Sitting Posture Detection using Adaptively Fused 3D Features

Sun Bei[1], Zeng Xing[1], Liu Taocheng[1], Lu Qin[1]
1. College of Mechatronics Engineering and Automation, National University of Defense Technology
Beys1990@163.com, 1019289020@qq.com, 469683093@qq.com, freda0126@sina.com

***Abstract*—This paper presents a sitting posture detection method based on a depth sensor. Effectively detecting bad sitting posture can help protect eyesight and prevent cervical spondylosis. Yet most current approaches conduct the sitting detection with traditional sensors, which leads to inferior performance. Therefore, this paper introduces a novel method based on a depth sensor, in which we 1) model the sitting posture by splitting body profile and locating six body joints, 2) extract two kinds of features: relative topological characters and local edge features, and 3) jointly fuse the two features for sitting posture detection. The proposed method is tested on our 3D sitting posture dataset, which includes 18 individuals, and totally 9 different sitting postures. The experimental results show that, using an adaptively fusion of the joints features, this method performs a high accuracy (more than 94%) and achieves a real-time detection (17 fps).**

***Index Terms*—depth image, sitting posture modeling, sitting posture detection**

I. INTRODUCTION

Sitting for a long time with bad postures can afflict harm on people's physical and mental health, which seriously causes myopia, waist cervical disease and other ailments [1]. However, timely detecting the bad posture can help prevent this depth damage on people's health. Moreover, recent studies have proved that jointly analysis the relationship between sitting habits and learning efficiency is of great significance, e.g., an stable sitting posture always represent a concentration focused statues [2].

Recent sitting posture detection approaches can be divided into two categories: 1) traditional sensor-based method, and 2) image processing-based method. The traditional sensor-based approaches require the subjects wearing a specific sensors, e.g., acceleration sensor, pressure sensor, which conducts the identification with the sensor data [3-5]. The most typical application is Tapia et al [6] exploiting five wireless acceleration sensors to place in the shoulder, wrist, but-tocks, thighs and ankle joints to identify the postures such as standing, sitting, walking, and running. However, though these methods are relatively simple, but it is inconvenience in practical application and easily susceptible to external interference. Moreover, traditional sensor-based methods is always used for motion detection, while not so effective for static sitting posture detection. The image processing-based method employs a camera to locate human body and conduct detecting using image features[7-9]. Compared with traditional sensor-based methods, the image processing-based approaches are more suitable for practical applications. Especially with the development of depth sensors, it is more accurate to make a 3D modeling for human posture, which is not affected by external environment and light variations. Raptis et al [10] use a Kinect sensor to obtain body joints, and based on these body joints they extract the skeleton angles features for posture classification applications. Nowadays, jointly using 3D sensors to resolve computer vision problems becomes a research topic, e.g., face detection, action recognition and human-computer interaction, yet very few studies focus on sitting posture detection.

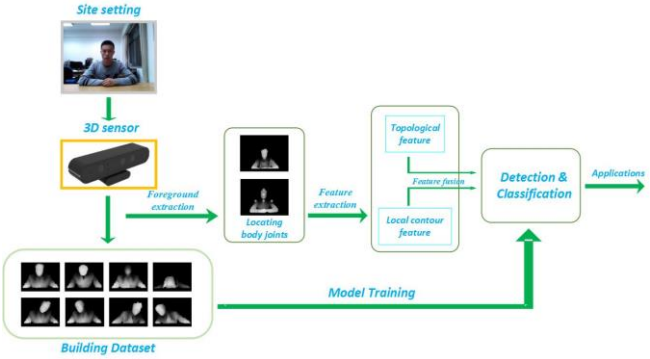

Fig.1 The proposed sitting posture detection method.

So, this paper presents a detection method using a 3D sensor. Aiming at applying for embedded platforms, while as the Kinect sensor requires a high configured environment, this paper adopts the Astra 3D sensor for data collection, which supports the Windows, Linux and Android. Fig. 1 shows the proposed detection method, which includes: 1) conducting a data reprocessing to remove interference and extract the body contours, 2) locating body joints and based on this to extract two different features: the relative topological characters and local edge features, and 3) using the KNN method to identify sitting postures. Moreover, this paper further builds a 3D sitting dataset. We totally invite 18 individuals to participate in the data collection, in which each person is asked to perform 9 posture and each with 100 images, such as upright, head down, left partial, right partial, and so on. Finally, the proposed method is tested by using 1) different features individually, and

This work is supported by the National Natural Science Foundation of China (61171136).

Bei Sun is with the National University of Defense Technology, Changsha, China (beys1990@163.com).

2) adaptively fused features. Finally, the comprehensive comparison with other state-of-the-art algorithms shows that: the proposed body joints-based method has a higher detection accuracy, what's more with a real-time detection speed.

