https://doi.org/10.1007/s00138-022-01290-1

ORIGINAL PAPER

# Estimating human body orientation from image depth data and its implementation

Bima Sena Bayu Dewantara[1] · Rizka Wahyu Aditiya Saputra[1] · Dadet Pramadihanto[1]

Received: 29 July 2020 / Revised: 15 December 2021 / Accepted: 14 February 2022 / Published online: 19 March 2022

**Abstract**

This paper proposes a human body orientation estimation method using the Kinect camera depth data. The input of our system consists of three one-dimensional distance-based signals which reflect the body's surface contours of the human upper body portion, i.e., the upper chest, upper abdomen, and lower abdomen. Such signals are then normalized using their distances to achieve the same amount of the lower parts. All normalized signals are concatenated to provide a mix of contour features. We used Support Vector Regression (SVR) to classify the feature and Kalman Filter to estimate the continuous orientations instead of using discrete orientations. We also extend our work by adding human motion direction to the robust estimate of human body orientation when walking. We conducted two evaluation schemes, i.e., body orientation at static position and body orientation when moving. The experimental results show that our system achieves impressive results by achieving mean average of angle error (MAAE) of 0.097° and 5.82° for estimating body continuous orientation at static position and estimating body continuous orientation when moving, respectively. Therefore, it is very promising to be applied in real implementations.

**Keywords** Orientation estimation · Depth · Kinect · Support Vector Machine · Kalman filter

## 1 Introduction

Recently, the presence of social service robots cannot be avoided in order to help or assisting everyday human life. In some cases, to mutually working with humans, a robot must be able to understand human intention [1] and attention [2,3]. Human intention can roughly be modeled as the orientation of a human body when standing still or walking, while a human attention can be estimated as the orientation of human head due to limited field of view of human visual sensor.

In terms of mobile robot, human body orientation has a greater chance to be used since it can show the intention of human directly from his motion directions. When the human motion is inline with the robot motion direction, then, it means the level of human awareness is high. Therefore, the robot can keep its current action that is in accordance with the interaction task. On the contrary, if the human motion direction is different with the robot motion direction, it means the human level of awareness is getting low. This should forces the robot to stop its current action and giving more attention to the targeted human in order to plan another action that must be taken to deal with the human activity.

The topics about estimating human body orientation have gained special concerns of some researchers. The topics are mainly decomposed into several focuses, i.e., sensor types, features used, and static or moving objects. Based on the sensor types, we can decompose it into four commonly used sensors, i.e., Laser Range Finder (LRF) [4–6], RGB camera [7–13], RGB-D camera [14–17], and ToF camera [18]. With respect to the features used, some researchers worked on photometric image as presented in [10–12,14], whereas the combination of RGB image and depth was proposed in [15–18]. The other approaches like human body shape features were also presented in [4–9,13,18] as the solution. And finally, related to the static and moving objects, we have evaluated some references as shown in [4,5,9,11,13,15,18], while the rest use static image experiments from the dataset.

We have analyzed and evaluated some of the references that we mentioned earlier and concluded the results in Table 1. Based on the results of the comparison of references from Table 1, in general, solutions related to the estimation of human body orientation are divided into two sources of information, namely images from the camera and distance

✉ Bima Sena Bayu Dewantara
bima@pens.ac.id

[1] Politeknik Elektronika Negeri Surabaya, Surabaya, Indonesia

**Table 1** A brief comparison of some related previous research results

| References | Sensors | Features | Methods | Datasets | Static/moving | Full/upper | Step/range | Advantages/disadvantages |
|---|---|---|---|---|---|---|---|---|
| Choi [14] | Asus Xtion | RGB | Yolo+CNN | Human 3.6M, Private | Static | Full | 45°/360° | Direct implementation to robot MAAE=10.6°, Acc=81.58% |
| Liu [15] | Kinect | RGB-D | DBNS | Private | Both | Full | 45°/360° | Robust to clutterred environment, illumination change, and partial occlusions, MAE=12.23°, Comp. time=338 ms, Acc=N/A |
| Lewandowski [16] | Kinect | RGB-D | CNN | Private | Static | Full | 45°/360° | System runs on 13.43 fps, Acc=N/A MAE=10.3° (single), MAE=5.68° (biternion) |
| Glas [4] | Lidar | Cross-sectional contour | KLD PF | Private | Both | Upper | N/A | Multiple human position and orientation tracking, Acc=N/A, Position Acc=4.6 cm, MAE=7.4° |
| Shimizu [5] | Lidar | Body shape | UKF tracker | Private | Both | Upper | 10°/360° | Direct implementation to robot Sensitive to the change of shape MAE=0.83°, Acc=93.66% |
| Tepencelik [6] | Lidar | Ellipse fitting | NN regression | Private | Static | Full | 22.5°/±90° | Better accuracy than RGB, mixing information from body and head, MAE=8.37°, Acc=N/A |
| Baltieri [7] | RGB Camera | Mixture of wrap distribution | Multi-level HOG+RT | TUD Multiview Pedestrian dataset, Sarc3D, 3DPeS | Static | Full | 45°/360° | Multiple human detection & tracking, performance drops in some oblique angle, Acc=65.37% |
| Weinrich [8] | RGB Camera | RGB | HOG+SVM Decision Tree | UCI ML Repo. | Static | Upper | 45°/360° | Computation time faster than the competitors, MAE=22.5°, Acc=64% |
| Ardiyanto [9] | RGB Camera | Body Shape | HOG+LBP+PLS+ ovements+UKF | INRIA, Fudan-Penn, Calvin-UB, TUD Multi-view, TUD Stadtmitte, Private | Both | Upper | 45°/360° | Multiple human detection & tracking, real implementation, MAE=N/A Acc=64% (dataset), Acc=70% (real) |
| Kohari [10] | RGB Camera | RGB | CNN | SURREAL | Static | Full | 10°/360° | Comp. time=8.73 ms, MAE=6.9°, Acc=47.7% (zero degree acc.)<br>Acc=89.7% (ten degree acc.)<br>Acc=97.5% (twenty degree acc.) |

**Table 1** continued

| References | Sensors | Features | Methods | Datasets | Static/moving | Full/upper | Step/range | Advantages/disadvantages |
|---|---|---|---|---|---|---|---|---|
| Yu [11] | RGB camera | RGB+human keypoints | Softmax | COCO, TUD, 3DPeS | Both | Full | 45°/360° | The system is able to detect human keypoints, MAE=15.3°, Acc=75.7% (22.5°), Acc=96.8% (45°) |
| Wu [12] | RGB camera | RGB | HRNet ResNet | COCO | Static | Full | 5°/360° | Using large scale high precision human body orientation, MAE=8.4°, Acc=95.1% (22.5°), Acc=99.7% (45°) |
| Chen [13] | RGB camera | Body shape 2D appearance | Edge detection, grids, back projection 2D to 3D | Video sequences | Both | Full | 30°/360° | Multiple persons, system runs 25 fps, Only for fixed environment, Need 4 cameras, Acc=N/A |
| Shinmura [18] | ToF camera | Coaxial-RGB-D | Weighted HOG+SVM | Private | Both | Full | 45°/360° | Robust enough in a complicated background, Failed when the arm swing to much, Acc=78.9%, MAE=N/A |

from LiDAR. The methods that use information as a function of distance from LiDAR [4–6], on average, have a longer range, faster computation time and fairly good precision with a fairly small mean of absolute error (MAE) compared to images. However, LiDAR is an expensive device with limited functionality and usually still requires the help of other devices such as cameras to get better functionality.

On the other hand, camera-based methods are divided into two models, namely using photometric information from RGB images and using a combination of photometric information from RGB images, and depth. Several methods based on photometric information from RGB images that use traditional features and classification methods [7–9,11,13] have an average accuracy rate of below 70% with an MAE above 10°, while some methods based on photometric information from RGB images using Deep Learning or Convolutional Neural Network (CNN) methods [10,12,14] get better accuracy results with a fairly small MAE. Unfortunately, the use of Deep Learning and CNN is quite heavy so it requires additional equipment in the form of a GPU which is quite expensive. However, various backgrounds are still a weakness of the photometric-based method, while the combination of RGB and depth images has a better solution for separating objects from the background. By utilizing depth information in addition to RGB, traditional classification methods [15,16,18] are able to produce an average accuracy of above 75%.

This paper basically extends our work in [17] where we have developed a RGB-D-based human body orientation estimation system in a static position. The extension is done by combining our existing system with the human motion model so that the system can be implemented to estimate human orientation in motion. There are two main contributions in this paper. First, we converted the human upper body contours into a set of one-dimensional signals as the function of distances. This technique is effective to robustly estimate the human body orientation and efficient as well, by using a Kinect camera only which is far cheaper compared to laser range finder sensor. In addition, our system does not require the help of other devices such as an expensive GPU. Second, a combination with a human motion model was proven effective to estimate human body orientation in motion. Therefore, our system is ready to be used in real implementation.

The rest of the paper is organized as follows. Section 2 describes our features extraction and processing the features combination for estimating human body orientation. Section 3 discusses an implementation of our combined-feature for a real scene application of human body orientation estimation. Section 4 presents the experimental results and discussions. And finally, we conclude our work and describe possible future work in Sect. 5.

