

# Human Posture Recognition Based on Images Captured by the Kinect Sensor

Regular Paper

---

Wen-June Wang<sup>1\*</sup>, Jun-Wei Chang<sup>1</sup>, Shih-Fu Haung<sup>1</sup> and Rong-Jyue Wang<sup>2</sup>

1 National Central University, Taoyuan City, Taiwan

2 National Formosa University, Yunlin County, Taiwan

\*Corresponding author(s) E-mail: wjwang@ee.ncu.edu.tw

Received 12 September 2015; Accepted 15 December 2015

DOI: 10.5772/62163

© 2016 Author(s). Licensee InTech. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

---

## Abstract

In this paper we combine several image processing techniques with the depth images captured by a Kinect sensor to successfully recognize the five distinct human postures of sitting, standing, stooping, kneeling, and lying.

The proposed recognition procedure first uses background subtraction on the depth image to extract a silhouette contour of a human. Then, a horizontal projection of the silhouette contour is employed to ascertain whether or not the human is kneeling. If the figure is not kneeling, the star skeleton technique is applied to the silhouette contour to obtain its feature points. We can then use the feature points together with the centre of gravity to calculate the feature vectors and depth values of the body. Next, we input the feature vectors and the depth values into a pre-trained LVQ (learning vector quantization) neural network; the outputs of this will determine the postures of sitting (or standing), stooping, and lying. Lastly, if an output indicates sitting or standing, one further, similar feature identification technique is needed to confirm this output. Based on the results of many experiments, using the proposed method, the rate of successful recognition is higher than 97% in the test data, even though the subjects of the experiments may not have been facing the Kinect sensor and may have had different statures. The proposed

method can be called a “hybrid recognition method”, as many techniques are combined in order to achieve a very high recognition rate paired with a very short processing time.

**Keywords** Posture Recognition, Neural Network Application, Feature Extraction, Image Processing

---

## 1. Introduction

In recent years, methods of human posture recognition have been studied in a range of different papers. In general, these methods can be divided into two types. The first type involves wearable sensors, which are put on the body or clothes of a human to measure certain values, such as the positions of limbs and the slope degree of the body. For instance, one study asked a participant to wear a garment with strain sensors to recognize 27 upper body postures [1]. In [2] and [3], the authors proposed a smart shirt system (SMASH) with acceleration sensors to recognize 21 human exercise postures. A waist-mounted triaxial accelerometer system was developed in [4] to classify human movement status. A wireless acceleration measuring system to monitor a human’s activity volume and recognize emergent situations was built in [5]. However, a disadvantage of all five of these studies was that the wearable sensors and

the accompanying batteries that the participants were required to wear can be a source of discomfort or inconvenience for them.

The other type of method used to recognize posture information is based on captured images of a human body. Some posture features can be represented by specifically coloured markers on the human torso and limbs. By recognizing the relative positions of the coloured markers, human postures can be recognized using the methods presented in [6, 7] and [8]. However, wearing coloured markers can be just as uncomfortable as wearing sensor devices.

Many studies have used image processing techniques to extract features from images of a human, using those features to identify the posture. More than 10 parameters (lengths and the largest widths of the upper and lower body, etc.) were used in [9] to recognize human postures including standing, sitting, kneeling and stooping. A 3D human-body-posture recognition method was proposed in [10] and [11] in which horizontal and vertical projections of a human body were extracted and compared to the corresponding projections of predefined 3D human posture models; this enabled the postures of standing, sitting, lying and stooping to be recognized. The human skeleton was analysed geometrically to produce posture classification results in [12, 13] and [14]. A segmentation algorithm using deformable triangulation or a set of Gaussian mixture models was proposed in [15-18] to divide the posture into different body parts. Moreover, in [19] a number of heuristic rules based on body-shape characteristics and skin-colour features were used to estimate five significant points, namely the tips of both hands, both feet, and the head of a human silhouette contour. The authors of [20] used entropy measurement as an underlying feature and a modified Hausdorff distance to evaluate the similarities between the posture which was being recognized and the posture template database. A temporal difference image sensor was used in [21] to extract the size and position of invariant line features, and then a Hausdorff distance classifier was employed to measure the similarities of those features against a library of objects. In [22], the authors extracted features using a discrete Fourier transform and then used a neural fuzzy network to classify the human body postures. In [23], the authors used a Support Vector Machine (SVM) to classify human postures from images captured by a time-of-flight sensor. The study presented in [24] applied height and width ratios and horizontal and vertical projections as fuzzy logic inputs for posture recognition. Some studies have used a Kinect sensor to recognize human postures; for instance, the authors of [25] presented a method which uses histograms of 3D joint locations from Kinect depth maps and discrete HMM (hidden Markov model) to achieve human posture recognition. To recognize the four human postures standing, sitting, lying and bending, a method was proposed in [26] based on the human skeleton captured by a Kinect

sensor. The authors of [27] recognized three human gestures from the vectors of 20 body-joint positions captured by a Kinect sensor. A method was developed in [28] that was based on colour and depth information gathered from similar sensor. Implementing a multilayer framework to understand human activity, in [29] a Kinect sensor was used to acquire a D-RGB-based skeleton tracking output for human activity recognition. In [30], SVM was applied to classify different postures by nine features, including forearm and thigh, as captured by a Kinect sensor. All of the above studies extracted features from images and used various classifiers to identify the different postures. Recognition rate, number of postures successfully recognized, computation time, and cost of the devices should be of concern for all proposed recognition methods.

In this paper, a new posture recognition method is proposed. The method uses only two devices to achieve its function: a laptop computer and a Kinect sensor. The Kinect sensor consists of a depth sensor, an RGB camera, a multi-array microphone and a motorized tilt [31]. The depth sensor is composed of an infrared ray emitter and a monochrome CMOS sensor to capture depth images with a resolution of 320×240 pixels; the RGB camera is used to capture colour images with a resolution of 640×480 pixels. The multi-array microphone can be used to receive the sound signal, but it will not be used in this study. The motorized tilt can adjust the Kinect sensor's elevation angle. The USB port is used for communication between the laptop computer and the Kinect sensor. The laptop computer is an Intel i5-520 running at 2.4GHz with 4G bytes DRAM. The image processing techniques used encompass the horizontal and vertical projection, star skeleton, LVQ neural network and image processing techniques. Five human postures, standing, sitting, stooping, kneeling, and lying, will be recognized. The reason for selecting these five postures is that they are the general and basic postures of the human form. Conclusions about other postures not mentioned here may be extrapolated from the gained results.

This study contributes to research about automatic home care systems. Elderly people who live alone can often benefit from a robot to provide home care services. These robots must have an ability to recognize the person's postures in normal and dangerous situations, in order to send accurate reports to the care centre.

The main contributions of this paper are as follows. Only one Kinect sensor is used, so the participant does not need to wear any sensors on their body. Because we are using the Kinect depth sensor, the captured image is unaffected by illumination of the environment, shadows, or similarities in the colour of the participant's clothing and that of the background. Three posture recognition methods involving body width and height ratio, neural network and length ratio are combined to recognize total five postures even when the subjects are facing in different directions.

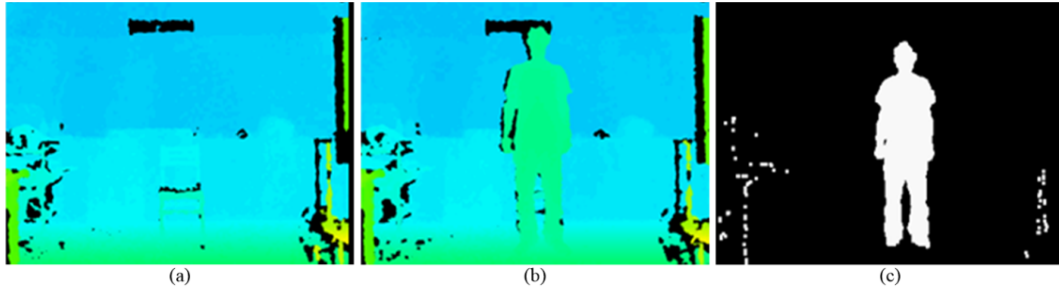


Figure 1. (a) The background depth image. (b) The depth image with a human. (c) The human silhouette.

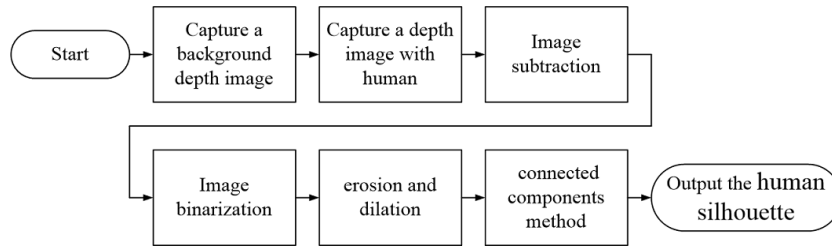


Figure 2. The flow chart of the human silhouette segmentation

Since it is the fusion of many techniques that helps the method achieve a very high recognition rate in a very short processing time, the proposed method can be called a “hybrid recognition method”. Note that this paper does not use Kinect SDK software in the recognition process; this is in contrast to the studies presented in [25-27], which all used the SDK skeleton to recognize postures. Comparisons between the results of the proposed method and those in [25-27] will be discussed in Section 4.

This paper is organized as follows. Section 2 introduces the techniques used for extracting the body posture features. Section 3 describes the training of the LVQ neural network and a final identification method for human-body-posture recognition. Then, the experimental results are shown and a discussion provided in section 4. The final section presents a conclusion.

## 2. Depth Image Processing

In this study, five human postures, standing, sitting, stooping, kneeling, and lying, will be recognized. Several image processing techniques will be introduced and implemented.