II. 3D SENSOR AND SITTING DATASET

3D sensors can output RGB and depth image, which can be better used for modeling body postures than ordinary RGB camera. As a typical depth sensor, Kinect sensor has shown widely application prospects in human-computer interaction, motion capturing and human behavior recognition[11][12]. However, as it has a high requirements of hardware: 1) win8 above operating system, 2) usb3.0 inter-face and 3) i7 above CPU, which makes it not available on most embedded plat-forms. Therefore, this paper adopts the Astra3D sensor for sitting posture detecting. Compared with Kinect, the Astra3D sensor has a smaller size, and can sup-port multi-platforms such as Windows, Android and Linux. Furthermore, the As-tra3D sensor has an effective measuring range from 0.6m to 8m and an accuracy of 1m ± 1-3mm, which meets the requirements of sitting posture detection very well.

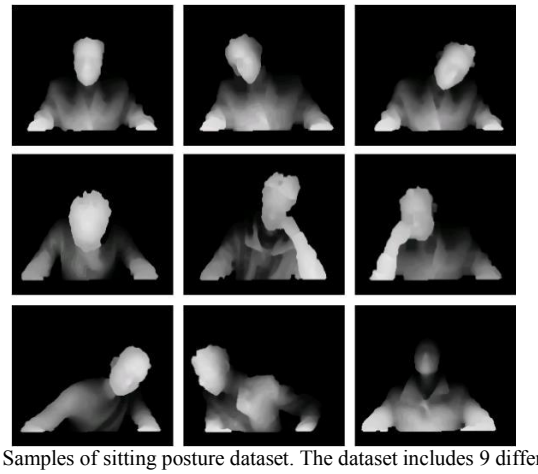

Fig.2 Samples of sitting posture dataset. The dataset includes 9 different postures, from left to right and from up to down respectively are: upright, right partial, left partial, head down, left hand cheeks, right hand cheeks, body left oblique, body right oblique, and head up.

Moreover, the value depth image does not denote the real depth: the depth image is the results converting the real depth to the grayscale of 0-255. So, to make a more accurate measurement, this paper utilizes the real depth for sitting postures detection. As Fig.1 shows, a 3D Astra sensor is fixed in front of a desk (about 1m away from the chair). During collection, each individual is asked to sit before the desk. Finally, totally 18 people are invited to participate in the collection. Fig.2 illustrates the 9 postures, including upright, upright, right partial, left partial, head down, left hand cheeks, right hand cheeks, body left oblique, body right oblique, and head up.

III. FEATURE AND DETECTION

A. Foreground Extraction

Most algorithms based on a background frame to conduct the foreground segmentation, in which they use the current frame to subtract a background frame, and based on the changing parts to locate the dynamic target. In such methods, it is very critical to accurately model the background frame. Recent approaches can be divided into 3 categories: 1) using a fixed background frame, 2) using the last frame, and 3) employing a dynamic Gaussian model [13] of the last few frames.

This paper adopts an improved foreground segmentation approach based on the third method. Considering the special scene of sitting state, which is shown in Fig.3, we can make the following supposes:

1) A distance exists between the people and surrounding background;

2) No other obstructions exists between the body and the camera;

3) The body always appears in the middle of image field;

4) The depth of a target would not change widely between two adjacent frames.

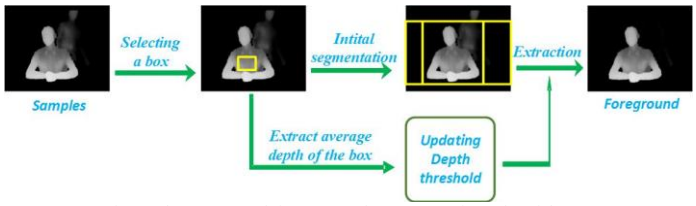

Fig.3 The proposed foreground segmentation algorithm.

Fig.3 presents the foreground segmentation algorithm, which including 4 parts:

1) Detecting the sitting happening. According to the site settings, people would sit about 1m away from the 3D sensor, so this paper supposes the subjects would appears in the distance of 0.8m to 1.2m. Based on this, we firstly conduct an initial reprocessing: set the pixel to 1 if its value is range from 0.8m to 1.2m, or else set to 0. Based on this we count the histogram of pixel width, and then judge whether someone is sitting.

2) Extracting the depth threshold of current frame. When someone sit, this paper: 1) extracting an block from the middle part of image (denoted as $[x, y, w, h]$, where (x, y) is the top left position, (w, h) denotes the width and height), 2) using a smoothing filter[14] in the following equation to remove singularity, and calculating the average depth as the threshold.

$$y(k) = \frac{1}{n}\sum_{i=-m}^{i=m} x(k+i)$$

where $m = (n-1)/2$ when n is an odd number and $m = n/2$ when n is an even number.