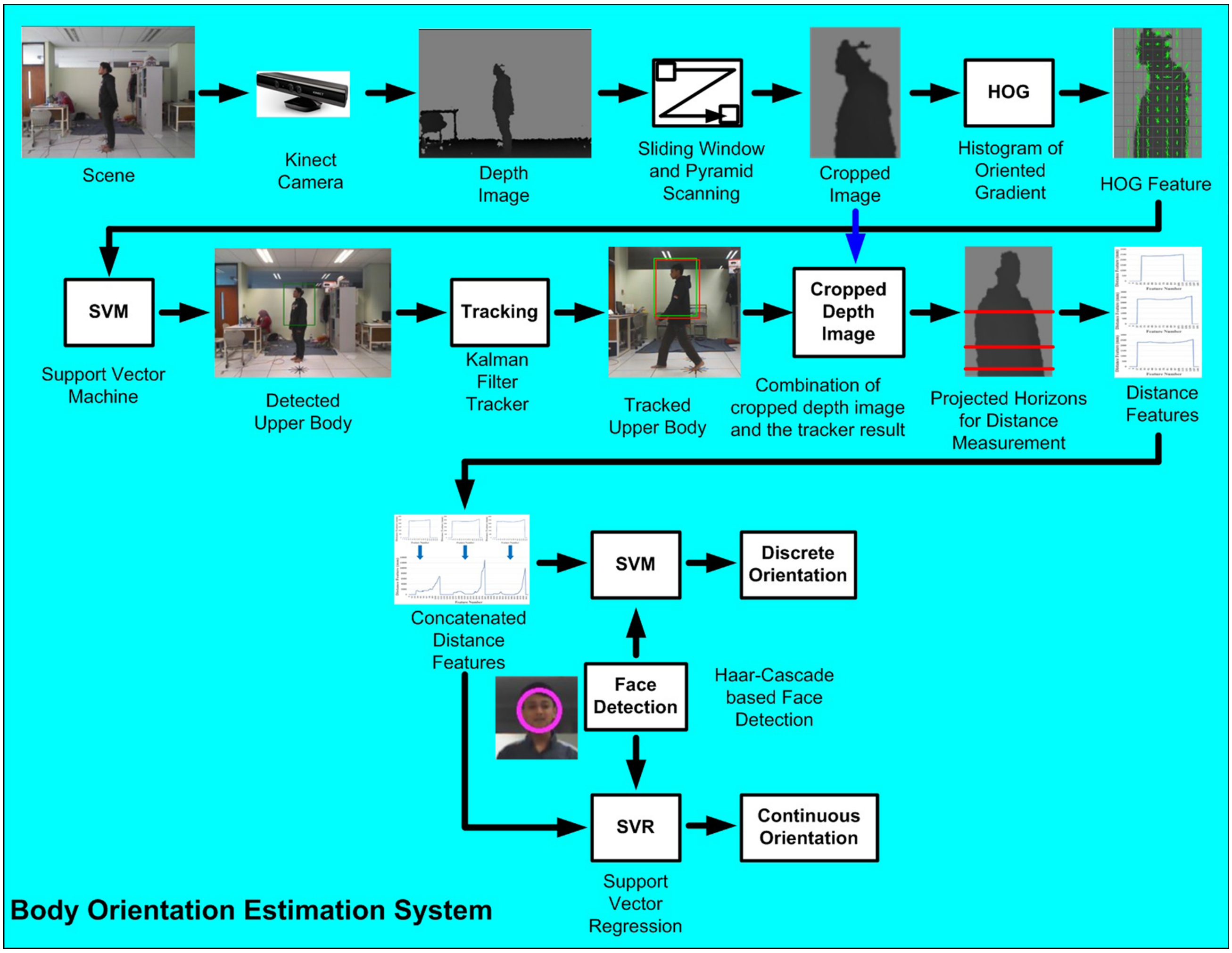

Scene
Kinect Camera
Depth Image
Sliding Window and Pyramid Scanning
Cropped Image
HOG
Histogram of Oriented Gradient
HOG Feature
SVM
Support Vector Machine
Detected Upper Body
Tracking
Kalman Filter Tracker
Tracked Upper Body
Cropped Depth Image
Combination of cropped depth image and the tracker result
Projected Horizons for Distance Measurement
Distance Features
Concatenated Distance Features
SVM
Discrete Orientation
Face Detection
Haar-Cascade based Face Detection
SVR
Continuous Orientation
Support Vector Regression
Body Orientation Estimation System

**Fig. 1** Whole design of our proposed human body orientation estimation system based on depth image

## 2 Depth-based human body orientation estimation

Based on our previous paper in Saputra et al. [17], our system description is shown in Fig. 1. We can briefly explain that a RGB-D camera is used to capture the scene and produces a RGB image and a depth image. The depth image is used to find human upper body based on the shape of upper half body. Once the upper body detected, then, three projected horizons are taken from human upper body segment. It produces three distance-based signals to be concatenated to become a body orientation feature. Support Vector Machine (SVM) is used to classify discrete orientation. With the help of face detection, the front and rear body parts can be distinguished so that the body orientation can be determined whether 0° or 180°. For other orientations, it is estimated directly using information from three curves projected to the upper body assisted by a Kalman Filter-based tracker.

### 2.1 Acquiring depth data from Kinect camera

In this work, a Kinect camera is utilized to capture scenes in front of the camera. The Kinect camera produces two outputs, i.e., RGB image and depth image. We use the depth image due to its advantage to easily separate object from background, so that the common problems that occur on the RGB image-based detection due to a variety of backgrounds such as false detection can be avoided. Figure 2 shows the result of depth image of human in a real environment.

Figure 2 shows that the shape of human can be seen clearly in silhouette even without human attribute details, like face and clothing color. The background is also less in detail, so that this advantage is actually very helpful to distinguish foreground object like a human from background.

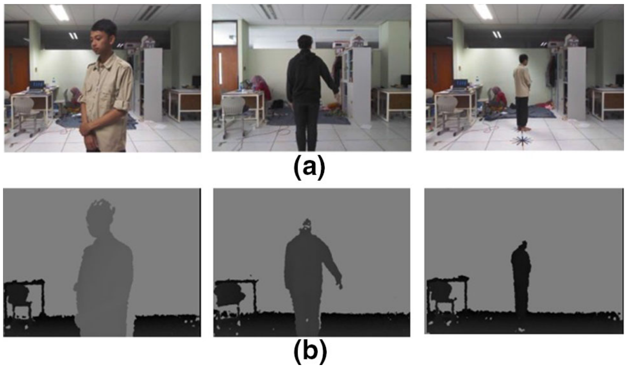


**Fig. 2** Result of depth image produced by Kinect camera. **a** the original input images shown in RGB and **b** the depth images

## 2.2 Human upper body detection and tracking

We employed the same method as shown in [19] by utilizing Histogram of Oriented Gradient (HOG) that is proposed by Dalal et al. [20] to extract the features of human upper body from the depth images. To train our human upper body detection system using HOG, we first cropped a frame which fits to the human upper body shape. Since the cropped frames may have different sizes, we resize all the depth images into the same size of 72 × 96 pixels. We chose this size because the human upper body shape can still be seen clearly and this reduction can speed up the computation time.

The frame is then divided into 9 × 12 blocks where each block contains 8 × 8 pixels. The gradient calculation is then carried out to each block to find 9 bins of gradient orientation, ranging from $0° - 180°$. The HOG feature extraction is formulated as follows.

$$m_{x,y} = \sqrt{(L_{x+1,y} - L_{x-1,y})^2 + (L_{x,y+1} - L_{x,y-1})^2}, \quad (1)$$

$$\phi_{x,y} = \tan^{-1}\left(\frac{L_{x,y+1} - L_{x,y-1}}{L_{x+1,y} - L_{x-1,y}}\right), -\frac{\pi}{2} < \phi < \frac{\pi}{2}, \quad (2)$$

where $m_{x,y}$ is a gradient at pixel (x,y), $L_{x,y}$ is a luminance value at pixel $(x, y)$ and $\phi_{x,y}$ is a calculated angle at pixel $(x, y)$. The HOG feature of human upper body is visualized using a histogram of $\phi$ that represents the overall characteristics of a human as shown in Fig. 3.

We have collected 5400 HOG features of human upper body as a positive samples dataset and 10,800 HOG features of nonhuman upper body as a negative samples dataset. The positive and negative samples datasets are used as a training dataset. We fed the training dataset to Support Vector Machine (SVM) using a library by Chang et al [21], called as LibSVM. The SVM's parameters setting is summarized in Table 2. All positive and negative samples are trained using LibSVM to get a model of human upper body. In our real implementation, we used the trained model to detect human upper body directly using a Kinect camera. The result of our detection system is shown in Fig. 4.

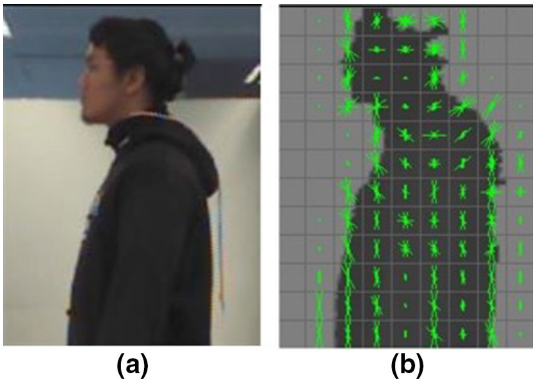


**Fig. 3** HOG feature on a depth image of human upper body. **a** the original image is shown in RGB and **b** the HOG feature of the depth image

**Table 2** Parameters setting of Support Vector Machine used to train the model of human upper body

| No. | Parameters | Value |
|---|---|---|
| 1. | Type | Multiclass C-SVC |
| 2. | Kernel type | Linear |
| 3. | C cost | 1 |

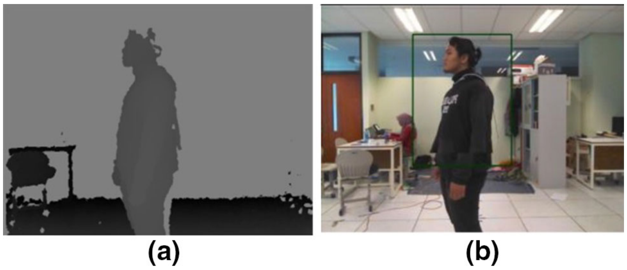


**Fig. 4** Result of human upper body detection. **a** the depth image and **b** the detected human upper body is bounding boxed

Once the human upper body is detected and bounding boxed, then we employ a tracker based on the Kalman filter [23] to stabilize the detection result. The state of our Kalman filter utilizes the model of a constant velocity of the center position of the bounding box, $(\dot{x_c}, \dot{y_c})$, and the person's relative distance to the camera, $\dot{d_c}$. The constant velocity model is applied to the center point of the bounding box, $(x_c, y_c)$, and the person's relative distance to the camera, $d_c$, includes its width, $w$, and height, $h$, and their corresponding process noises, $(\epsilon_{x_c}, \epsilon_{y_c}, \epsilon_{d_c}, \epsilon_w, \epsilon_h, \epsilon_{\dot{x_c}}, \epsilon_{\dot{y_c}})$, which is formulated as follows.

$$x_c(t) = x_c(t-1) + \dot{x_c}(t-1)\Delta t + \epsilon_{x_c},$$
$$y_c(t) = y_c(t-1) + \dot{y_c}(t-1)\Delta t + \epsilon_{y_c},$$
$$d_c(t) = d_c(t-1) + \dot{d_c}(t-1)\Delta t + \epsilon_{d_c},$$

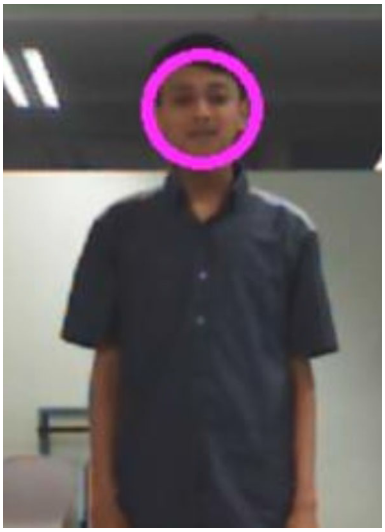


Fig. 5 Detectable face normally indicates a human frontal body part

$$
\begin{aligned}
w(t) &= w(t-1) + \epsilon_w,\\
h(t) &= h(t-1) + \epsilon_h,\\
\dot{x_c}(t) &= \dot{x_c}(t-1) + \epsilon_{\dot{x_c}},\\
\dot{y_c}(t) &= \dot{y_c}(t-1) + \epsilon_{\dot{y_c}},\\
\dot{d_c}(t) &= \dot{d_c}(t-1) + \epsilon_{\dot{d_c}},
\end{aligned}
\tag{3}
$$

where $\epsilon_{x_c} = 0.1$ and $\epsilon_{y_c} = 0.1$ are the noise's values set for the center point of the bounding box, $\epsilon_{d_c} = 0.01$ is the noise's value set for the person's relative distance to the camera, and $\epsilon_w = 0.1$ and $\epsilon_h = 0.1$ are the noise's values set for the width and height of the bounding box.

### 2.3 Face detection

The orientation of the human body can be easily determined by the presence or absence of the face. His face would be visible if anyone looks forward as shown in Fig. 5. Conversely, if the person faces backward, his face cannot be seen, while the left and the right direction also reveals a few face appearances that can be traced from the initial location of the face.

In this paper, we used the Haar-Cascade method [22] to detect the human face. The Haar–Cascade was employed inside the human upper body's bounding box only, to reduce the computation time. The detected face was useful for distinguishing human facing front or back, as it is difficult to differentiate the human stomach and back only by using contours.

