head-base position variation showed the lowest accuracy ($M = 8.38$, $SD = 0.65$). Compared to variations in distance and head-base, the average path efficiency showed excellent results with variations in lighting conditions ($M = 85.33\%$, $SD = 8.30$).

Participants were asked to rate from 1 to 5 (the fifth was the highest) on eleven questionnaire questions. Table VI shows the mean and SD scores for each question. Participants were delighted with the feedback from the orientation movement path displayed on the UI. According to the questionnaire results, it was easier to move the cursor horizontally than vertically. Furthermore, the questionnaire showed that THC could be used for precise orientation control. The robotic arm simulation was reliable enough to illustrate the advantages of control using only head movements captured by a single camera without the aid of any other devices or sensors. Thus, THC proved that the proposed system could help disabled people.

VI. DISCUSSION

The simulation results of the proposed cursor control method were then compared with the Camera Mouse in terms of performance metrics of target selection accuracy, travel time, and path efficiency. Changes in room lighting conditions, the distance between participants and the camera, and the head-base position were also considered to suit daily computer use. THC performance results showed efficient path cursor movement and accurate target selection. However, users with Camera Mouse experienced decreased performance when the room conditions were dark and when the participant's position was a bit far from the camera. This constraint occurred due to the drift effect and method dependence on high face contrast for track functionality.

Meanwhile, the results showed no significant difference in control performance when THC was used in different lighting conditions. In both systems, the travel time increased with the complexity of the task. THC had better control over the efficiency of the cursor path even though the cursor movement speed was increased. These results indicated that movement speed and target selection accuracy were essential factors to consider when assessing the performance of access methods. Due to the light reflection on the glasses, the automatic feature selection using Camera Mouse could not be selected accurately in participants wearing glasses. As a result, for these participants, the feature was manually picked. In addition, the Camera Mouse had to be placed directly in front of the user for optimal tracking performance. Different computer orientations would change the direction of the camera towards the user.

One of the aims of this application is to mimic a regular computer mouse for basic computer use. So that disabled people that have restricted movements can still operate freely on computers. Overall, the difference in performance between the two computer cursor control systems indicates that users' skills and preferences influence effectiveness. The mechanism in THC was made as effective as possible to provide a more reliable access option for those with difficulty keeping their head still. The over-sensitive cursor control on the Camera Mouse made it difficult for individuals with single muscle activation to maintain the cursor steady. As a result, the computer screen had many unintentional cursor movements or selections. This problem could be overcome using the Kalman filter on THC to control cursor movement better.

Head movements can also control the direction or orientation of 3D objects or robots. The experimental results showed that all participants could use the UI and a simulation of the movement of the robotic arm and gripper. Controls were adapted to the task at hand so that THC could be used in the relevant setting. THC was not limited to two-dimensional application controls like most interfaces with traditional head controls. THC allowed robot control at three DOF on the x-, y-, and z-axes. Instead of predefined execution operations, this technology enabled the user to direct the actual movement of the robot. The camera sensor module was integrated into the computer, allowing a quick and simple setup. Calibration was performed in just a few seconds to identify the neutral head position. The controls never made any unwanted movements during the experiment, demonstrating the system's robustness. Various gestures could be easily accommodated through the UI design. Individual limitations in motion constraints were not considered in the current analysis method. In the future, THC could be equipped with a combination of IMUs and other input modalities, such as eye trackers or voice recognition systems. Combining input modalities enables greater application functionality and opens new applications with a head gesture-based touchless method.

We have demonstrated the ability of the proposed method, THC,[1,2] to control the cursor and object orientation using head movements under various demanding conditions. In the video, the controls are moved slowly to mimic the movements of people with motor limitations. The video shows the results of the head movements and controls performed on the UI. Optimizing accuracy and computation time in real applications is recommended for further research. More accurate face detection methods allow for better application of methods. Furthermore, functionality in the UI, such as different box sizes, can be added to test cursor control that is more closely related to actual computer usage conditions.

VII. Conclusion

A new solution for touchless assistive devices using head movements has been proposed. Our touchless head-control (THC) method could be used to control the mouse cursor and objects or robots. This algorithm could be operated using an RGB camera and did not require additional equipment, such as sensors or electrodes. The obtained results allowed accurate on-screen mouse control and precise control of the robot's orientation. Experiments conducted on a group of people demonstrated the usefulness of the proposed method in real-time applications. All subjects could intuitively control mouse cursors and objects with head movements. This system provided the possibility to remotely control the movement of robots, such as robotic arms and grippers. In addition, the system did not require time-consuming adjustments or calibrations. Calibration was only necessary to determine the neutral head position within seconds. Users could activate and deactivate the control with head movement. During the experiment, the system never made any unwanted movements, which confirms the system's robustness. The interaction design could be easily adapted to the user's demands. Future studies would cover various standardized tasks with healthy people and disabled individuals to evaluate system performance by potential users. The results indicated that the THC system was adequate for human-computer interaction and access control to assist disabled people with restricted motor skills.

[1] https://youtu.be/zRdCA2ZnrOo

[2] https://youtu.be/7UZPYJKDSWE

References

[1] J. M. Vojtech, S. Hablani, G. J. Cler, and C. E. Stepp, "Integrated head-tilt and electromyographic cursor control," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 28, no. 6, pp. 1442–1451, Jun. 2020.