### 2.1 Human Silhouette Segmentation

First, the Kinect sensor captures a background depth image without any humans. Next, it captures one more image with a human and subtracts the current depth image from the background depth image to get the subtractive image. The subtraction result is then binarized to create a binary image in which black pixels denote background and white pixels are foreground. Then, erosion and dilation are applied several times to repair the imperfections of the human silhouette and to remove noise. The above process

is demonstrated in Figures 1(a), (b) and (c). If the noise is not cleaned completely, the connected components method is applied to extract the largest region of white pixels, which is regarded as the human silhouette in the binary image as shown in Figure 1(c). The whole process of human silhouette segmentation is shown in Figure 2. Since the effective detection range of the Kinect sensor is between 2 m and 4 m, the human subject should stand inside this range. The size of the segmented human silhouette is at least 3000 pixels in our experiments, so 3000 is set as a threshold to judge whether the human subject is in the detection range or not. If the size of the segmented human silhouette is not larger than 3000 pixels, the following processes will not start.

There are two advantages to capturing the human image using the Kinect sensor. One is related to the influences of illumination. The shadow effects of the subject can be eliminated, since the Kinect sensor can be considered a distance measurement sensor. The captured depth image consists of only a set of distance values between the sensor and the measured objects in the sensing range. There is no illumination information in the depth image. The other advantage is that there is no colour information in the depth image. If there is a white object in the background, and the human wears white clothes in the image captured by a regular camera, then background subtraction will result in an incomplete silhouette contour of the human’s body which cannot be used to recognize the human’s postures. In order to obtain a clear and complete human silhouette, the Kinect sensor can therefore be a highly useful tool.

### 2.2 Feature Extraction

Since the extracted features form the entirety of the data from which the postures will be recognized, they are of

great importance in this process. An overview of some of these features follows.

### 2.2.1 The ratio of the upper and lower human body

First, the silhouette's centre of gravity must be calculated. The silhouette is divided into the upper and lower body based on the centre of gravity, so that the upper body is the part of the silhouette above the centre of gravity and the lower body is the part below the centre of gravity. The centre of gravity can be calculated by (1).

$$\begin{cases} x_c = \frac{1}{N_c} \sum_{i=1}^{N_c} x_i \\ y_c = \frac{1}{N_c} \sum_{i=1}^{N_c} y_i \end{cases} \quad (1)$$

where  $(x_c, y_c)$  is the coordinate of the centre of gravity inside the silhouette.  $N_c$  is the total number of white pixels, and  $x_i$  and  $y_i$  are x-axis and y-axis values of the  $i$ -th pixel inside the silhouette, respectively. The red point in Figure 3 is the silhouette's centre of gravity.



Figure 3. The centre of gravity of a clean human silhouette

Then, the body width can be obtained by computing the horizontal projection histogram for each row of pixels inside the silhouette from top to the bottom. According to the position of the centre of gravity, the maximum values of the projection histogram on the upper and lower body can be found respectively, as shown in Figure 4. The ratio between the maximum value of the projection histogram on the upper body and that on the lower body is calculated from equation (2).

$$K_p = \frac{N_L}{N_U} \quad (2)$$

where  $N_U$  is the maximum upper-body width value and  $N_L$  is the maximum lower-body width value;  $K_p$  is the

ratio, which can be a feature value of human posture. It should be noted that the lateral kneeling posture is very different from other postures in terms of the ratio of upper and lower body, as it is much larger. Therefore, the ratio is especially useful when recognizing the kneeling posture. Based on the experiments, it is found that when the human is kneeling, his feature value  $K_p$  is the largest among all of the postures, as shown in Figure 5. To distinguish the kneeling posture, a threshold value is given; when  $K_p \geq 1.4$ , the human's posture is judged to be the lateral kneeling posture. The selection of the threshold value 1.4 will be explained in section 3 below.

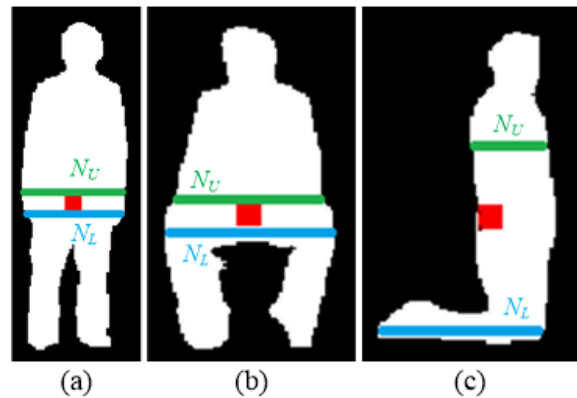


Figure 4. The maximum widths of upper body and lower body for different postures. (a) Standing, (b) sitting, and (c) kneeling.

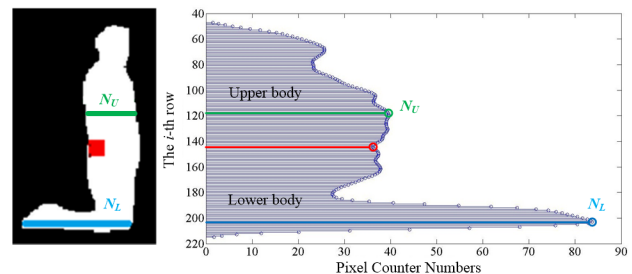


Figure 5. The horizontal projection histogram of the lateral kneeling posture

### 2.2.2 The establishment of the feature vectors

Since the centre of gravity of the human silhouette is known, the distance between the centre of gravity and the edge contour of the human silhouette can be calculated by equation (3).

$$d_i = \sqrt{(x_i^e - x_c)^2 + (y_i^e - y_c)^2} \quad (3)$$

where  $d_i$  denotes the distance values between  $(x_i^e, y_i^e)$  (any point on the edge contour) and the centre of gravity  $(x_c, y_c)$ . The calculation of distance values starts from the left-most edge point and moves in a clockwise direction to the end point, which is near the start point as shown in

Figure 6. Then, a sequence curve of distance values  $d_i$  is obtained as shown in Figure 7.

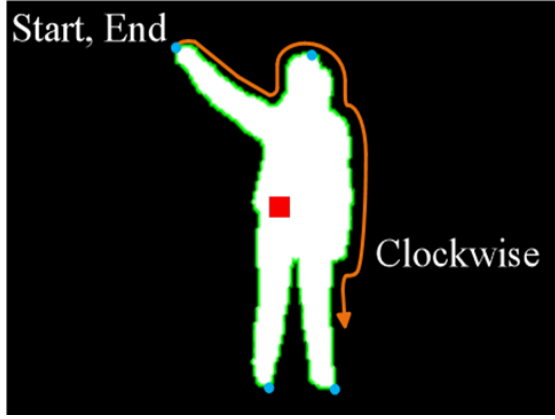


Figure 6. The calculation sequence of the distance values  $d_i$

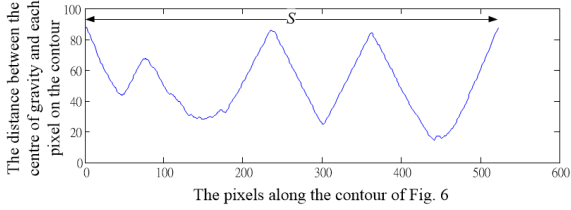


Figure 7. The sequence curve of the distance values  $d_i$

Let the curve in Figure 7 be filtered through a low-pass filter to remove the noise, thus obtaining the smoother curve denoted by  $\hat{d}_i$  as shown in Figure 8, where the green points denote tip points such as the human head, hands, and feet, as the feature points of the silhouette. It can be seen that the green points in Figure 8 are the blue points of the contour in Figure 6.

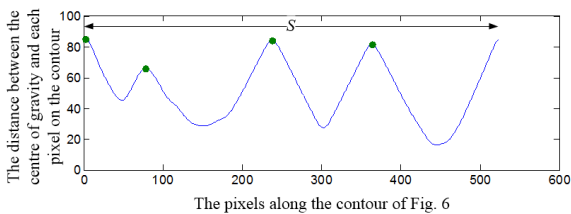


Figure 8. The peak points on the distance value curve

After obtaining the feature points, the next task is to acquire the feature vectors. First we let  $(x_{pi}, y_{pi})$  denote the location of the feature point  $p_i$ , and connect each feature point to the centre of gravity. Then, a feature skeleton structure is obtained as shown in Figure 9. However, the centre of gravity may be not within the human's silhouette (see Figure 9(b)), such that the feature skeleton may be incorrect. In this situation, the centre of gravity must be repositioned to within the silhouette in order to obtain a true feature skeleton. A vertical line and a horizontal line crossing on

the centre of gravity are therefore plotted and the line with two crossing points exactly on the human's silhouette is chosen. Finally, the centre of gravity is shifted to the centre position of the two crossing points. The new gravity point is shown in Figure 10.