### 2.4 Distance as a representation of feature

This section explains the core of our work. By utilizing the bounding box area of the upper body, we measured the relative distance of the body surface from the camera. We take the distance at each point, from the left end to the right end of the bounding box, from three horizontal projection lines on the upper chest, upper abdomen, and lower abdomen. We manually set the height of the three horizontal projection lines at the beginning, namely 0.45 × $h$, 0.75 × $h$, and 0.95 × $h$. By using the Kinect's library, the visualization of the three projected horizontal lines and their contour curves as the function of distances is shown in Fig. 6.

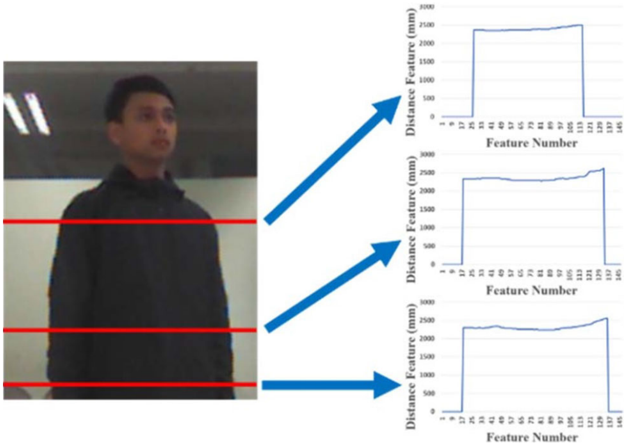


Fig. 6 Curves as the function of the distance of the projected horizontal lines on the human upper body area at $\alpha = -30°$

After taking a set of sample features at a certain angle, and the results confirmed can be used, we then collect distance data based on the 12 discrete angular orientations needed for SVM training. Figure 7 shows the results of the capture of the three signals from the line projection results on the three parts of the upper human body.

The main reason for using three horizontal line projections which represent the human upper body contours is to maintain the unique feature of orientation due to hand movements when walking. When walking, the arm will swing, where the upper chest does not change too much, the swing of the upper arm changes slightly, and the swing of the forearm changes a lot. By keeping the surface contours of the upper chest unchanged, and enriching the amount of variation in the upper and lower arm, this strategy makes our method reliable even when people walk.

### 2.5 Features normalization and combination

Before using distance-based feature data, we normalize the data to eliminate the effect of distance, $d$, on data. We subtract all amplitude of points of the curves by the nearest point distance value of the body from the camera. Therefore, we will only get a contour curve with the lowest value on the y-axis = 0. Then, to strengthen the characteristics of the curve, we squared the values of each point so that we get a sharper and clearer contrast amplitude difference. To standardize the

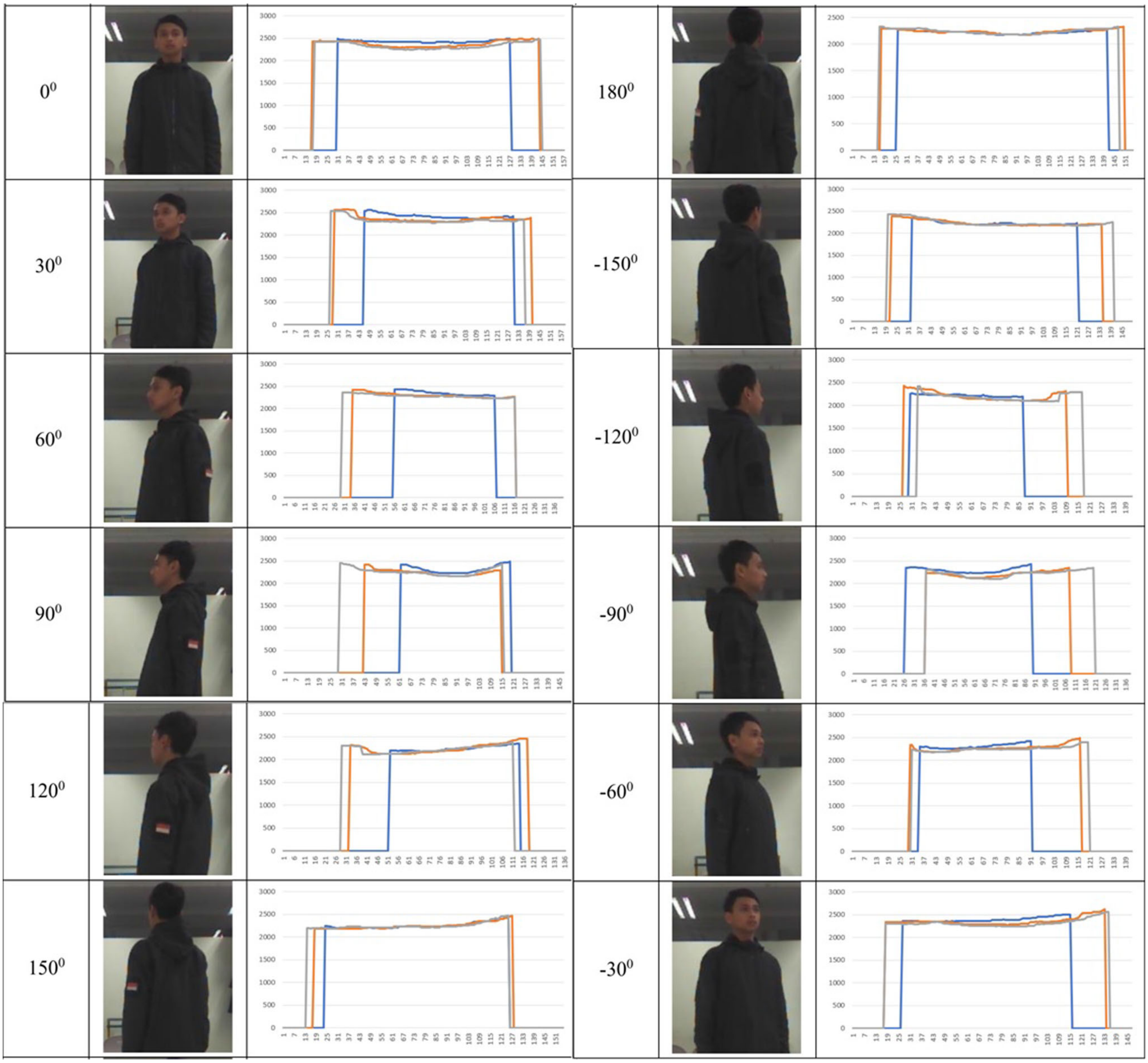


**Fig. 7** Examples of the curve as a function of the distance of the projected horizontal lines on the human upper body area for twelve orientation angles

**Table 3** Parameters setting for collecting training dataset

| No. | Parameters | Value |
|---|---|---|
| 1. | Relative distance, $d$ | 1.5 m, 2 m, 2.5 m, 3 m and 3.5 m |
| 2. | Type of clothes | shirt or t-shirt, jacket, cloak and hijab |
| 3. | Orientation, $\alpha$ (moving) | $-150°$, $-120°$, $-90°$, $-60°$, $-30°$, $0°$, $+30°$, $+60°$, $+90°$, $+120°$, $+150°$, $+180°$ |
| 4. | Orientation, $\alpha$ (stand still) | $-150°$, $-120°$, $-90°$, $-60°$, $-30°$, $0°$, $+30°$, $+60°$, $+90°$, $+120°$, $+150°$, $+180°$ |

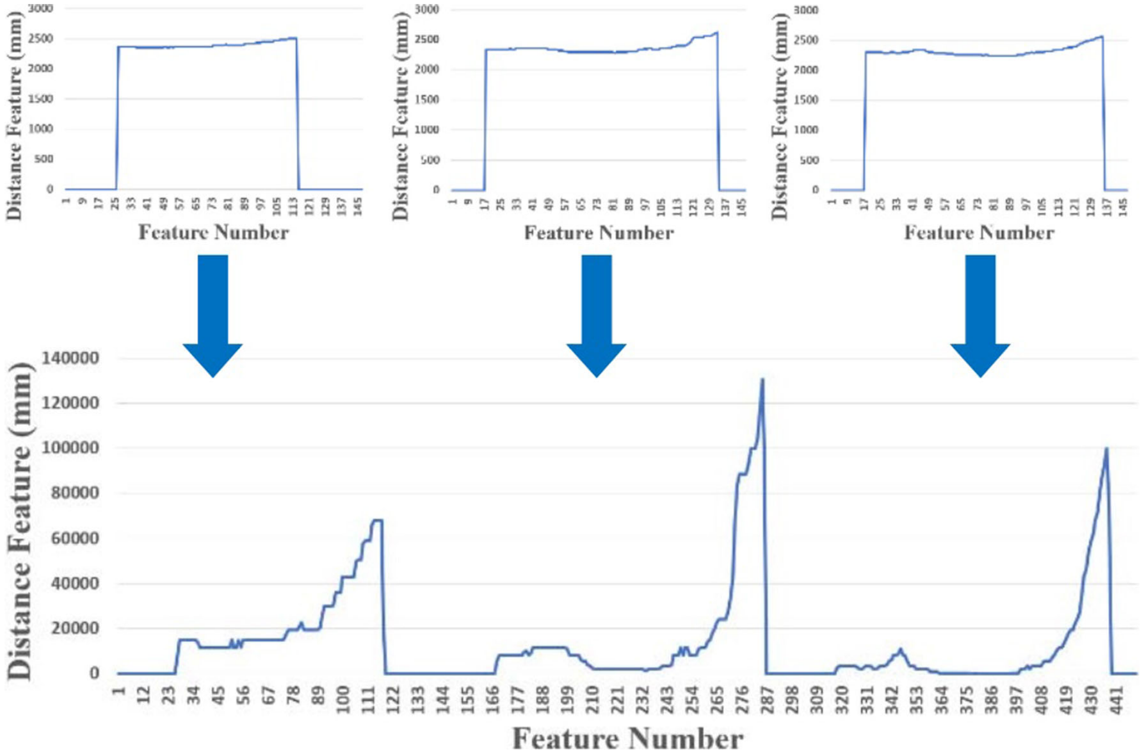


**Fig. 8** Distance features normalization and combination of a human with orientation of −30°

width of the data, we use the signal interpolation technique with the following formulation.

$$y = \frac{(x - x_1)(y_2 - y_1)}{(x_2 - x_1)} + y_1, \tag{4}$$

where $(x_1, y_1)$ and $(x_2, y_2)$ are the position of the first point and the second point of linear equation, respectively. $(x, y)$ is a pair of input–output point between the first point and the second point of a linear equation. Following that, three rows of data will be merged into one row of data. Figure 8 provides an example of a distance feature normalization and combination of a human when $\alpha = -30°$.

## 2.6 Classifying human body orientation

Following the main purpose of this study to estimate the orientation of the human body, $\alpha$, the normalized squared-distance feature needs to be classified into several orientation classes that have been previously designed. We discretized the body orientation classes with step of 30°, i.e., −150°, −120°, −90°, −60°, −30°, 0°, +30°, +60°, +90°, +120°, +150°, and +180°. We use Support Vector Machine (SVM) as a classifier.

Before SVM can be used to estimate body orientation, it must be trained first with the dataset of normalized squared-distance feature-based body orientation data. We collected 5,400 training data as the dataset, where the data was collected from 15 persons and divided into 12 discrete orientation classes. Each orientation class consists of 450 data taken based on mix between subject's relative distances to the camera, type of clothes and the human body postures taken from twelve angles. The data parameters are summarized in Table 3. Figure 9 shows the examples of a normalized squared-distance of body orientation training data samples, ranging from 0° up to 360°.