[2] J. K. Muguro *et al.*, "Development of surface EMG game control interface for persons with upper limb functional impairments," *Signals*, vol. 2, no. 4, pp. 834–851, Nov. 2021.

[3] A. Jackowski, M. Gebhard, and R. Thietje, "Head motion and head gesture-based robot control: A usability study," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 26, no. 1, pp. 161–170, Jan. 2018.

[4] M. A. Haq, S.-J. Ruan, M.-E. Shao, Q. M. U. Haq, P.-J. Liang, and D.-Q. Gao, "One stage monocular 3D object detection utilizing discrete depth and orientation representation," *IEEE Trans. Intell. Transp. Syst.*, early access, May 23, 2022, doi: 10.1109/TITS.2022.3175198.

[5] C. Gerdtman, Y. Bäcklund, and M. Lindén, "A gyro sensor based computer mouse with a USB interface: A technical aid for motor-disabled people," *Technol. Disability*, vol. 24, no. 2, pp. 117–127, Jun. 2012.

[6] H. Zhang, B.-C. Chang, Y.-J. Rue, and S. K. Agrawal, "Using the motion of the head-neck as a joystick for orientation control," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 27, no. 2, pp. 236–243, Feb. 2019.

[7] M. A. Velasco, A. Clemotte, R. Raya, R. Ceres, and E. Rocon, "Human-computer interaction for users with cerebral palsy based on head orientation. Can cursor's movement be modeled by Fitts's law?" *Int. J. Hum.-Comput. Stud.*, vol. 106, pp. 1–9, Oct. 2017.

[8] A. Laddi, V. Bhardwaj, N. Kapoor, D. Pankaj, and A. Kumar, "Unobtrusive head gesture based directional control system for patient mobility cart," in *Proc. Int. Conf. Signal Process., Comput. Control (ISPCC)*, Sep. 2015, pp. 236–240.

[9] B. E. Dicianno, R. A. Cooper, and J. Coltellaro, "Joystick control for powered mobility: Current state of technology and future directions," *Phys. Med. Rehabil. Clin. North Amer.*, vol. 21, no. 1, pp. 79–86, Feb. 2010.

[10] M. R. Williams and R. F. Kirsch, "Evaluation of head orientation and neck muscle EMG signals as command inputs to a human–computer interface for individuals with high tetraplegia," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 16, no. 5, pp. 485–496, Oct. 2008.

[11] J. J. Tellez-Guzman *et al.*, "Velocity control of mini-UAV using a helmet system," in *Proc. Workshop Res., Educ. Develop. Unmanned Aerial Syst. (RED-UAS)*, Nov. 2015, pp. 329–335.

[12] K. Tadano and K. Kawashima, "A pneumatic laparoscope holder controlled by head movement," *Int. J. Med. Robot. Comput. Assist. Surgery*, vol. 11, no. 3, pp. 331–340, Sep. 2015.

[13] H. A. Alhamzawi, "Control mouse cursor by head movement: Development and implementation," *Appl. Med. Informat.*, vol. 40, nos. 3–4, pp. 39–44, 2018.

[14] D. Sawicki and P. Kowalczyk, "Head movement based interaction in mobility," *Int. J. Hum.–Comput. Interact.*, vol. 34, no. 7, pp. 653–665, Nov. 2017.

[15] R. H. Abiyev and M. Arslan, "Head mouse control system for people with disabilities," *Expert Syst.*, vol. 37, no. 1, Feb. 2020, Art. no. e12398.

[16] N. Rudigkeit, M. Gebhard, and A. Graser, "An analytical approach for head gesture recognition with motion sensors," in *Proc. 9th Int. Conf. Sens. Technol. (ICST)*, Dec. 2015, pp. 1–6.

[17] N. Rudigkeit, M. Gebhard, and A. Gräser, "Towards a user-friendly AHRS-based human-machine interface for a semi-autonomous robot," in *Proc. Int. Conf. Intell. Robots Syst.*, Sep. 2014, pp. 1–4.

[18] H. Ohtsuka, T. Kato, K. Shibasato, and T. Kashimoto, "Non-contact head gesture maneuvering system for electric wheelchair using a depth sensor," in *Proc. 9th Int. Conf. Sens. Technol. (ICST)*, Dec. 2015, pp. 98–103.

[19] A. Bochkovskiy, C.-Y. Wang, and H.-Y. Mark Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.

[20] W. Rahmaniar, Q. M. U. Haq, and T.-L. Lin, "Wide range head pose estimation using a single RGB camera for intelligent surveillance," *IEEE Sensors J.*, vol. 22, no. 11, pp. 11112–11121, Jun. 2022.

[21] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1867–1874.

[22] X. Zhu, Z. Lei, X. Liu, H. Shi, and S. Z. Li, "Face alignment across large poses: A 3D solution," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 1, pp. 78–92, Nov. 2015.

[23] M. Betke, J. Gips, and P. Fleming, "The camera mouse: Visual tracking of body features to provide computer access for people with severe disabilities," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 10, no. 1, pp. 1–10, Mar. 2002.