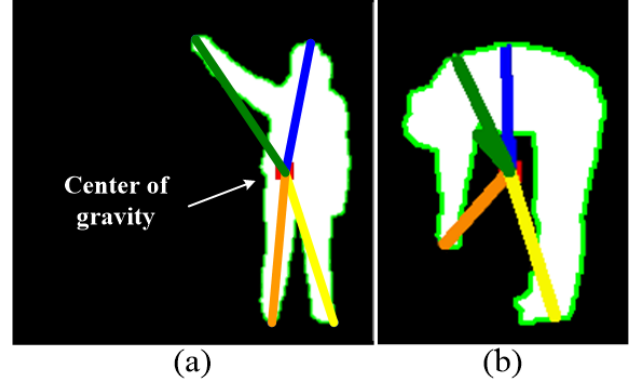


Figure 9. Feature skeleton when the centres of gravity are (a) inside the human silhouette and (b) outside the human silhouette

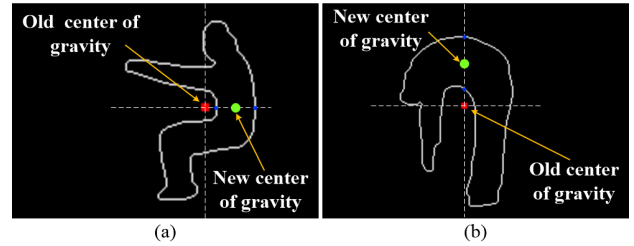


Figure 10. New centre of gravity on (a) a sitting posture with raised arms and (b) a stooping posture

Let each branch of the feature skeleton shown in Figure 9 be one of the human feature vectors  $V_i = [\hat{x}(p_i), \hat{y}(p_i)]$  which denotes the vector from the feature point  $p_i$  to the centre of gravity, as in equation (4).

$$\begin{cases} \hat{x}(p_i) = x_{pi} - x_c \\ \hat{y}(p_i) = y_{pi} - y_c \end{cases} \quad (4)$$

Then, the Cartesian coordinate  $[\hat{x}(p_i), \hat{y}(p_i)]$  is transformed into a polar coordinate  $[L_i, \theta_i]$  as follows:

$$L_i = \sqrt{(\hat{x}(p_i))^2 + (\hat{y}(p_i))^2}$$

and

$$\theta_i = \tan^{-1} \left( \frac{\hat{y}(p_i)}{\hat{x}(p_i)} \right) = \begin{cases} \theta_i, & \text{if } \theta_i \text{ is positive} \\ \theta_i + 360, & \text{if } \theta_i \text{ is negative} \end{cases} \quad (5)$$

where  $i=1, 2, \dots, m$ , and  $m$  is the total number of branches of the feature skeleton. In general the maximum number of branches is five. The transformation is illustrated in Figure 11.



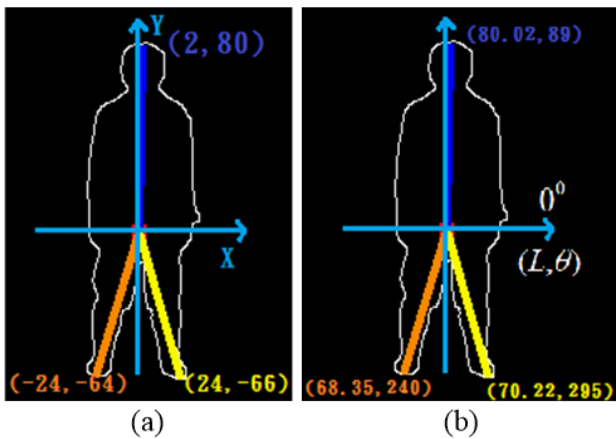


Figure 11. Feature vectors with (a) Cartesian coordinate and (b) polar coordinate

### 3. LVQ neural network and a final identification

After feature extraction, the LVQ neural network is applied to classify the postures using the extracted feature vectors. The LVQ shown in Figure 12 is a supervised neural network which is often used for pattern classification (see [32-34]). In this study, we decided to use an LVQ as the classifier of human posture recognition because of its simple structure, fast operation, and strong fault tolerance [34]. However, two things should be noted regarding the training of the LVQ neural network. First, the input arrangement of the LVQ neural network should be ordered; second, the feature vectors  $V_i$  should be normalized as  $\bar{V}_i$ . The following is a detailed explanation of these two points.

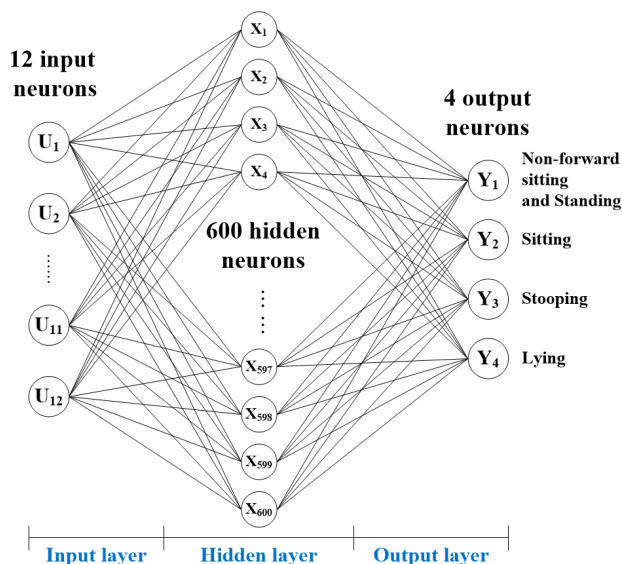


Figure 12. The structure of the LVQ neural network

#### 3.1 Inputs arrangement in LVQ neural network

Let the LVQ neural network have 12 inputs which contain 10 feature vectors and the two depth values  $D_1$  and  $D_2$  ( $D_1$

and  $D_2$  will be defined later). According to experimental experience, the order of the feature vectors which are inputted into the network will affect the recognition rate. Therefore, two order arrangements of feature vectors are proposed, as shown in Table 1, in which  $U_i$  is the  $i$ -th input neuron of the LVQ neural network, and  $j$  is the length and  $\Theta_j$  is the angle of the  $j$ -th feature vector, respectively.

$U_1$	$U_2$	$U_3$	$U_4$	$U_5$	$U_6$
1	$\Theta_1$	2	$\Theta_2$	$D_1$	3
$U_7$	$U_8$	$U_9$	$U_{10}$	$U_{11}$	$U_{12}$
$\Theta_3$	$D_2$	4	$\Theta_4$	5	$\Theta_5$

Table 1. The Order Arrangement for Input Neurons

In order to arrange the inputs of the LVQ neural network, let two disks be divided into six regions with different degree ranges as shown in Figure 13. If any one feature vector is located in Region 1, which is the sector between  $45^\circ$  and  $135^\circ$  (see the left side of Figure 13), then order arrangement I is followed. Otherwise, order arrangement II is followed (see the right side of Figure 13). The two arrangements are presented below.

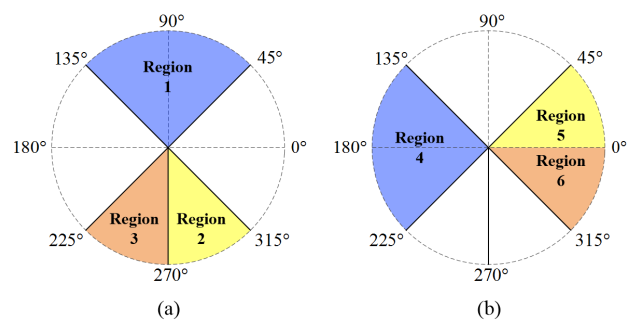


Figure 13. The region division for the inputs of the LVQ neural network. (a) Order arrangement I. (b) Order arrangement II.

(I) Order arrangement I (used if any one feature vector is located in Region 1, which is a sector between  $45^\circ$  and  $135^\circ$  as shown in Figure 13(a)).

Stage 1. There is a feature vector in Region 1 whose angle is much closer to  $90^\circ$  than all other feature vectors. This feature vector is called  $V_1$ . Then,  ${}_1=L_1$  and  $\Theta_1=\theta_1$  are assigned.

Stage 2. If there exist feature vectors in Region 2, the vector angle which is closest to, but does not exceed,  $270^\circ$ , will be called  $V_2$ . Then,  ${}_2=L_2$  and  $\Theta_2=\theta_2$  are assigned. If  $V_2$  does not exist, then  ${}_2=0$  and  $\Theta_2=0^\circ$  are assigned.

Stage 3. If there is a feature vector in Region 3 whose angle is much closer to  $270^\circ$  than all other feature vectors, it will be called  $V_3$ . Then,  ${}_3=L_3$  and  $\Theta_3=\theta_3$  are assigned. If  $V_3$  does not exist, then  ${}_3=0$  and  $\Theta_3=0^\circ$  are assigned.

Stage 4. If there is a feature vector which does not satisfy the above three conditions, but whose angle is the closest to, and anticlockwise of,  $0^\circ$ , this feature vector is called  $V_4$ . Then,  ${}_4=L_4$  and  $\Theta_4=\theta_4$  are assigned. Then go to stage 5. If  $V_4$  does not exist, this means that there are no remaining vectors, so  ${}_4=0$ ,  $\Theta_4=0$ ,  ${}_5=0$  and  $\Theta_5=0$  are assigned. Then move on to stage 6.

Stage 5. If one final vector feature remains, it will be called  $V_5$ . Then,  ${}_5=L_5$  and  $\Theta_5=\theta_5$  are assigned. If  $\bar{V}_5$  does not exist,  ${}_5=0$  and  $\Theta_5=0$  are assigned.

Stage 6. The remaining inputs are  $D_1$  and  $D_2$ , where  $D_1=\frac{D_R-D_C}{100}$  and  $D_2=\frac{D_L-D_C}{100}$ .  $D_R$  is the depth value of the terminal point of the feature vector  $V_2$ ,  $D_L$  is the depth value of the terminal point of the feature vector  $V_3$  and  $D_C$  is the depth value of the centre of gravity (see Figure 14).

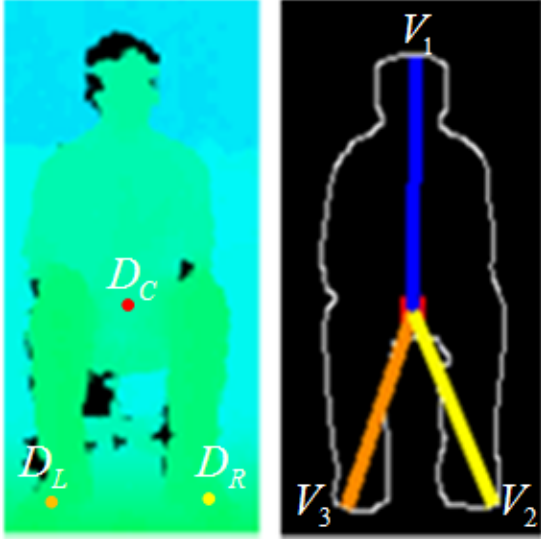


Figure 14. The depth values

Figure 15 shows the order of the feature vectors when using order arrangement I for the standing and lateral sitting postures. It is seen that there must be at least one feature vector in Region 1 when the order arrangement I is used; in other words, the position of the human's head is in Region 1. Therefore, by using order arrangement I for the inputs of the LVQ network, the standing and sitting postures can be recognized. Furthermore,  $D_1$  and  $D_2$  are used to establish recognition of the forward-facing sitting posture in which the centre of gravity,  $V_2$  and  $V_3$  have different depth values.