Retrieval of the dataset with distance variations is performed to see the impact of distance on body contour changes. The closer the distance, the more apparent the contours of the body, so that the fluctuation in the distance value of each projected horizontal point is clearer. Conversely, the farther the distance, the contours of the body, particularly the front and back sides, would be more difficult to discern, since the difference in the distance value of each projected horizontal point looks identical. On the other hand, clothes variations are intended to let our system to deal with various shapes and sizes of the body by wearing compact, loose, or thick clothes. And finally, various of the human body postures, especially hand swingings, are used to adapt our system for both stand still and moving person.

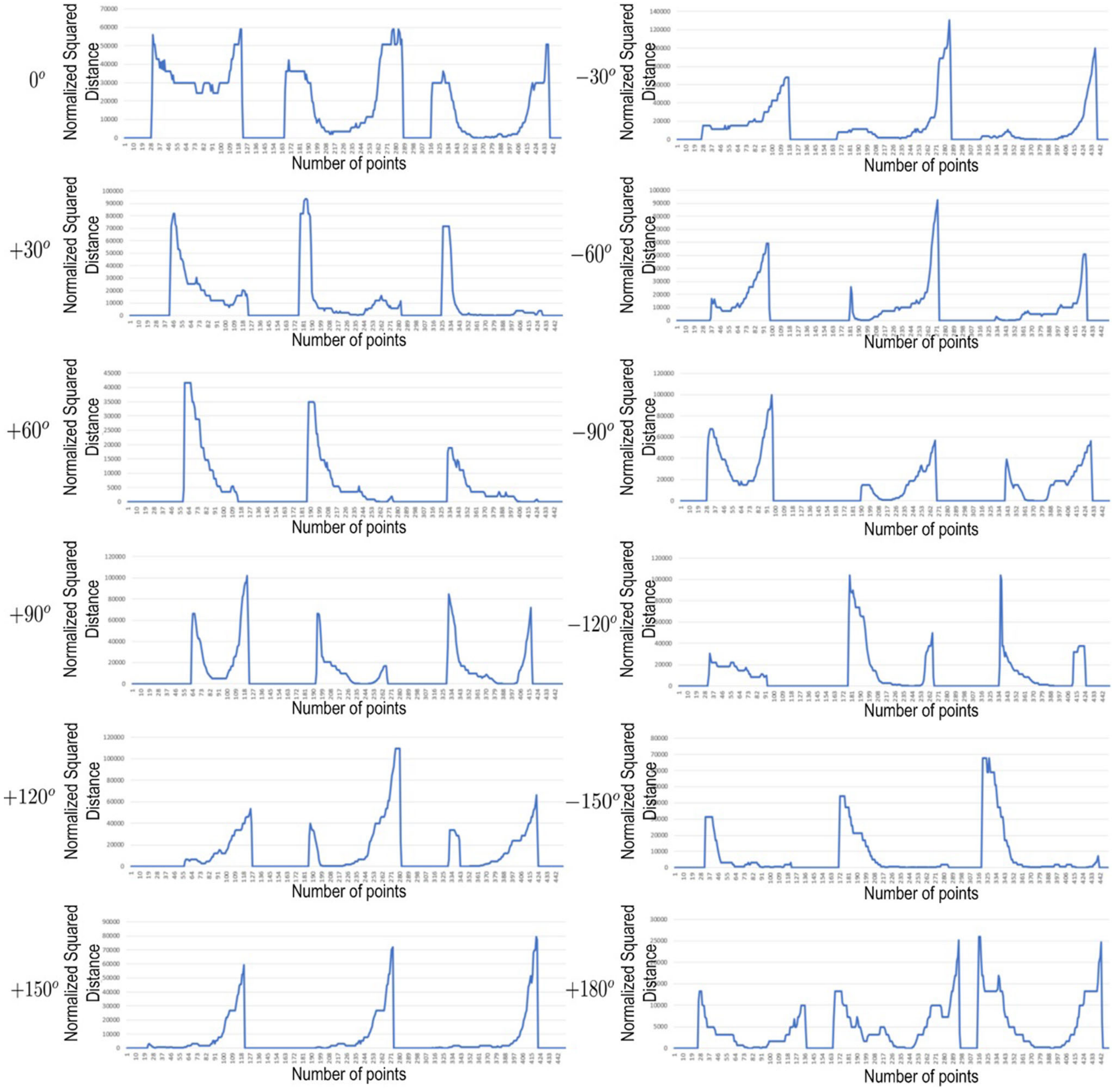


**Fig. 9** Examples of a normalized squared-distance of the discrete body orientation training data samples, ranging from 0° up to 360°

# 3 Estimating human body orientation in a real application

## 3.1 Target person detection and tracking

In this system, if the number of people detected is only one, that person will be automatically identified as the target partner. If it turns out that there is more than one person in the image frame, the target partner will be determined by the size of the bounding box of the largest of all individuals that can be detected by the system. This indicates that anyone nearest to the camera has a greater capacity for interaction than other people who are further away from the camera. To strengthen identification, we have introduced the ability of the system to track the displacement of the target partner positions by using the Kalman filter as described in Eq. 3. Figure. 10 demonstrates the effects of the identification and control of someone with the largest bounding box size. People who are considered as not targeted will not be bounding boxed.

**Table 4** Parameters setting of Support Vector Regression (SVR) for smoothing the orientations

| No. | Parameters | Value |
|---|---|---|
| 1. | Type | epsilon-SVR |
| 2. | Kernel type | Linear |
| 3. | C cost | 1 |
| 4. | epsilon | 0.1 |

### 3.2 Orientation tracking and smoothing

To deal with the real implementation, we have changed SVM with Support Vector Regression (SVR). This replacement is because we want the estimation results to be shown in a continuous form. So that the actual body orientation angle can be obtained precisely. The SVR is also using the LibSVM library by Chang et al. [21] with the parameters setting as shown in Table 4.

### 3.3 Combining with motion

There is a big difference when we try to measure the body's orientation as we remain in a position and walk. When the position of the body remains in place, the orientation calculation focuses only on rotational movements, whereas while walking, orientation is not only determined by rotation, but also by translation. Therefore, we combine the orientation estimation method that we discussed above by considering the direction of body movement that is modeled from the position shift in frame $t-1$ and frame $t$. Figure 11 shows our proposed online body continuous orientation estimation system to deal with human motion. The motions of the human body can be represented by the changes in position between two consecutive frames, which are formulated as follows.

$$s = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}, \tag{5}$$

$$\theta = \tan^{-1}\frac{y_2 - y_1}{x_2 - x_1}, \tag{6}$$

where $(x_1, y_1)$ is the body position at time of frame $t-1$, $(x_2, y_2)$ is the body position at time of frame $t$. $s$ is the length of displacement, and $\theta$ is the walking direction. We fused the body orientation, $\alpha$, and the walking direction, $\theta$, using a Kalman filter [24]. To fuse these two variables, we focused on the measurement update phase. The measurement update obtained only taking the $\alpha$ data is:

$$\hat{x}_\alpha(k|k) = \hat{x}(k|k-1) + K_\alpha(k)\left(z_\alpha(k) - H_\alpha\hat{x}(k|k-1)\right), \tag{7}$$

$$K_\alpha(k) = P_\alpha(k|k)(H_\alpha)^T(R_\alpha)^{-1}, \tag{8}$$

$$(P_\alpha(k|k))^{-1} = P^{-1}(k|k-1) + (H_\alpha)^T(R_\alpha)^{-1}H_\alpha, \tag{9}$$

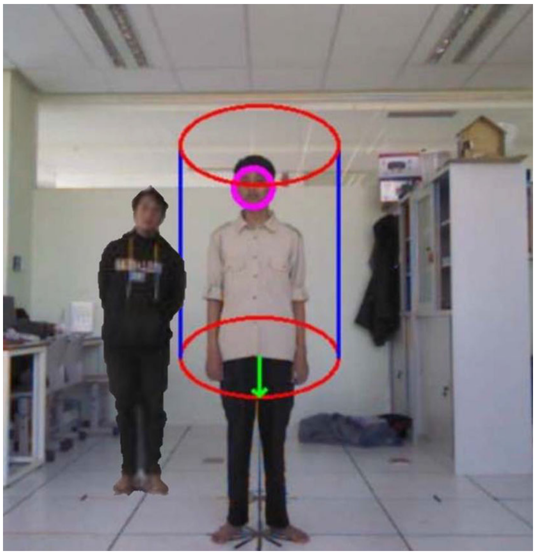


Fig. 10 Our system detects and tracks the target partner from the biggest detected object. The green arrow represents the body orientation, while the red ellipse is the roof and bottom of the tube which represents the bounding box area of the detected human upper body in the planar plane

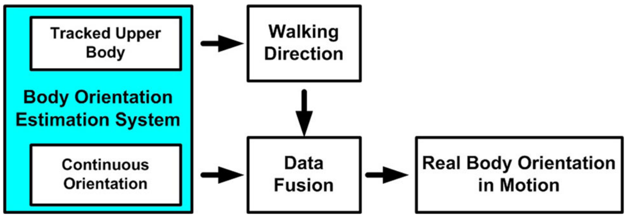


Fig. 11 Block diagram of our online body continuous orientation estimation in motion

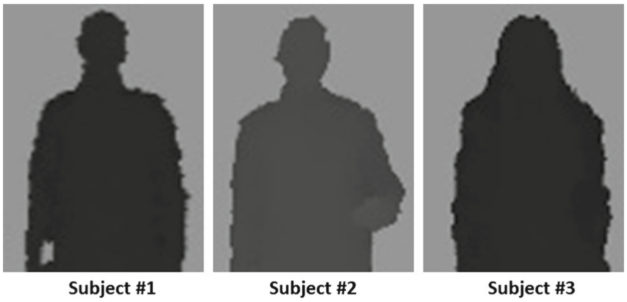


Fig. 12 Example of three participant images. Subject #1 using a jacket, Subject #2 using a t-shirt and Subject #3 using a hijab

where $\hat{x}_\alpha(k)$ is the estimation value of the body orientation at current state. $K_\alpha(k)$ is the Kalman gain for the body orientation. $H_\alpha$ is the observation matrix of the body orientation. $R_\alpha$ is the error covariance of estimation result of the body orientation. $z_\alpha$ is the scaling vector of the body orientation at current state. $P_\alpha(k)$ is the mean of error value estimation of the body orientation at current state. In the same way, the measurement update obtained only taking the $\theta$ data is:

$$\hat{x}_\theta(k|k) = \hat{x}(k|k-1) + K_\theta(k)\left(z_\theta(k) - H_\theta \hat{x}(k|k-1)\right), \tag{10}$$

$$K_\theta(k) = P_\theta(k|k)(H_\theta)^T(R_\theta)^{-1}, \tag{11}$$

$$(P_\theta(k|k))^{-1} = P^{-1}(k|k-1) + (H_\theta)^T(R_\theta)^{-1}H_\theta, \tag{12}$$

where $\hat{x}_\theta(k)$ is the estimation value of the walking direction at current state. $K_\theta(k)$ is the Kalman gain for the walking direction. $H_\theta$ is the observation matrix of the walking direction. $R_\theta$ is the error covariance of estimation result of the walking direction. $z_\theta$ is the scaling vector of the walking direction at current state. $P_\theta(k)$ is the mean of error value estimation of the walking direction at current state. The measurement update obtained by fusing $\alpha$ and $\theta$ data is:

$$\begin{aligned} \hat{x}_{\alpha+\theta}(k|k) &= \hat{x}(k|k-1) + K_{\alpha|(\alpha+\theta)}(k)(z_\alpha(k) \\ &\quad - H_\alpha \hat{x}(k|k-1)) + K_{\theta|(\alpha+\theta)}(k)(z_\theta(k) \\ &\quad - H_\theta \hat{x}(k|k-1)), && (13) \\ K_{\alpha|(\alpha+\theta)}(k) &= P_{\alpha+\theta}(k|k)(H_\alpha)^T(R_\alpha)^{-1}, && (14) \\ K_{\theta|(\alpha+\theta)}(k) &= P_{\alpha+\theta}(k|k)(H_\theta)^T(R_\theta)^{-1}, && (15) \\ (P_{\alpha+\theta}(k|k))^{-1} &= P^{-1}(k|k-1) + (H_\alpha)^T(R_\alpha)^{-1}H_\alpha \\ &\quad + (H_\theta)^T(R_\theta)^{-1}H_\theta, && (16) \end{aligned}$$

where $\hat{x}_{\alpha+\theta}(k)$ is the estimation value of the fusion at current state. $K_{\alpha|(\alpha+\theta)}(k)$ and $K_{\theta|(\alpha+\theta)}(k)$ are the Kalman gain for the fusion for the body orientation and the walking direction, respectively. $P_{\alpha+\theta}(k)$ is the mean of error value of fusion at current state.