(II) Order arrangement II (used if there is no feature vector located in Region 1).

Stage 1. If there is a feature vector in Region 4 whose angle is much closer to  $180^\circ$  than all the other feature vectors, this feature vector is called  $V_1$ . Then,  ${}_1=L_1$  and  $\Theta_1=\theta_1$  are assigned. Otherwise,  ${}_1=0$  and  $\Theta_1=0$  are assigned.

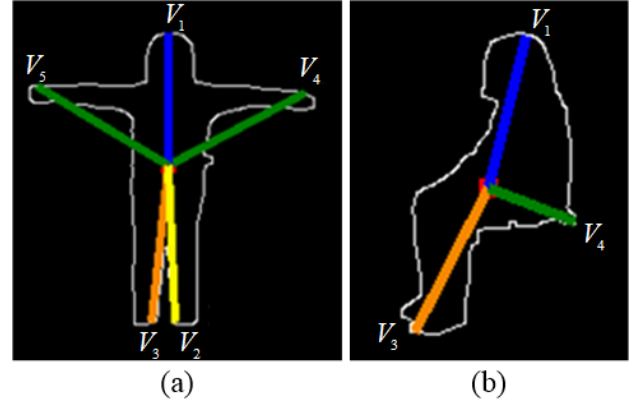


Figure 15. The feature vector arrangement according to order arrangement I. (a) Standing. (b) Sitting.

Stage 2. If there exist feature vectors in Region 5, the vector angle closest to  $0^\circ$  will be called  $V_2$ . Then,  ${}_2=L_2$  and  $\Theta_2=\theta_2$  are assigned. If there is no feature in this region, then  ${}_2=0$  and  $\Theta_2=0$  are assigned.

Stage 3. If there exists a feature vector in Region 6 whose angle is closer to  $0^\circ$  than all other feature vectors, this feature vector will be called  $V_3$ . Then,  ${}_3=L_3$  and  $\Theta_3=\theta_3$  are assigned. If there is no feature in this region, then  ${}_3=0$  and  $\Theta_3=0$  are assigned.

Stage 4. If there is a feature vector which does not satisfy the above three conditions, but whose angle is the closest to, and anticlockwise of,  $0^\circ$ , this feature vector is called  $V_4$ . Then,  ${}_4=L_4$  and  $\Theta_4=\theta_4$  are assigned. Then go to the next stage. If  $V_4$  does not exist, then there are no remaining vectors; then,  ${}_4=0$ ,  $\Theta_4=0$ ,  ${}_5=0$  and  $\Theta_5=0$  are assigned. Go to stage 6.

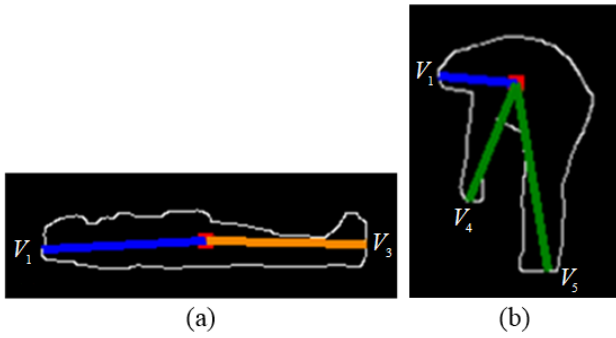
Stage 5. If there is a last remaining vector feature called  $V_5$ , then  ${}_5=L_5$  and  $\Theta_5=\theta_5$  are assigned. If  $V_5$  does not exist, then  ${}_5=0$  and  $\Theta_5=0$  are assigned.

Stage 6. Lastly, the remaining inputs  $D_1=0$  and  $D_2=0$  are set.

Figure 16 shows the lying and stooping postures with their feature vectors arranged according to order arrangement II. The order arrangement II is used for recognizing those postures in which there must be feature vectors in Region 4, Region 5, or Region 6; in other words, the human's head is considered to be in Region 4, Region 5, or Region 6. Therefore, using order arrangement II for the inputs of the LVQ neural network, the lying and stooping postures can be recognized. Furthermore,  $D_1$  and  $D_2$  are not used in order arrangement II.

### 3.2 Feature vectors normalization

Having assigned the order of the inputs of the LVQ neural network, it should be noted that if the human is far away from (or near to) the Kinect sensor, then the perceived size of the human will be smaller (or larger). This may affect the



**Figure 16.** The feature vector arrangement according to order arrangement II. (a) Lying. (b) Stooping.

accuracy of the recognition; therefore, the values in Table 1 should be normalized in advance. Let  $\bar{L}_i = l_i / L_{\max}$  and  $\theta_i = \Theta_i / 360$ , where  $L_{\max} = \max_i(L_i)$ ,  $i = 1, \dots, m$ . Then, the normalized feature vector is denoted by  $\bar{V}_i$ . Therefore, Table 1 should be replaced by Table 2 as follows. Another advantage of normalizing the feature vectors is that no matter what the height of the human subject, the feature vectors are only ratio values, so that the proposed posture recognition algorithm can apply to different statures.

$U_1$	$U_2$	$U_3$	$U_4$	$U_5$	$U_6$
$r_1$	$\theta_1$	$\bar{L}_2$	$\theta_2$	$D_1$	$\bar{L}_3$
$U_7$	$U_8$	$U_9$	$U_{10}$	$U_{11}$	$U_{12}$
$\theta_3$	$D_2$	$\bar{L}_4$	$\theta_4$	$\bar{L}_5$	$\theta_5$

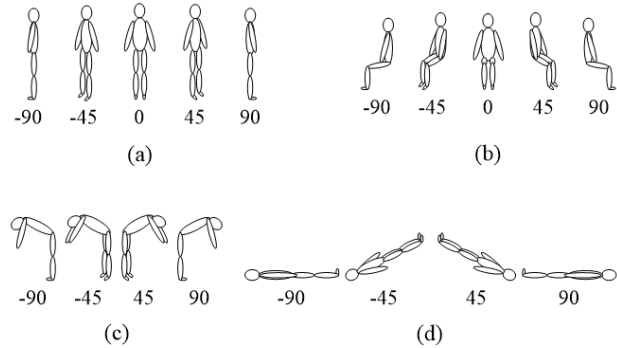
**Table 2.** The Normalized Order Arrangement for Input Neurons

### 3.3 The operation of the LVQ network

The previous section has shown how the lateral kneeling posture is recognized by the upper and lower body ratio of a human. However, there are still many different postures to be recognized. Using an LVQ neural network is the next recognition process presented here. The used LVQ has 12 input neurons, 600 hidden neurons and four output neurons. The 12 input neurons contain five lengths and five angles of feature vectors and two depth values. There are 1105 sets of training data with which to train the LVQ neural network. Since the hidden layer needs enough hidden neurons to memorize the training data, the number of hidden neurons is 600. The four output neurons represent the four classes of posture, sitting or standing, stooping, and lying, respectively. It is noted that one of those outputs may denote non-forward sitting or standing; therefore, an extra check is needed to determine whether the posture is sitting or standing.

The training data contain 292 standing postures, 320 sitting postures, 240 stooping postures and 253 lying postures. All training data contain those postures shown in Figure 17. Figure 17(a) shows five standing postures with five orientations, respectively. Figures 17(b), (c), and (d) show

the different postures with the different respective orientations. After training, the weights between the inputs and the hidden neurons will be obtained.



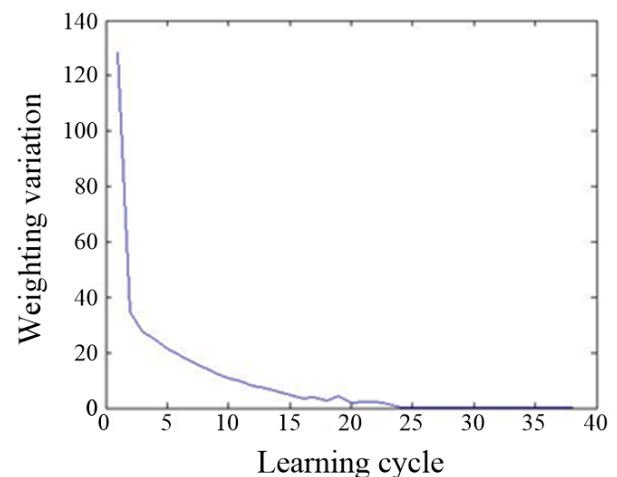
**Figure 17.** The postures of the training data. (a) Standing. (b) Sitting. (c) Stooping. (d) Lying.

The output weights  $W$  are set as equation (6).

$$W = \begin{bmatrix} 1 & 1 & \dots & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & \dots & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & \dots & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \quad (6)$$

where  $W$  is a  $4 \times 600$  matrix. For instance, the element at position (1, 150) in  $W$  is 1 at (6), which means the weight at the link between hidden neuron  $X_{150}$  and output  $Y_1$  is equal to 1.

The training of the LVQ neural network is stopped when the input weight variation is less than 0.05, as shown in Figure 18. The training terminates at the 38<sup>th</sup> cycle.



**Figure 18.** The training curve with input weight variation and training cycle

Since the LVQ is a supervised neural network, training and testing should follow order arrangement I or II when setting the inputs. The reason for using two order arrangements is that, based on the results of many experiments, a much higher recognition rate is achieved when using two



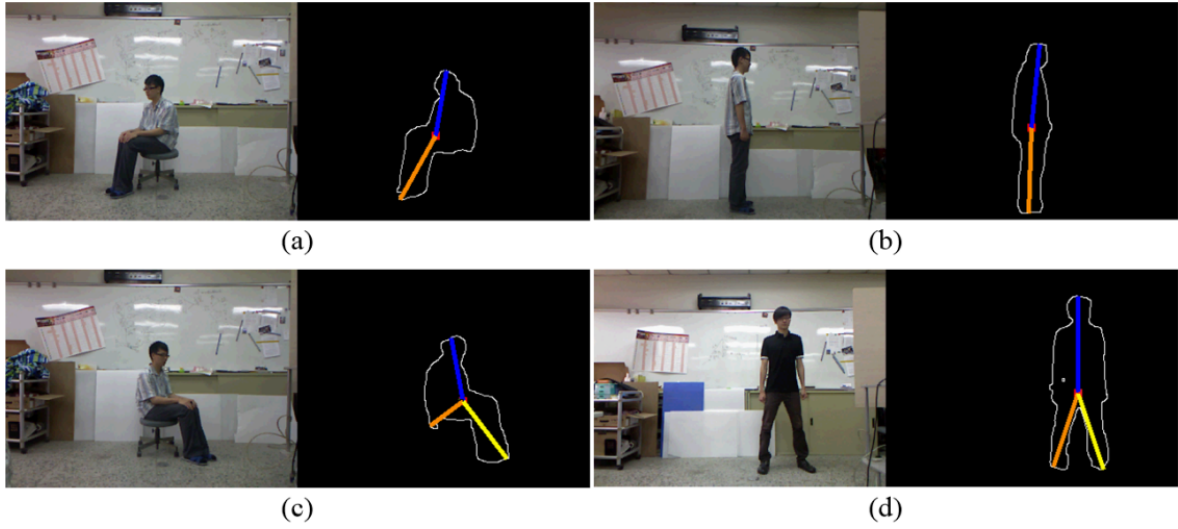


Figure 19. Human feature vectors of the non-forward-facing sitting and the standing postures

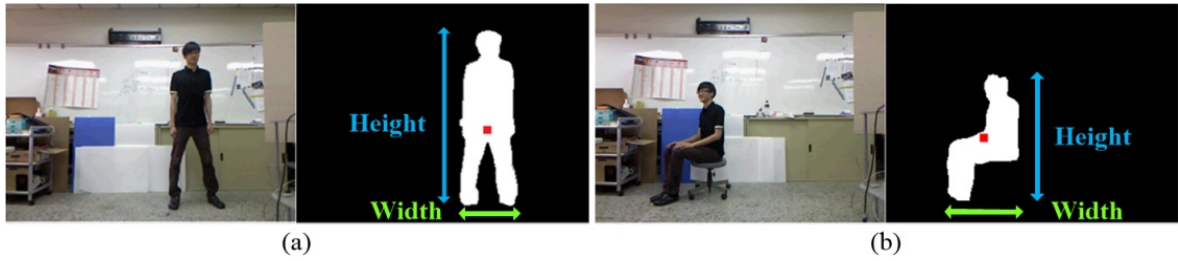


Figure 20. Ratio of the width and height of the human silhouette contour

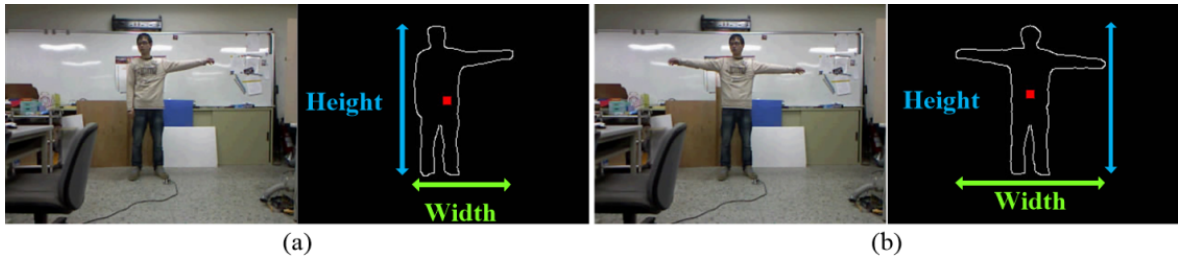


Figure 21. The hands at different positions

different arrangements for two different cases than when only using one arrangement.

### 3.4 One more check

After obtaining the outputs of LVQ, one more step is necessary for posture recognition. It can be seen that the feature vector structure in Figure 19(a) is similar to that in Figure 19(b), and the vector structure in Figure 19(c) is similar to that in Figure 19(d), respectively. This causes confusion of posture recognition in the LVQ neural network. Therefore, an extra step is needed to determine the posture when the subject is standing or sitting but not facing the sensor. The two ratios are defined as follows. One is  $R_s = \frac{\text{height of sitting}}{\text{width of sitting}}$  and the other is  $R_d = \frac{\text{height of standing}}{\text{width of standing}}$ , where the numerator and denominator in  $R_d$  or  $R_s$  are the width and height of a human, respectively. It is noted that  $R_d$  is

much larger than  $R_s$  as shown in Figure 20. Here,  $R_s$  is about 1.92 and  $R_d$  is about 3.83. But what is the best way to measure the exact height and width of a human? The simplest method would be to use the distance between the highest (or left-most) and lowest (or right-most) points of the silhouette for the height (or the width). However, this method can easily lead to inaccuracies; for instance, if the subject were to raise their arms above their head as shown in Figure 21. Alternatively, if the edge points of the silhouette to be selected were noise points, then the value of height (or width) could be incorrect. Therefore, the following method of measuring the correct height and width of a human silhouette is necessary.

Here, the horizontal and vertical projection histograms are considered to obtain the height and width of the human silhouette. The human's standing posture is shown on the

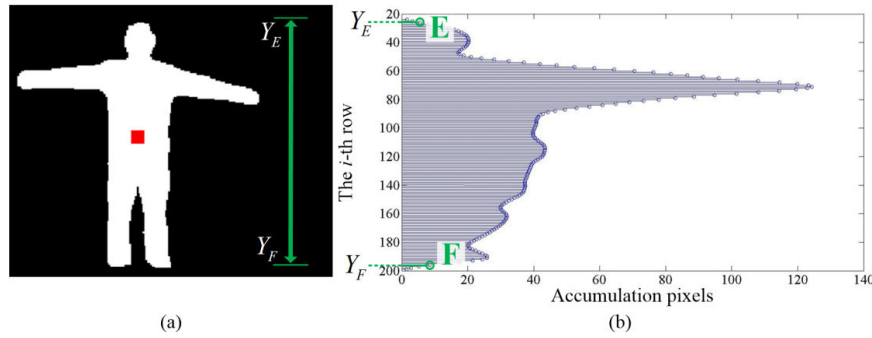


Figure 22. Horizontal projection of the standing posture

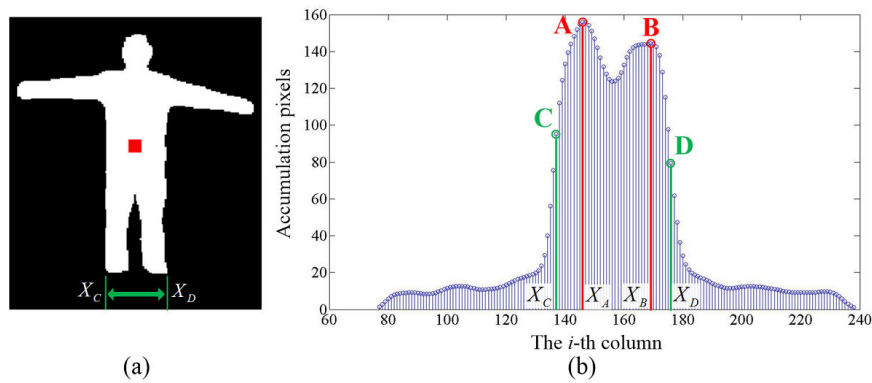


Figure 23. Vertical projection of the standing posture

left side of Figure 22, and the horizontal projection histogram of this posture is shown on the right side of Figure 22. In order to exclude the possible noise above the head and below the sole of the foot, a row on the horizontal projection histogram is selected in which the number of accumulated pixels is greater than five; then, the number of all such rows will be considered to correspond to the height of the person. In other words, rows with fewer than five accumulated pixels are ignored. In Figure 22, the length of the interval  $[y_E, y_F]$  is the height of the person, where  $y_E$  is the y-axis value of the point E and  $y_F$  is the y-axis value of the point F. On the other hand, the vertical projection histogram is used to measure the width of the person. However, some errors may occur due to noise or the subject raising their arm or arms (see Figure 23). Therefore, from each column of vertical projection, the two points (such as points C and D in Figure 23) which have the highest positive slope and negative slope are chosen, respectively. The length of the interval  $[x_C, x_D]$  will be the width of the human silhouette without arms where  $x_C$  is the x-axis value of the point C and  $x_D$  is the x-axis value of the point D. Finally, the ratio of the width and height  $R$  is calculated as follows:

$$R = \frac{y_E y_F}{x_C x_D} \quad (7)$$

when  $R \geq 3.1$ , the human posture is standing; when  $R < 3.1$ , the human posture is sitting.

There are two threshold values in this paper. In order to find suitable threshold values for  $K_p$  in (2) and  $R$  in (7), a lot of measurements are needed. In this study, more than 10 people were each measured more than 10 times with different distances. We then selected the threshold values for  $K_p$  and  $R$  from the average of those measurements.

All techniques for posture recognition have now been presented. Now let us summarize the above recognition techniques in the following procedure. A flow chart is also shown in Figure 24.

### 3.5 Procedure of the recognition process

**Step 1:** Image capture and image pre-processing.

**Step 2:** Feature extraction. The features include the upper and lower body ratio of the human silhouette  $K_p$ , and the feature vectors. If  $K_p \geq 1.4$ , then the human's posture is judged to be the kneeling posture.

**Step 3:** The operation of LVQ neural network. The four postures, sitting facing forward, stooping, lying, and finally, non-forward-facing standing or sitting can be identified using the outputs of the LVQ neural network.