## 4 Experiments and discussion

### 4.1 Experimental setup

#### 4.1.1 Computer and sensor specifications

In all our experiments, we used a computer equipped with specifications: Intel Core i7-7700HQ @ 2.80 GHz, 16 GB RAM, NVidia GeForce GTX 1050 GPU, Windows 10 64-bit Enterprise, Windows 10, Visual C++ 2013 Ultimate developer program, and the OpenCV 3.1.0 and Kinect10 libraries. The specifications of the Kinect sensor are Kinect for Windows V1, USB 2.0 connectivity, and a resolution of 640x480 pixels at a frame rate of 30 Hz (RGB and depth).

#### 4.1.2 Participants

For system testing requirements, we hired three participants who we asked to use three types of clothing, namely shirt or t-shirt, jacket, and hijab as shown in Fig. 12. Each participant in the clothing combination was asked to do several test combinations taking into account seven kinds of distance, 12 orientation angles, and two actions, which are fixed in place or walking.

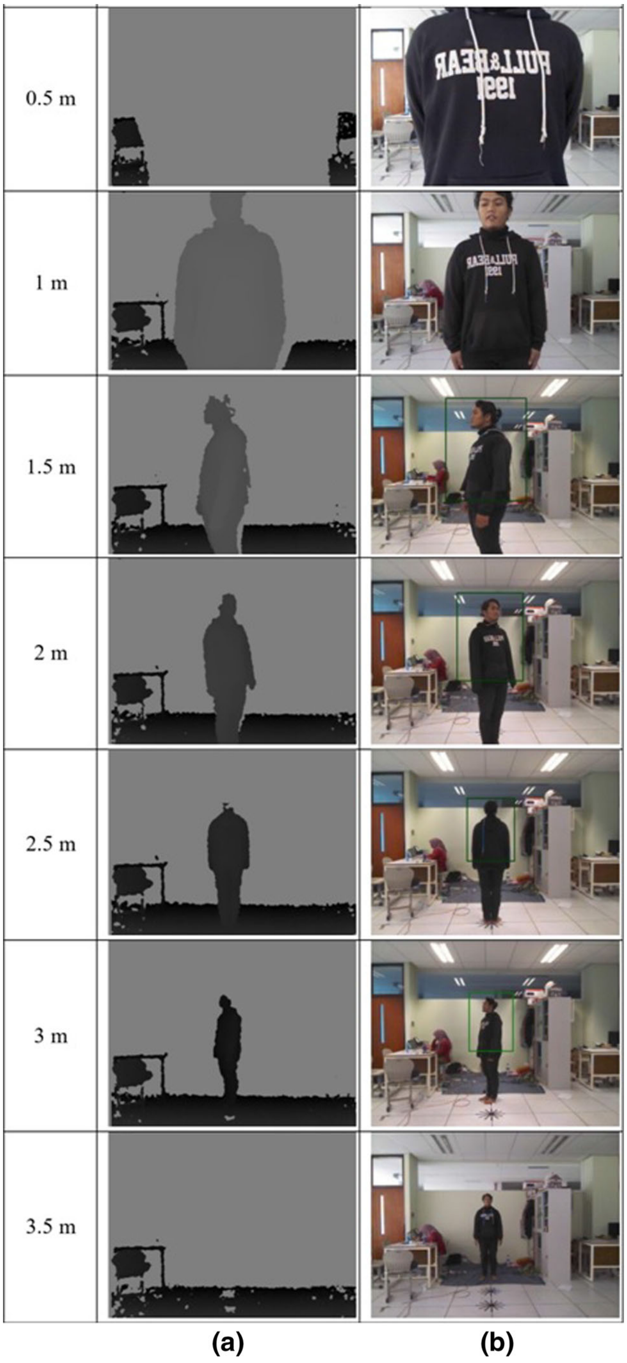


**Fig. 13** Detecting the human upper body from the depth images with distances ranging from 0.5 m up to 3.5 m. **a** the depth image, and **b** the human's upper body detection results on the RGB image

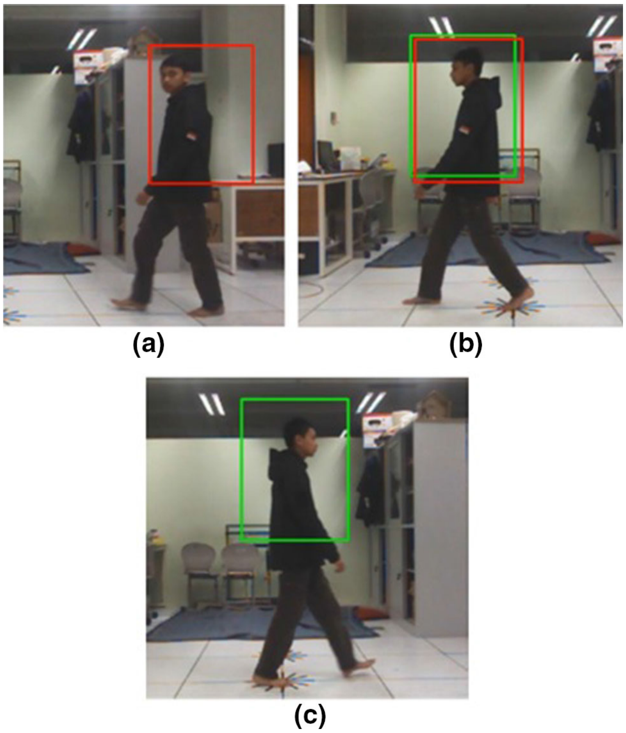


**Fig. 14** Human upper body detection and tracking. **a** the detection, **b** the detection and tracking run together, and **c** the tracker still working even the detector failed to detect human upper body

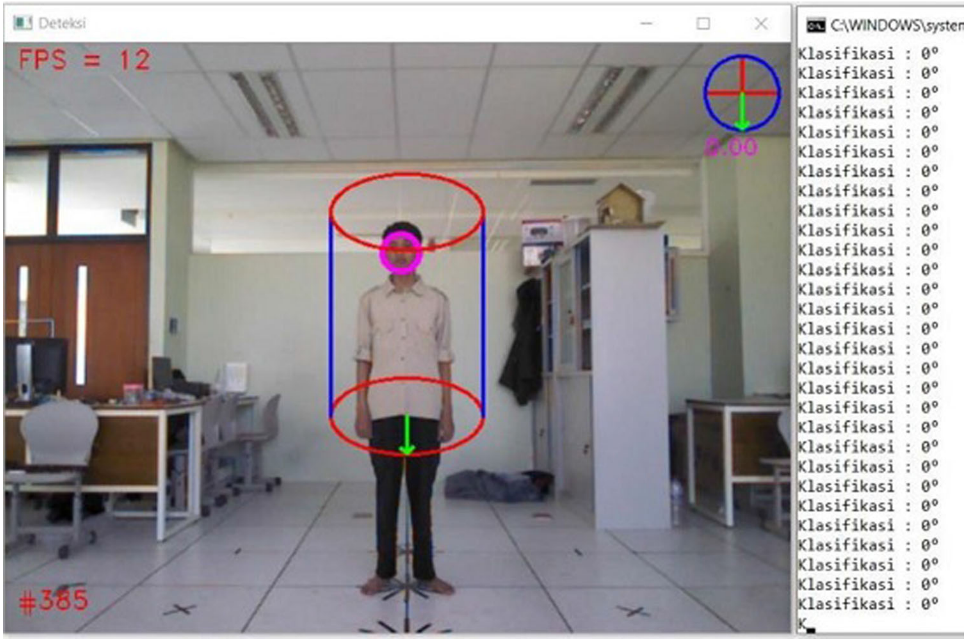
## 4.2 Human upper body detection and tracking

In this section, the evaluation is carried out by asking the participants to stand at certain distances, with various styles of clothing, and to perform different angles of body orientation. The measured distance is a multiple of 0.5 m, starting from 0.5 m to 3.5 m. The resolution of the frame used for the RGB image and the depth image is the same, 640 × 480 pixels. Figure 13 shows the depth image and the human's upper body detection results on the RGB image.

The outcome of the evaluation is that the system can detect the upper body of the subject at a distance of 1.5 m, 2 m, 2.5 m, and 3 m. The object is not observed at a distance of 0.5 m and 1 m because the head is not visible. For positive training data, all data are depth images from the top of the head to the lower abdomen. The data at a distance of 0.5 m are therefore not included in the positive training data so that the upper body is not detected. The human body cannot be tracked at a distance of 3.5 m, as Kinect has limitations in the object detection range above 3 m, so that the human upper body can be effectively detected at a distance of between 1.5 and 3 m.

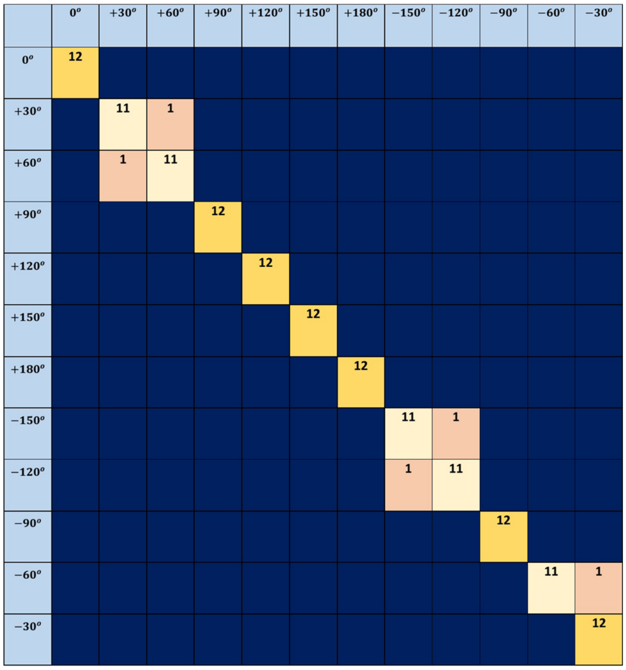
To maintain and improve the results of a human's upper body detection, an object tracking system is needed. We use the Kalman filter to track the position of the human upper body and the size of the bounding box. Figure 14 shows an example of detection and tracking test results. The red box is the result of a human's upper body detection, while the green box is the result of a human's upper body tracking.