**Step 4:** If one of the outputs of the LVQ network is standing or sitting, one more check is required, as follows. If  $R \geq 3.1$ ,

the human's posture is judged to be "standing". Otherwise the posture is judged to be "sitting".

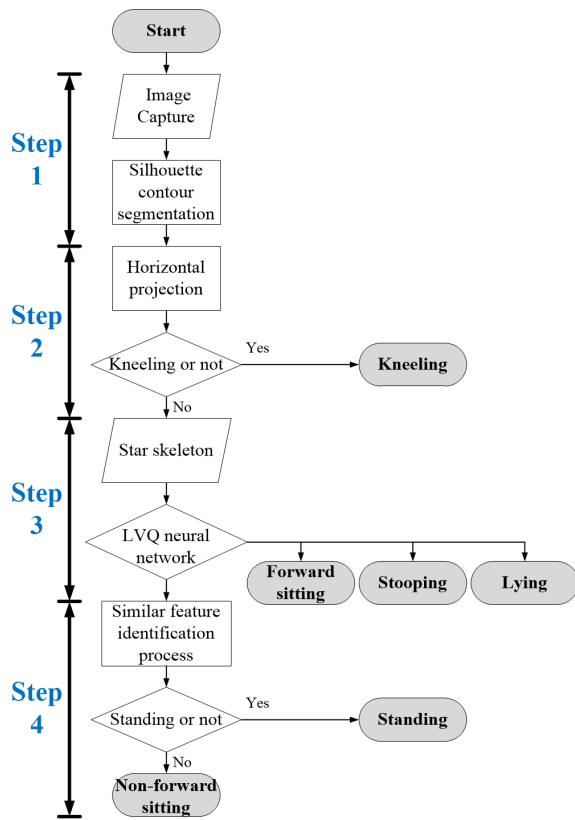


Figure 24. Flow chart of the posture recognition process

#### 4. Experiment results and discussion

In all the experiments performed for this study, the Kinect and the PC are on the same table at a height of 70 cm, and the human subject is in front of the table at a distance of 2.5 to 4.5 m. By using a Kinect sensor, the proposed method is implemented to recognize five human postures: kneeling, standing, sitting, stooping and lying. In the experiments, each posture can be oriented in five different directions:  $-90^\circ$ ,  $-45^\circ$ ,  $0^\circ$ ,  $45^\circ$ , and  $90^\circ$ , as shown in Figure 25. The standing posture shown in Figure 25(a) is tested 80 times and each orientation is tested 16 times. The sitting posture shown in Figure 25(b) is tested 100 times and each orientation is tested 20 times. The other three postures are tested 80 times each, and each orientation is tested 20 times. The posture recognition algorithm is implemented in C++ language. Eight students were taken as the subjects of the experiments. It can be seen from Table 3 that the successful recognition rate for each experiment subject is over 98.15%. The successful recognition rate for each posture is over 97.25%, and the total average successful recognition rate is 99.0125%. This is very high.

Although the proposed posture recognition process contains several steps, in the experiments it takes less than three milliseconds to recognize a posture. Figure 26 shows

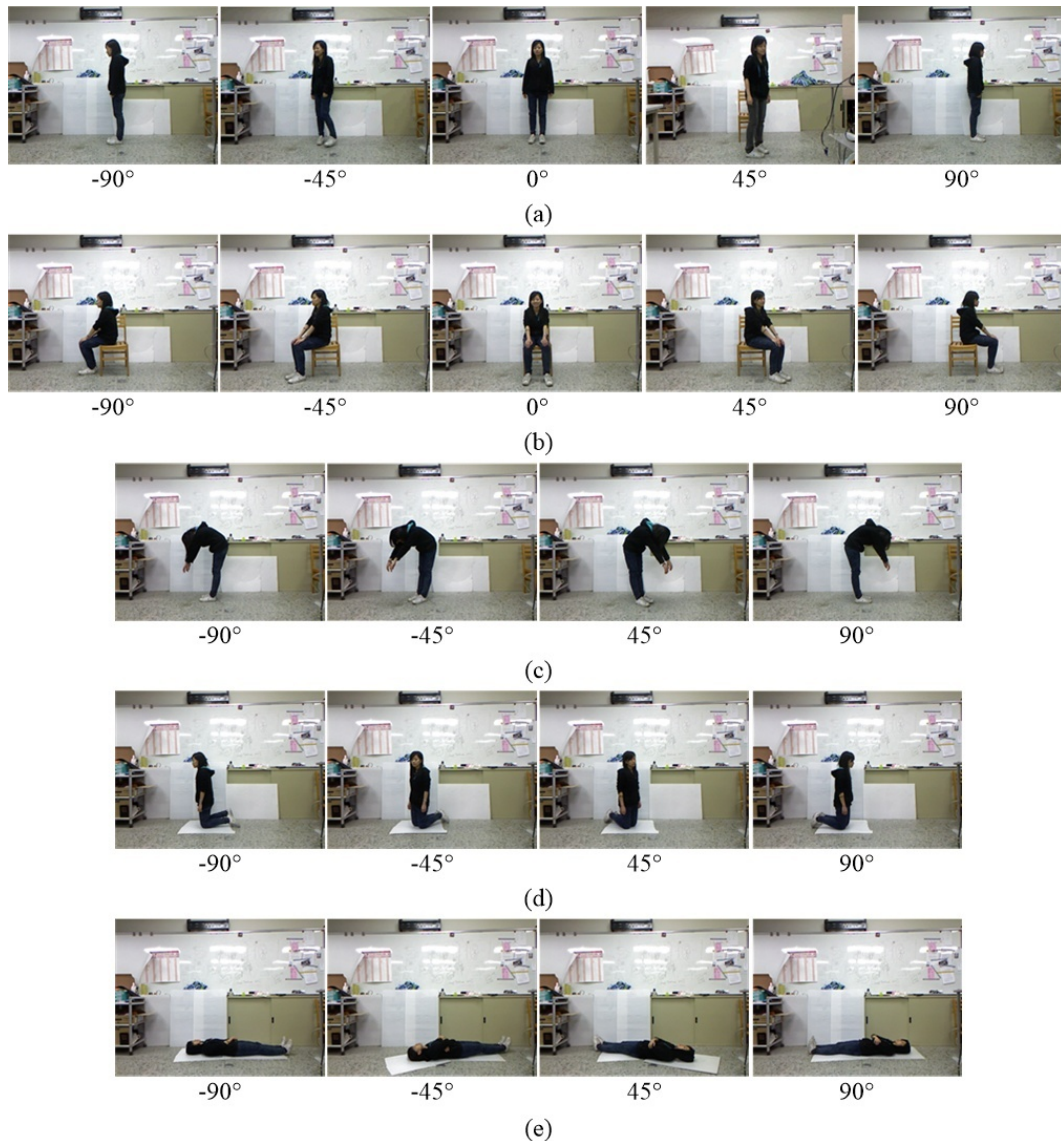
all of the human postures that were successfully recognized in different environments and with different distances between the Kinect sensor and the subject, as shown in the first column of the figure. The fourth indoor environment is with low luminance. In each case the worst recognition rate is still over 97%. Figure 27 shows a breakdown of the computation time required to recognize each of the 10 postures, in which  $T_L$  indicates the total computation time for recognizing a certain posture. The remaining notations  $T_i$ ,  $i=1, 2, \dots, 6$ , indicate the computation time used by each step of the posture recognition process and are defined as follows:  $T_1$  is the image processing to remove noise;  $T_2$  is the silhouette contour segmentation from the captured images;  $T_3$  is the horizontal projection and keeling posture judgment;  $T_4$  is the extraction of feature vectors;  $T_5$  is the LVQ neural network recognition of the forward-facing sitting, stooping and lying postures; and  $T_6$  is the identification of the standing or non-forward-facing sitting postures. If  $T_i=0$ , then the  $i$ -process is not needed. For instance, the processes  $T_4$ ,  $T_5$  and  $T_6$  are not necessary to recognize posture (iii) in Figure 26. It is seen that  $T_2$  is larger than all the other values of  $T_i$ ,  $i=1, 3, 4, 5, 6$ , for recognition of all postures, since the segmentation process includes connected component implementation, which takes more time due to the large subject silhouettes. It is seen all the 10 postures were successfully recognized in less than three milliseconds. It can therefore be concluded that the algorithm can be applied to a real-time posture recognition application.

	Standing	Sitting	Stooping	Kneeling	Lying	Recognition success rate for each subject
A	79/80	99/100	80/80	80/80	79/80	99.3%
B	80/80	98/100	76/80	80/80	80/80	98.6%
C	80/80	98/100	80/80	79/80	79/80	98.85%
D	79/80	94/100	80/80	80/80	80/80	98.55%
E	80/80	100/100	80/80	80/80	79/80	99.75%
F	79/80	100/100	80/80	80/80	80/80	99.75%
G	79/80	97/100	80/80	80/80	80/80	99.15%
H	80/80	92/100	79/80	80/80	80/80	98.15%
Recognition success rate for each posture (%)	99.38	97.25	99.22	99.69	99.53	99.0125

Table 3. The Posture Recognition Rates

We compare the performance of the proposed method to that of three alternative methods proposed in [25, 26] and [27], respectively. In [25] there are 10 postures to be recognized: walking, standing up, sitting, picking up, carrying, throwing, pushing, pulling, waving hands and clapping hands. Of these 10 postures, three, standing up, sitting, and picking up, showed performance similar to that shown for the postures of standing, sitting and stooping,





**Figure 25.** The recognized postures with different orientations. (a) Standing. (b) Sitting. (c) Stooping. (d) Kneeling. (e) Lying.

respectively, in our paper. Table 4 shows that our recognition rate for these three postures is much higher. The study presented in [26] used four feature extraction methods to recognize the four postures of standing, sitting, lying, and bending. We choose those two of the four methods that have the best recognition accuracy in order to make a comparison with ours. The chosen two methods used seven joint-angles with scaling and nine joint-angles with scaling to extract features, respectively. The comparison is included in Table 4. It is seen that when the test subjects are not facing the Kinect sensor, the recognition rates of some postures are very low (see Table 3 in [26]). The study presented in [27] used four classifiers – back-propagation neural network, support vector machine, decision tree and naïve Bayes – to recognize human postures, respectively. The back-propagation neural network had the highest success rates. However, the authors only recognized three postures and did not provide the recognition results of test subjects facing in different directions. The proposed

method not only recognizes five postures (including the three postures recognized in [27]), it also deals with test subjects facing in different directions.