From the evaluation results, Fig. 14a shows that only the red box is visible at the system's startup, because, in the initial conditions, the system performs detection first, while the tracking result has not yet appeared because the Kalman filter is still initializing. In Fig. 14b, a red box and a green box appear together to indicate that the tracking system has been running in conjunction with the detection system. In Fig. 14c, a green box still appears which means that the



```
Klasifikasi : 0°
Klasifikasi : 0°
Klasifikasi : 0°
Klasifikasi : 0°
Klasifikasi : 0°
Klasifikasi : 0°
Klasifikasi : 0°
Klasifikasi : 0°
Klasifikasi : 0°
Klasifikasi : 0°
Klasifikasi : 0°
Klasifikasi : 0°
Klasifikasi : 0°
Klasifikasi : 0°
Klasifikasi : 0°
Klasifikasi : 0°
Klasifikasi : 0°
Klasifikasi : 0°
Klasifikasi : 0°
Klasifikasi : 0°
Klasifikasi : 0°
Klasifikasi : 0°
Klasifikasi : 0°
Klasifikasi : 0°
Klasifikasi : 0°
Klasifikasi : 0°
Klasifikasi : 0°
Klasifikasi : 0°
Klasifikasi : 0°
K
```

**Fig. 15** Result of the human upper body orientation at $d = 2.5\,\text{m}$ and $\alpha = 0°$. The green arrow represents the body orientation, while the red ellipse is the roof and bottom of the tube which represents the bounding box area of the detected human upper body in the planar plane



|  | 0° | +30° | +60° | +90° | +120° | +150° | +180° | −150° | −120° | −90° | −60° | −30° |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0° | 12 |  |  |  |  |  |  |  |  |  |  |  |
| +30° |  | 11 | 1 |  |  |  |  |  |  |  |  |  |
| +60° |  | 1 | 11 |  |  |  |  |  |  |  |  |  |
| +90° |  |  |  | 12 |  |  |  |  |  |  |  |  |
| +120° |  |  |  |  | 12 |  |  |  |  |  |  |  |
| +150° |  |  |  |  |  | 12 |  |  |  |  |  |  |
| +180° |  |  |  |  |  |  | 12 |  |  |  |  |  |
| −150° |  |  |  |  |  |  |  | 11 | 1 |  |  |  |
| −120° |  |  |  |  |  |  |  | 1 | 11 |  |  |  |
| −90° |  |  |  |  |  |  |  |  |  | 12 |  |  |
| −60° |  |  |  |  |  |  |  |  |  |  | 11 | 1 |
| −30° |  |  |  |  |  |  |  |  |  |  |  | 12 |

**Fig. 16** Confusion matrix of discrete classification result performed by three different persons

tracker is trying to maintain the bounding box even though the detection system failed to detect the upper body. This shows that the tracking system successfully defended the object and improved the detection results of the human upper body so that later the object would not be easily lost from detection.

## 4.3 Estimating body orientation in static position

### 4.3.1 Classifying body orientation in discrete angles

In this section, experiments are carried out to evaluate the accuracy and precision of the system by knowing the number of correct orientation classifications and how big is the angle errors when the body forms a certain orientation angle. At this stage, we involved three participants using three different types of clothes. We ask each participant to stand 1.5 m away from the camera and face a certain orientation angle according to the marking line on the floor. The angles tested have a step of 30°, starting from 0° to 360°. The experiment was repeated for distances of 2 m, 2.5 m, and 3 m. Figure 15 is an example of the results of testing the orientation of the body orientation at a distance of 2.5 m and $\alpha = 0°$. The performance of the discrete classification of body orientation is shown by confusion matrix in Fig. 16.

The outcomes of the human body orientation evaluation are shown in Table 5. Our system successfully achieved 96.53% of the classification accuracy with the mean absolute of angle error (MAAE) of 1.041°. The highest accuracy of 100% can be achieved at most distances, while the lowest accuracy of 83.33% at $d = 1.5$ m achieved by the third person. This occurs due to variation in body posture because the third person using a hijab (please refer to Fig. 12), so that it has little effect on the contour of the body because it is covered by cloth.

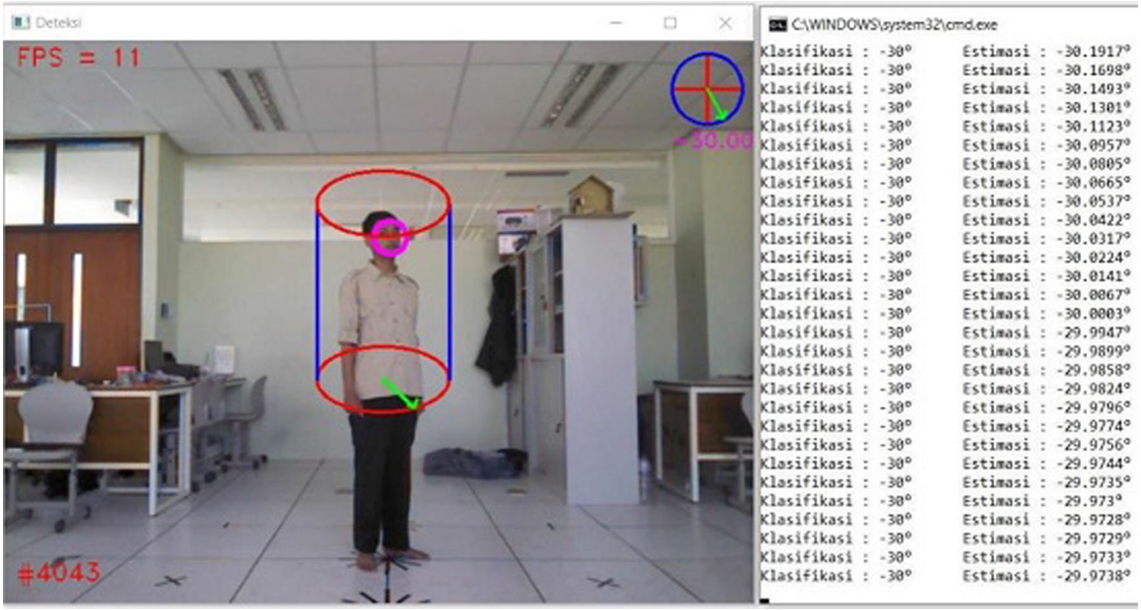
**Table 5** Results of the human body orientation evaluation

| Subject No. | Distance ($d$) (meter) | Number of correct classification | Accuracy (%) | MAAE (°) |
|---|---|---|---|---|
| 1. | 1.5 | 11 of 12 | 91.67 | 2.5 |
|  | 2 | 12 of 12 | 100 | 0 |
|  | 2.5 | 12 of 12 | 100 | 0 |
|  | 3 | 11 of 12 | 91.67 | 2.5 |
| 2. | 1.5 | 12 of 12 | 100 | 0 |
|  | 2 | 12 of 12 | 100 | 0 |
|  | 2.5 | 12 of 12 | 100 | 0 |
|  | 3 | 12 of 12 | 100 | 0 |
| 3. | 1.5 | 10 of 12 | 83.33 | 5 |
|  | 2 | 12 of 12 | 100 | 0 |
|  | 2.5 | 11 of 12 | 91.67 | 2.5 |
|  | 3 | 12 of 12 | 100 | 0 |
| **The average value** |  |  | **96.53** | **1.041** |

### 4.3.2 Estimating body orientation in continuous angles

In this section, we conduct experiments to obtain the accuracy and precision of static body orientation with continuous values. To stabilize detection, we involve tracking using the Kalman filter. Evaluation is done to determine the MAAE of the system by calculating the difference in the value of the actual orientation of the body with the orientation value of the estimated system. We tested the orientation angle with a step of 30° ranging from 0° to 360°. The experiment was carried out by three persons at $d = 1.5$ m, 2 m, 2.5 m, and 3 m. Figure 17 shows an example of static continuous orientation precision test results at $d = 2.5$ m and $\alpha = -30°$.

The outcomes of the human body continuous orientation evaluation are shown in Fig. 18. The MAAE values are the average difference between the ground truth of body orientation angle with the estimated continuous orientation angle from the 12 orientation angles. Our system successfully achieved 0.097° of the MAAE in total. By achieving the MAAE value of less than 0.1°, it indicates that our continuous orientation estimation system is working effectively. The best MAAE value of 0.0003° is achieved by the second person with $d = 3$ m, while the worst MAAE value of 0.2995° is achieved by the third person with $d = 3$ m. This occurs because of differences in body posture that is different from other subjects, and the distance of objects that are too far away from the camera.



```
Klasifikasi : -30°      Estimasi : -30.1917°
Klasifikasi : -30°      Estimasi : -30.1698°
Klasifikasi : -30°      Estimasi : -30.1493°
Klasifikasi : -30°      Estimasi : -30.1301°
Klasifikasi : -30°      Estimasi : -30.1123°
Klasifikasi : -30°      Estimasi : -30.0957°
Klasifikasi : -30°      Estimasi : -30.0805°
Klasifikasi : -30°      Estimasi : -30.0665°
Klasifikasi : -30°      Estimasi : -30.0537°
Klasifikasi : -30°      Estimasi : -30.0422°
Klasifikasi : -30°      Estimasi : -30.0317°
Klasifikasi : -30°      Estimasi : -30.0224°
Klasifikasi : -30°      Estimasi : -30.0141°
Klasifikasi : -30°      Estimasi : -30.0067°
Klasifikasi : -30°      Estimasi : -30.0003°
Klasifikasi : -30°      Estimasi : -29.9947°
Klasifikasi : -30°      Estimasi : -29.9899°
Klasifikasi : -30°      Estimasi : -29.9858°
Klasifikasi : -30°      Estimasi : -29.9824°
Klasifikasi : -30°      Estimasi : -29.9796°
Klasifikasi : -30°      Estimasi : -29.9774°
Klasifikasi : -30°      Estimasi : -29.9756°
Klasifikasi : -30°      Estimasi : -29.9744°
Klasifikasi : -30°      Estimasi : -29.9735°
Klasifikasi : -30°      Estimasi : -29.973°
Klasifikasi : -30°      Estimasi : -29.9728°
Klasifikasi : -30°      Estimasi : -29.9729°
Klasifikasi : -30°      Estimasi : -29.9733°
Klasifikasi : -30°      Estimasi : -29.9738°
```

**Fig. 17** Result of the human upper body continuous orientation at $d = 2.5$ m and $\alpha = -30°$

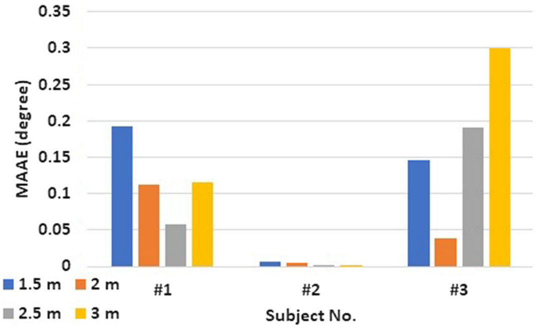


**Fig. 18** Mean absolute errors of the human body continuous orientation that were obtained by three participants at four relative distances

### 4.3.3 Comparing the performance with the other methods

In this section, we compared our method with the other methods proposed by the other researchers. We have compared our work with the works in [5,9–11,18]. The results of the comparison may not be truly fair due to the different types of sensors, data sets, and step of angle used. At least, however, these results will give us an idea of the performance of each method, particularly for use in the actual implementation. Table 6 shows the comparison results of the human body orientation in static position.

According to the results presented in Table 6, we compared the accuracy and precision of the orientation estimation results for each method. Not all methods compare the accuracy and precision simultaneously, such as the HOG+LBP+PLS +movements [9] and the Depth Weighted HOG [18], they do not display data regarding the precision of the orientation angle. The other three, including our method, test the precision of the orientation angle. As a result, our proposed method can outperform Shape +Motion [5] and Keypoints [11] with a high degree of precision, where the

**Table 6** Comparison results of the human body orientation evaluation in static position

| Method | Sensor Type | Dataset | Step (°) | Accuracy (%) | MAE (°) |
|---|---|---|---|---|---|
| HOG+LBP+PLS +Movements [9] | Monocular Camera | TUD | 45 | 64.00 | N/A |
| Depth Weighted HOG [18] | ToF RGB Camera | Private (Real) | 45 | 78.90 | N/A |
| Shape+Motion [5] | 2D LRF | Private (Real) | 10 | 93.66 | 0.83 |
| Keypoints [11] | Monocular Camera | TUD and 3DPeS | 45 | 96.80 | 15.30 |
| CNN [10] | Monocular Camera | Private* (Real, RGB) | 30 | 77.78 | 37.50 |
| CNN [10] | Monocular Camera | Private* (Real, Depth) | 30 | 43.06 | 49.59 |
| **Ours (discrete)** | Kinect | Private* (Real, Depth) | 30 | 96.53 | 1.041 |
| **Ours (continuous)** | Kinect | Private* (Real, Depth) | 30 | **100** | **0.097** |

(*) indicates the methods are compared using our dataset

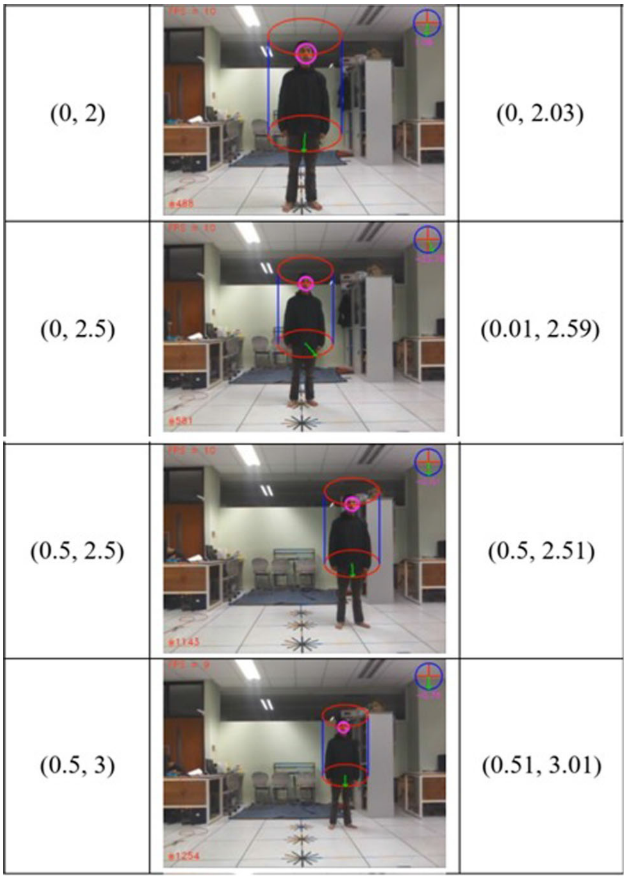


**Fig. 19** Examples of the body position estimation using Kinect. The first column is the ground truth positions in the real world coordinate. The second column is the detected human upper body. The third column is the estimated body positions in the real world coordinate

angular error is less than 0.1°. As for the level of accuracy, the Keypoints method can be the best with an accuracy of 96.8%. However, it should be noted that this method is tested using a dataset, not using real data.

For a fairer test in terms of using the same dataset and testing protocol, we replicated the CNN method proposed by Kohari et al [10]. We retrained and tested with the same dataset we used. Unfortunately the results obtained from the replication are quite far behind from our proposed method.

## 4.4 Estimating body orientation in motion

### 4.4.1 Estimating human body position in world coordinate

In experiments that require the position of people moving from one place to another, then estimating the position of the human body based on depth images is a crucial initial step. The test is carried out with several coordinates in the range of the Kinect sensor. Figure 19 shows examples of the estimated position results from the Kinect camera. In the first column, it contains ground truth position data on real coordinates in meters, whereas in the third column, it contains position estimation data using Kinect depth data at real coordinates in meters. Table 7 shows the results of the estimated body position and error value. The error value is calculated based on the distance between the actual position (ground truth) and the estimation results.

**Table 7** Results of the body position estimation

| Ground truth position (x,y) (meters) | Estimated position (x,y) (meters) | Error (meters) |
|---|---|---|
| (0, 1.5) | (−0.03, 1.52) | 0.036 |
| (0, 2) | (0, 2.03) | 0.03 |
| (0, 2.5) | (0.01, 2.59) | 0.09 |
| (0, 3) | (−0.02, 3.07) | 0.073 |
| (0.5, 2) | (0.5, 2.02) | 0.02 |
| (0.5, 2.5) | (0.5, 2.51) | 0.01 |
| (0.5, 3) | (0.51, 3.01) | 0.014 |
| (1, 2.5) | (1.03, 2.57) | 0.076 |
| (1, 3) | (1.09, 3.01) | 0.09 |
| (−0.5, 2) | (−0.56, 2.02) | 0.063 |
| (−0.5, 2.5) | (−0.58, 2.51) | 0.08 |
| (−0.5, 3) | (−0.56, 2.99) | 0.06 |
| (−1, 3) | (−1.09, 2.99) | 0.09 |
| **The average of error position** | | **0.056** |

**Table 8** Results of the body orientation estimation when moving

| Subject | Pattern | #Frame | #Correct | Acc. (%) | MAAE (°) |
|---|---|---|---|---|---|
| 1 | Square | 283 | 252 | 89.05 | 4.93 |
| | Diamond | 284 | 231 | 81.34 | 8.4 |
| 2 | Square | 305 | 269 | 88.197 | 5.31 |
| | Diamond | 361 | 285 | 78.947 | 9.47 |
| 3 | Square | 466 | 391 | 83.90 | 7.24 |
| | Diamond | 326 | 266 | 81.59 | 8.28 |
| **The average value** | | | | **83.84** | **7.27** |

From the test results, we obtained a relatively small average error value, which is 0.056 m or 5.6 cm. The biggest error occurred at position (0, 2.5), (1, 3), and (−1, 3) which is 0.09 m. Large errors in position (1, 3) and (−1, 3) are caused by body position that is close to the Kinect's field of view (FoV); the right and left bounds, respectively, while the smallest error occurs at position (0.5, 2.5). According to this experiment, we are very confident that our method is very feasible to be used to estimate the position of displacement in the real world coordinate, as long as it is within the camera's FoV limit.

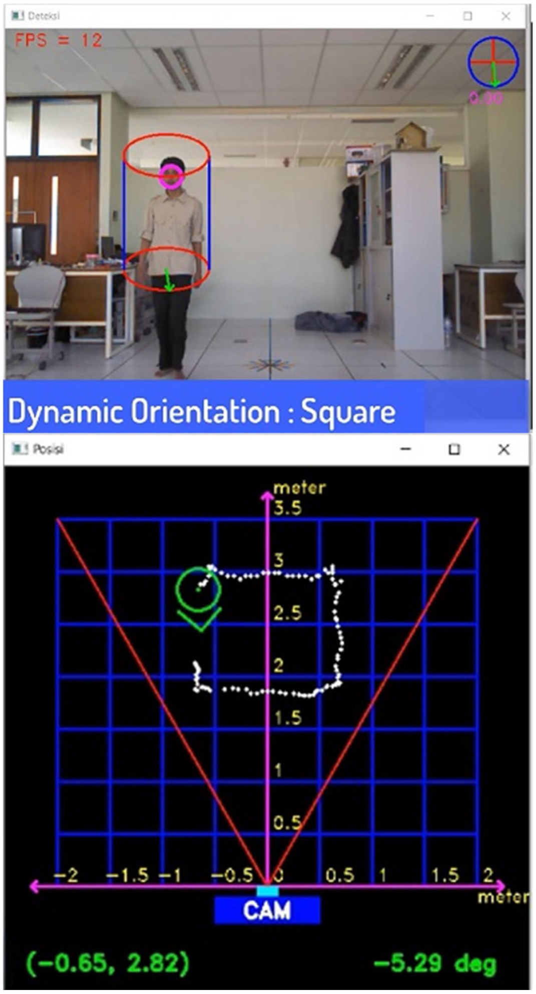


Fig. 20 Result of the human body continuous orientation when walking in the square pattern. Each participant was asked to walk following the square pattern with the pattern's midpoint at a distance of 2.5 m. The trajectory and body orientation results are recorded to measure the performance of the position detector and orientation estimator

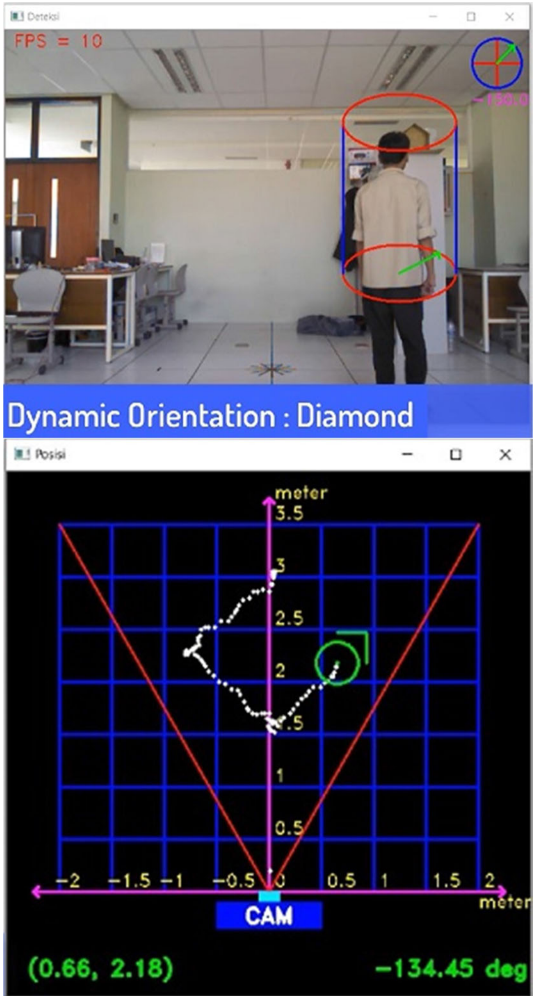


Fig. 21 Result of the human body continuous orientation when walking in the diamond pattern. Each participant was asked to walk following the diamond pattern with the pattern's midpoint at a distance of 2.5 m. The trajectory and body orientation results are recorded to measure the performance of the position detector and orientation estimator

Table 9 Comparison results between our proposed method with LRF-based method

| Method | Motion Pattern | MAAE (°) |
|---|---|---|
| Shape+Motion [5] | Forward | 7.28 |
| | Circular | 11.2 |
| Ours | Square | 5.82 |
| | Diamond | 8.71 |

### 4.4.2 Estimating body orientation while moving

In this section, we examined our proposed method of estimating body orientation when the target person is walking. We employed three participants as models to demonstrate the walking movement by creating a certain pattern. Accuracy evaluation is performed by automatically counting the number of correct angles of orientation (with a tolerance of ±20°) to the number of frames generated in each trial sequence. The patterns to be performed by each participant are a square and a diamond shape. The square pattern was chosen to test the output of the system at several angles, such as 0°, +90°, +180°, and −90°, while the diamond pattern is used to measure device output at several angles, namely +45°, +135°, −135°, and −45°. Data were recorded for two rounds per participant. Figures 20 and 21 show the examples of the results of a continuous orientation estimation test for the square pattern displacement and the diamond pattern displacement, respectively.