	Standing	Sitting	Stooping	Kneeling	Lying
Our method	99.38	97.25	99.22	99.69	99.53
Paper [25]	93.5	91.5	97.5	-	-
Seven joint-angles in [26]	81.29	100	44.59	-	89.775
Nine joint-angles in [26]	76.18	97.39	70.42	-	48.25
Back propagation neural network in [27]	100	100	-	-	100

**Table 4.** A Comparison of the Average Success Rates of Posture Recognition

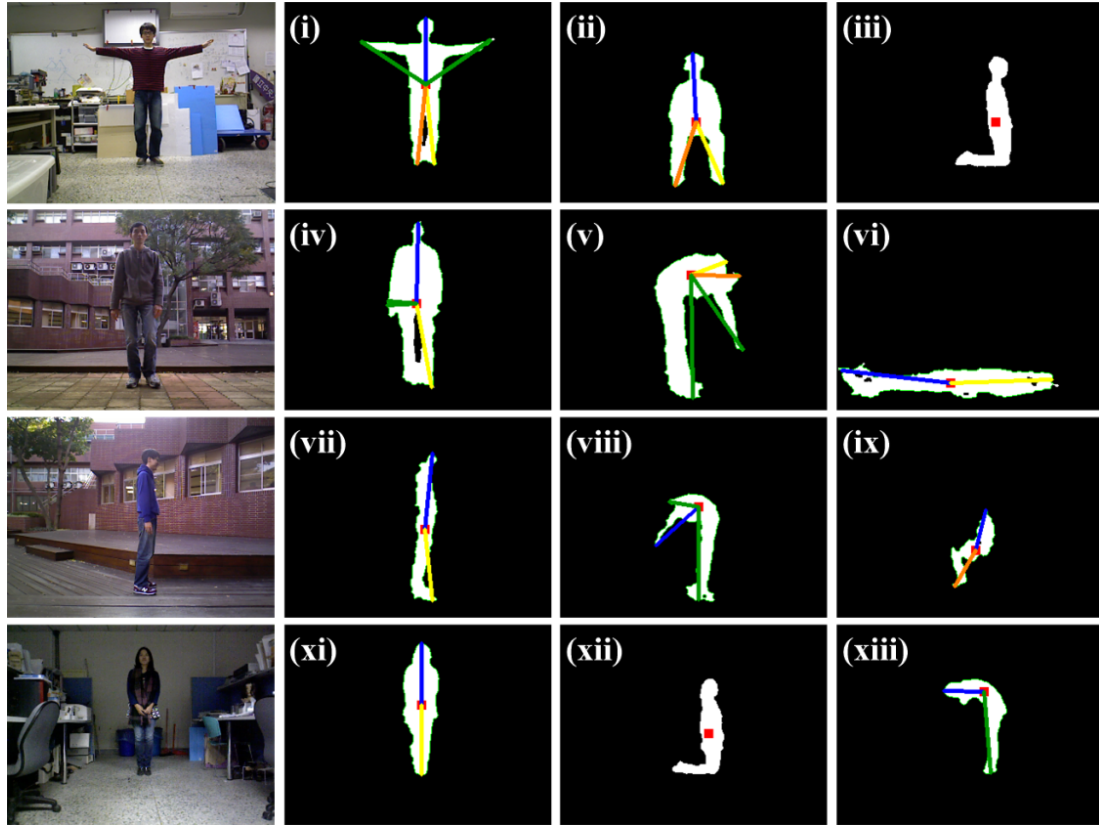


Figure 26. Illustration of the different environments and tested postures

Note that the recognition methods in [25-27] were based on skeletal joint positions of the subject drawn from the Kinect software development kit (SDK). If the subject's head is not at the top of the silhouette, for example in the postures of stooping or lying (see Figure 28), these methods may therefore give incorrect recognition. Even so, we find that the average recognition time for each image when using the proposed method is less than three milliseconds; in other words, the lowest recognition frame rate is 333.33 frames/per second (see Figure 27). It is clear that the proposed method achieves recognition very fast and with very high efficiency.

The proposed method does, however, have limitations. For instance, when the subject is kneeling and facing the Kinect sensor, the lower legs are hidden, as shown in Figure 29(a), which may cause the recognition process to fail. In this situation, the horizontal projection cannot display the features of the kneeling posture clearly where the maximum upper-body width value  $N_U$  is 67 and the maximum lower-body width value  $N_L$  is 67, as shown in Figure 29(b). Then, the feature value  $K_p$  is  $1 < 1.4$ . According to our experiments, the recognition of the postures of kneeling, stooping and lying may fail if the subject's orientation is not restricted the  $[-90^\circ, -45^\circ]$  or  $[45^\circ, 90^\circ]$  ranges.

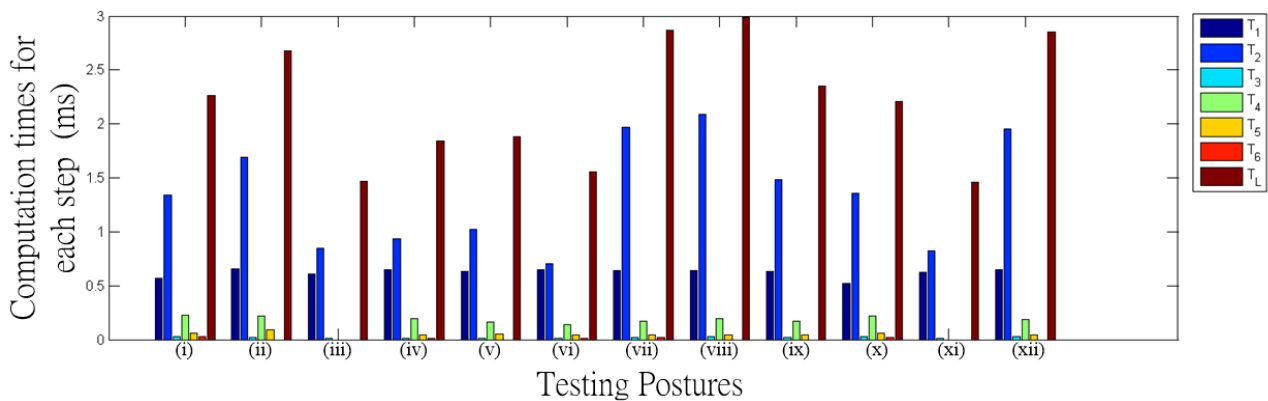
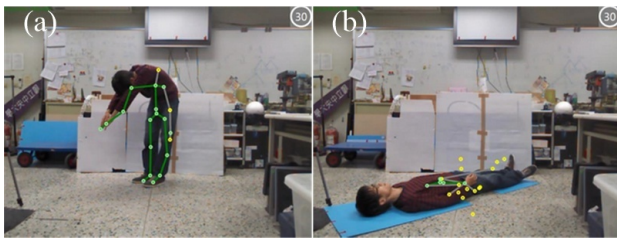
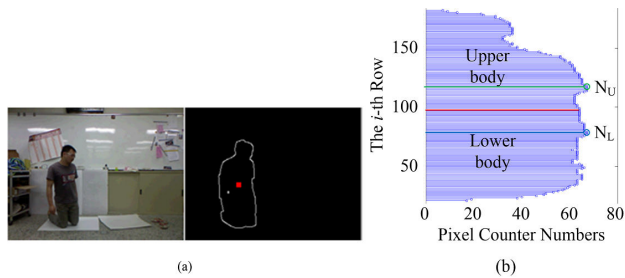


Figure 27. Breakdown of the total posture recognition computation time





**Figure 28.** Skeletal maps as created by the Kinect sensor for Windows SDK 1.8 software [35]



**Figure 29.** (a) Legs hidden in the kneeling posture. (b) horizontal projection of the posture in Figure 29(a).

## 5. Conclusion

This paper has proposed an effective procedure to recognize the five human postures of standing, sitting, stooping, kneeling and lying, even when the human subjects have different statures or orientations. In the experiments, it is found that the average success rate of the proposed posture recognition method is higher than 99%. The Kinect sensor which provides the depth information can avoid the influence of illumination and shadow in image processing. By extracting many features and using the LVQ neural network, an efficient posture recognition procedure is produced. The proposed posture recognition method has three advantages: firstly, it has a very high recognition rate; secondly, it requires fewer training data sets; and finally, it uses a more economical sensor compared to other methods. However, it must be admitted that, where part of the subject's body is hidden, some further study is required, as such situations may cause a recognition failure due to incorrect feature extraction. This issue can be considered as a subject for future studies.

It is believed that a more reliable method of posture recognition could recognize more complex postures. In our future research, we hope to develop a more reliable posture recognition technique that could prove extremely useful as part of a homecare system for monitoring elderly people who live alone. The system will be able to detect abnormal postures produced by a fall or a medical emergency, and then immediately alert the emergency services.

## 6. Acknowledgements

The authors would like to thank the Ministry of Science and Technology of Taiwan for its support under Contracts NSC102-2221-E-008-085-MY3.