Table 8 shows the results of the evaluation of the continuous orientation estimate when each episode of the experiment is completed by the target person. From the test results, we achieved an overall accuracy of 83.84%. With the best

**Table 10** Comparison results of the computation time between our proposed method with others

| Method | Author's machine | Our machine | Specifications |
|---|---|---|---|
| Liu [15] | 338 ms | - | i7 2.3 GHz CPU,<br>8 GB RAM,<br>128 GB SSD,<br>Windows 7 |
| Chen [13] | 40 ms | | Intel Core-2 Duo<br>CPU 1.8 GHz<br>1 GB RAM<br>GPU NVidia<br>GeForce 8600 GT |
| Lewandowski [16] | 74.46 ms | - | GPU NVidia<br>GTX 1080Ti |
| Kohari [10] | 8.73 ms | | GPU GeForce<br>Titan X Pascal |
| Kohari [10] | | 38 ms | i7 2.8 GHz CPU,<br>16 GB RAM,<br>128 GB SSD,<br>GPU MX940,<br>Windows 10 |
| Kohari [10] | | 266 ms | i7 2.8 GHz CPU,<br>16 GB RAM,<br>128 GB SSD,<br>Windows 10 |
| **Ours** | | 83.33 ms | i7 2.8 GHz CPU,<br>16 GB RAM,<br>128 GB SSD,<br>Windows 10 |

accuracy of 89.05% when the first participant carried out a rectangle-shaped displacement, whereas the lowest accuracy was 78.95% when the third participant carried out a diamond-shaped motion. This is due to the decreased accuracy when participants make a turn. As he turns, the speed at which the angle changes is slower than the body's speed as he turns. This causes the estimated value produced by the Kalman filter to take time to change the actual orientation value.

### 4.4.3 Comparing the performance with the other method

To prove its performances, we compared our body orientation estimation method with the method proposed in paper [5]. Both approaches used body contour features that were evaluated for practical implementation. The description of the comparative results obtained by the two methods is shown in Table 9. Although the two methods cannot be fairly compared based on their different patterns of movement, at least these results will provide a summary of the performance of our proposed method.

The results shown in [5] were evaluated with motions that have minimal sharp angles, whereas the patterns we use have several sharp angle patterns that are very susceptible to estimation errors when people are in the corner. However, with the results of the MAAE that turned out to be smaller, we conclude that the method we are proposing is very promising to be applied in a real system.

## 4.5 Computation time

Besides accuracy and MAAE, we also measure the computational time required by our method to complete the entire process in it. The whole process starts from capturing RGB images and depth, detection of the upper body with HOG, taking body contour features and normalization, face detection, classification, and tracking with Kalman filters varying from 100 msec to 83.33 msec or 10–12 fps. We also compared our computation time with the other methods as shown in Table 10.

Based on Table 10, it can be seen that some of the previous works [10,13,16] have better computation time than our method because they used a GPU. For methods that did not use a GPU [10,15], our proposed method is superior with a computation time below 100 msec.

# 5 Conclusion

We have presented our proposed method for estimating body orientation. The combination of the three one-dimensional distance feature-based signals which reflect the body's surface contours taken by Kinect camera was proved to be effective to characterize the difference of each body orientation. This combination significantly strengthens our feature to estimate the body orientation. improved the estimator performances when using to estimate body orientation in motion. Based on the experiments, our method outperforms the other image-based baseline methods. A comparison with LRF sensor-based method exhibits our method is also comparable.

The experiments in a real scene environment show that our method is very promising and is applicable for an online application. However, minimizing the step angle is crucial to improve the continuous estimation results. Other than that, improving the computation time for real-time application by utilizing middleware-based programmings such as Robotic Operating System (ROS) or Open Robotic Tool Middleware (OpenRTM) will also be our next focus in the future.

**Acknowledgements** The author would like to thank all of our SVG laboratory members as their help and participation in our experiments to evaluate system performances.

## Declarations

**Conflict of interest** The authors declare that they have no conflict of interest.

## References

1. Li, S., Zhang, L., Diao, X.: Deep-learning-based human intention prediction using RGB images and optical flow. J. Intell. Rob. Syst. **97**, 95–107 (2020)
2. Saeed, A., Al-Hamadi, A.: Boosted human head pose estimation using kinect camera, In: IEEE International Conference on Image Processing (ICIP) (2015)
3. Dewantara, B.S.B., Miura, J.: Estimating head orientation using a combination of multiple cues. IEICE Trans. Inf. Syst. **E99–D**(6), 1603–1613 (2016)
4. Glas, D.F., Miyashita, T., Ishiguro, H., Hagita, N.: Laser-based tracking of human position and orientation using parametric shape modeling. Adv. Robot. **23**(4), 405–428 (2009)
5. Shimizu, M., Koide, K., Ardiyanto, I., Miura, J., Oishi, S.: LIDAR-based body orientation estimation by integrating shape and motion information, In: Proceedings of 2016 IEEE International Conference on Robotics and Biomimetics (ROBIO), pp. 1948–1953 (2016)
6. Tepencelik, O.N., Wei, W., Chukoskie, L., Cosman P.C., Dey, S.: Body and Head Orientation Estimation with Privacy Preserving LiDAR Sensors, Semantic Scholar (2021)
7. Baltieri, D., Vezzani, R., Cucchiara, R.: People orientation recognition by mixtures of wrapped distributions on random trees. Proc. Eur. Conf. Computer Vis. **7576**, 270–283 (2012)
8. Weinrich, C., Vollmer, C., Gross, H.: Estimation of human upper body orientation formobile robotics using an SVM decision tree on monocular images, In: International Conference on Intelligent Robots and Systems, pp. 2147–2152 (2012)
9. Ardiyanto, I., Miura, J.: Partial least squares-based human upper body orientation estimation with combined detection and tracking. Image Vis. Comput. **32**(11), 904–915 (2014)
10. Kohari, Y., Miura, J., Oishi, S.: CNN-based Human Body Orientation Estimation for Robotic Attendant, Workshop on Robot Perception of Humans, IAS-15, (2018)
11. Yu, D., Xiong, H., Xu, Q., Wang, J., Li, K.: Continuous pedestrian orientation estimation using human keypoints, In: 2019 IEEE International Symposium on Circuits and Systems (ISCAS) (2019)
12. Wu, C., Chen, Y., Luo, J., Su, C.C., Dawane, A., Hanzra, B., Deng, Z., Liu, B., Wang, J.Z., Kuo, C.H.: MEBOW: monocular estimation of body orientation in the wild. IEEE/CVF Conf. Comput. Vis. Pattern Recogn. (CVPR) **1**, 3448–3458 (2020)
13. Chen, L., Panin, G., Knoll, A.: Human body orientation estimation in multiview scenarios. ISVC Part II, LNCS **7432**, 499–508 (2012)
14. Choi, J., Lee, B.J., Zhang, B.T.: Human body orientation estimation using convolutional neural network, In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), (2016)
15. Liu, W., Zhang, Y., Tang, S., Tang, J., Hong, R., Li, J.: Accurate estimation of human body orientation from RGB-D sensors. IEEE Trans. Cybern. **43**(5), 1442–1452 (2013)
16. Lewandowski, B., Seichter, D., Wengefeld, T., Pfennig, L., Drumm, H., Gross, H.M.: Deep orientation: fast and robust upper body orientation estimation for mobile robotic applications, In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 441–448 (2019)
17. Saputra, R.W.A., Dewantara, B.S.B., Pramadihanto, D.: Human body's orientation estimation based on depth image, In: The 21th IEEE International Electronic Symposium, pp. 100–106 (2019)
18. Shinmura, F., Deguchi, D., Ide, I., Murase, H., Fujiyoshi, H.: Estimation of human orientation using coaxial RGB-depth images, In: VISAPP 2015 - 10th International Conference on Computer Vision Theory and Applications (VISIGRAPP), pp. 113–120 (2015)
19. Dewantara, B.S.B., Ardilla, F., Thoriqy, A.A.: Implementation of depth-HOG based human upper body detection on a mini PC using a low cost stereo camera, In: International Conference of Artificial Intelligence and Information Technology (ICAIIT), (2019)
20. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection, In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, (2005)
21. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. ACM Trans. Intell. Syst. Technol. **2**(27), 1–27 (2011)
22. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features, In: International Conference on Computer Vision and Pattern Recognition, (2001)
23. Kim, Y., Bang, H.: Introduction to Kalman filter and its applications, IntechOpen, pp. 1–16 (2018)
24. Caron, F., Duflos, E., Pomorski, D., Vanheeghe, P.: GPS/IMU data fusion using Multisensor Kalman filtering: introduction of contextual aspects. Inf. Fusion **7**(2), 221–230 (2006)

**Bima Sena Bayu Dewantara** is a lecturer in the department of computer engineering from Politeknik Elektronika Negeri Surabaya, Indonesia. He received the B.Eng degree in information technology from Politeknik Elektronika Negeri Surabaya, Indonesia, the M.Eng. degree in electrical engineering from Sepuluh Nopember Institute of Technology, Indonesia, and the Dr. Eng. degree in computer science and engineering from Toyohashi University of Technology, Japan, in 2004, 2010, and 2016, respectively. Currently, he works in the major field of autonomous intelligent systems that cover pattern recognition, computer vision, machine learning, signal processing, robotics system, and human-machine interaction. Until now, he has published more than 20 international and national journal papers and more than 50 international and national paper proceedings. He received The Best Achievement Award from Toyohashi University of Technology, Japan, in 2016 and also received some Best Paper Awards in 2016, 2018, 2019, 2020, and 2021.

**Rizka Wahyu Aditiya Saputra** received the B.Eng. degree in computer engineering from Electronic Engineering Polytechnic Institute of Surabaya, Indonesia, in 2019. Then, he joined the Information Technology Consultant Company in Surabaya, Indonesia, as software developer. Since December 2020, he has been a Computer Expert for e-Government specialty at the Department of Communication and Informatics in Tulungagung, Indonesia. His research interests include computer vision, robotics system, data analysis, and software development.

**Dadet Pramadihanto** is a professor in robotics and computer vision at the Politeknik Elektronika Negeri Surabaya. He received a bachelor's degree in control systems engineering from Sepuluh Nopember Institute of Technology and receive Master of Engineering and Doctor of Philosophy degrees in Computer vision from Osaka University, Japan, in 1997 and 2003, respectively. He is now lecturer in computer engineering department and head of Robotics and Intelligent Systems Centre (RoISC) at the Politeknik Elektronika Negeri Surabaya. His research areas of interest include development of humanoid robots, computer vision in robotics, underwater and aerial robots, and cyber-physical systems.