## 7. References

- [1] Mattmann C, Amft O, Harms H, Troster G, Clemens F. Recognizing Upper Body Postures Using Textile Strain Sensors. In: 2007 11th IEEE International Symposium on Wearable Computers; 11-13 October; Boston, MA. 2007. pp. 29-36. DOI: 10.1109/ISWC.2007.4373773.
- [2] Harms H, Amft O, Roggen D, Troster G. Rapid Prototyping of Smart Garments for Activity-Aware Applications. *Journal of Ambient Intelligence and Smart Environments*. 2009; 1(2): 87-101. DOI: 10.3233/AIS-2009-0015.
- [3] Harms H, Amft O, Troster G. Estimating Posture-Recognition Performance in Sensing Garments Using Geometric Wrinkle Modeling. *IEEE Transactions on Information Technology in Biomedicine*. 2010; 14(6): 1436-1445. DOI: 10.1109/TITB.2010.2076822.
- [4] Karantonis DM, Narayanan MR, Mathie M, Lovell NH, Celler BG. Implementation of a Real-time Human Movement Classifier Using a Triaxial Accelerometer for Ambulatory Monitoring. *IEEE Transactions on Information Technology in Biomedicine*. 2006; 10(1): 156-167. DOI: 10.1109/TITB.2005.856864.
- [5] Jeong D-U, Kim S-J, Chung WY. Classification of Posture and Movement Using a 3-axis Accelerometer. In: *International Conference on Convergence Information Technology*. November 21-23; Gyeongju, Korea. 2007. pp. 837-844. DOI: 10.1109/ICCIT.2007.202.
- [6] Ukida H, Kaji S, Tanimoto Y, Yamamoto H. Human Motion Capture System using Color Markers and Silhouette. In: *Proceedings of the IEEE Instrumentation and Measurement Technology Conference*; 24-27 April; Sorrento, Italy. 2006. pp. 151-156. DOI: 10.1109/IMTC.2006.328334.
- [7] Chiu CY, Wu CC, Wu YC, Wu MY, Chao SP, Yang SN. Retrieval and Constraint-based Human Posture Reconstruction from a Single Image. *Journal of Visual Communication and Image Representation*. 2006; 17(4): 892-915. DOI: 10.1016/j.jvcir.2005.01.002.
- [8] Liu HY, Wang WJ, Wang RJ, Tung CW, Wang PJ, Chang IP. Image Recognition and Force Measurement Application in the Humanoid Robot Imitation. *IEEE Transactions on Instrumentation and Measurement*. 2012; 61(1): 149-161. DOI: 10.1109/TIM.2011.2161025.
- [9] Li C-C, Chen Y-Y. Human Posture Recognition by Simple Rules. In: *IEEE International Conference on Systems, Man and Cybernetics*; 8-11 October; Taipei, Taiwan. 2006. p. 3237 - 3240. DOI: 10.1109/ICSMC.2006.384616.
- [10] Boulay B, Bremond F, Thonnat M. Posture Recognition with a 3d Human Model. In: *The IEE Inter-*

- national Symposium on Imaging for Crime Detection and Prevention; 7-8 June; 2005. pp. 135-138. DOI: 10.1049/ic:20050085.
- [11] Boulay B, Bremond F, Thonnat M. Applying 3d Human Model in a Posture Recognition System. *Pattern Recognition Letters*. 2006; 27(15): 1788-1796. DOI: 10.1016/j.patrec.2006.02.008.
- [12] Castiello C, D'Orazio T, Fanelli AM, Spagnolo P, Torsello MA. Model-free Approach for Posture Classification. In: *IEEE Conference on Advanced Video and Signal Based Surveillance*; 15-16 September; Cerno, Italy. 2005. pp. 276-281. DOI: 10.1109/AVSS.2005.1577280.
- [13] Xie F, Xu G, Cheng Y, Tian Y. Human Body and Posture Recognition System Based on an Improved Thinning Algorithm. *IET Image Processing*. 2011; 5(5): 420-428. DOI: 10.1049/iet-ipr.2009.0303.
- [14] Fujiyoshi H, Lipton AJ. Real-time Human Motion Analysis by Image Skeletonization. In: *Fourth IEEE Workshop on Applications of Computer Vision*; 19-21 October; Princeton, New Jersey. 1998. pp. 15-21. DOI: 10.1109/ACV.1998.732852.
- [15] Hsieh J-W, Chuang C-H, Chen S-Y, Chen C-C, Fan KC. Segmentation of Human Body Parts Using Deformable Triangulation. *IEEE Transactions on Systems Man and Cybernetics Part A-systems and Humans*. 2010; 40(3): 596-610. DOI: 10.1109/TSMCA.2010.2040272.
- [16] Chen C-C, Hsieh J-W, Hsu Y-T, Huang C-Y. Segmentation of Human Body Parts Using Deformable Triangulation. In: *18th International Conference on Pattern Recognition*; 20-24 August; Hong Kong. 2006. pp. 355-358. DOI: 10.1109/ICPR.2006.1035.
- [17] Chuang C-H, Hsieh J-W, Tsai L-W, Fan K-C. Human Action Recognition Using Star Templates and Delaunay Triangulation. In: *International Conference on Intelligent Information Hiding and Multimedia Signal Processing*; 15-17 August; Harbin, China. 2008. pp. 179-182. DOI: 10.1109/IIH-MSP.2008.342.
- [18] Hsieh J-W, Hsu Y-T, Liao H-YM, Chen C-C. Video-Based Human Movement Analysis and Its Application to Surveillance Systems. *IEEE Transactions on Multimedia*. 2008; 10(3): 372 - 384. DOI: 10.1109/TMM.2008.917403.
- [19] Juang CF, Chang CM, Wu JR, Lee DM. Computer Vision-Based Human Body Segmentation and Posture Estimation. *IEEE Transactions on Systems Man and Cybernetics Part A-systems and Humans*. 2009; 39(1): 119-133. DOI: 10.1109/TSMCA.2008.2008397.
- [20] Chen D-T, Liao H-YM, Tyan H-R, Lin C-W. Automatic Key Posture Selection for Human Behavior Analysis. In: *IEEE 7th Workshop on Multimedia Signal Processing*; 30 October-2 November; 2005. pp. 1-4. DOI: 10.1109/MMSP.2005.248572.
- [21] Chen SS, Akselrod P, Zhao B, Carrasco JAP, Linares-Barranco B, Culurciello E. Efficient Feedforward Categorization of Objects and Human Postures with Address-Event Image Sensors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2012; 34(2): 302-314. DOI: 10.1109/TPAMI.2011.120.
- [22] Juang CF, Chang CM. Human Body Posture Classification by a Neural Fuzzy Network and Home Care System Application. *IEEE Transactions on Systems Man and Cybernetics Part A - Systems and Humans*. 2007; 37(6): 984-994. DOI: 10.1109/TSMCA.2007.897609.
- [23] Diraco G, Leone A, Siciliano P. Human Posture Recognition with a Time-of-flight 3d Sensor for In-home Applications. *Expert Systems with Applications*. 2013; 40(2): 744-751. DOI: 10.1016/j.eswa.2012.08.007.
- [24] Brulin D, Benezeth Y, Courtial E. Posture Recognition Based on Fuzzy Logic for Home Monitoring of the Elderly. *IEEE Transactions on Information Technology in Biomedicine*. 2012; 16(5): 974-982. DOI: 10.1109/TITB.2012.2208757.
- [25] Xia L, Chen C-C, Aggarwal JK. View Invariant Human Action Recognition Using Histograms of 3d Joints. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*; 16-21 June; Providence, RI. 2012. pp. 20-27. DOI: 10.1109/CVPRW.2012.6239233.
- [26] Le T-L, Nguyen M-Q, Nguyen T-T-M. Human posture recognition using human skeleton provided by Kinect. In: *International Conference on Computing, Management and Telecommunications*; 21-24 January; Ho Chi Minh, Vietnam. 2013. pp. 340-345. DOI: 10.1109/ComManTel.2013.6482417.
- [27] Patsadu I, Nukoolkit C, Watanapa B. Human gesture recognition using Kinect camera. In: *International Joint Conference on Computer Science and Software Engineering (JCSSE)*; 30 May-01 June 2012; Bangkok, Thailand. 2012. pp. 28-32. DOI: 10.1109/JCSSE.2012.6261920.
- [28] Southwell BJ, Fang G. Human Object Recognition Using Colour and Depth Information from an RGB-D Kinect Sensor. *International Journal of Advanced Robotic Systems*. 2013; 10(171) DOI: 10.5772/55717.
- [29] Granata C, Ibanez A, Bidaud P. Human Activity-understanding: A Multilayer Approach Combining Body Movements and Contextual Descriptors Analysis. *International Journal of Advanced Robotic Systems*. 2015; 12(89) DOI: 10.5772/60525.
- [30] Zhang Z, Liu Y, Li A, Wang M. A novel method for user-defined human posture recognition using Kinect. In: *7th International Congress on Image and*

- Signal Processing; 14-16 October; Dalian, China. 2014. p. 736-740. DOI: 10.1109/CISP.2014.7003875.
- [31] Catuhe D. Programming with the Kinect for Windows software development kit Redmond. 1st ed. Microsoft Press; 2012.
- [32] Pilevar MT, Feili H, Soltani M. Classification of Persian Textual Documents Using Learning Vector Quantization. In: International Conference Natural Language Processing and Knowledge Engineering; 24-27 September; Dalian, China. 2009. pp. 1-6. DOI: 10.1109/NLPKE.2009.5313761.
- [33] Xiao H, Yu L, Chen K. An Efficient Method of Language Identification Using Lvq Network. In: International Conference on Signal Processing; 26-29 October; 2008. pp. 1690-1694. DOI: 10.1109/ICOSP.2008.4697462.
- [34] Wang C, Zhang H, Yu C Research on Color Recognition of Urine Test Paper Based on Learning Vector Quantization (LVQ). In: International Conference on Instrumentation, Measurement, Computer Communication and Control; 8-10 December; Harbin, China. 2012. pp. 850-853. DOI: 10.1109/IMCCC.2012.205.
- [35] Kinect for Windows [Internet] Available from: <http://www.microsoft.com/en-us/kinectforwindows/develop/>. Accessed on 11 Sep 2015.