3) Updating the depth threshold based on a Gaussian model[13]. When computing a depth threshold, we based on the following formula to update the depth threshold, in which the weight is set as $\{0.1, 0.1, 0.2, 0.2, 0.4\}$.

$$Tr = \sum_{i=1}^{5} w_i * Tr_i$$

4) Conducting foreground extraction using the depth threshold. As illustrated in Fig.4, the proposed method can effectively split the foreground.

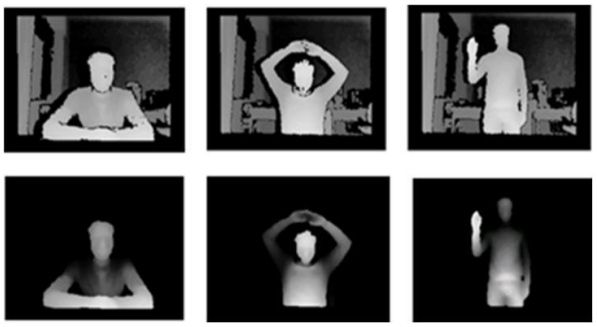

Fig. 1 The illustrations of foreground extracting results.

B. *Locating Body Joints*

Compared with extracting more features for detecting applications, it is more critical to exploit more significant characters, moreover, using less feature dimensions can help reduce the calculation. Therefore, aiming at using less features while acquiring high accuracy, this paper designs a novel feature model based on body joints, which includes the head vertex joint A, the head center joint B, the left shoulder joint C1, the shoulder center joint C2, the right shoulder joint C3, and the trunk center joint D.

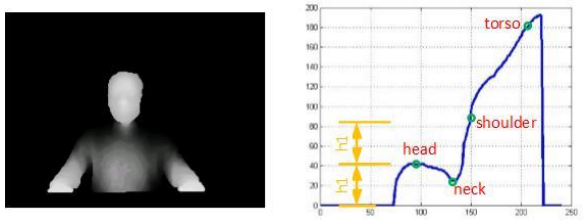

Fig. 5 The histogram of body contour width.

Given the body contours, we firstly count its width histogram. Fig.5 presents a histogram result, from which we find: 1) the width of neck is smaller than other parts, such as head and shoulder, 2) the width from neck to shoulder would significantly increase. Based on this, we firstly make an initial segmentation of the head and shoulder, and then employ the Least Mean Square Distance algorithm [15] shown in the following formula to accurately locate head.

$$\mathrm{R} = \frac{\sum_{x \in X} dis(x, Y) + \sum_{y \in Y} dis(y, X)}{Nx + Ny}$$

where X denotes the searching sub-graph, Y is the template, a is a point of X, b is a point belonging to Y, $\mathrm{d}is(i, \mathrm{J})$ denotes the least distance of point i to graph J, Nx and Ny respectively denote the number of X and Y. When R reaches the minimum value, the algorithm considers it is the head part.

When locating the head, this algorithm seeks the body joints as following:

1) Down traversing along the image, and stopping researching when the contour width is 2 times of head center joints, which we consider it is the shoulder.

2) Respectively counting the left, right and center joints of shoulder.

3) Continue traversing down to the bottom of the body contour, and taking the center of bottom 10 pixels as the trunk center joint.

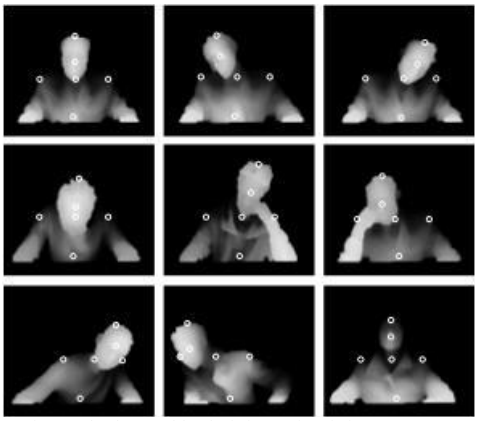

Fig. 2 The located body joints of 9 sitting postures.

Fig.6 illustrates the located body joints of 9 sitting postures. From which we find the joints would locate significant different in each sitting posture, e.g., when head tilts, the angle of head center and shoulder joint would be various changed, when body tilts, the line of head center joint to stunk center joint will have a clear shift, when lying on the desk, the average body depth would be significant smaller than normal sitting. So, based on this, we conduct the sitting posture detection.

C. *Feature Extraction and Fusion*

Using more different feature patterns can help improve the recognition performance, so given the 6 body joints, this paper develops the extraction of two different features: 1) the local edge features of each joint, and 2) the relative topological between joints.

Local edge features. The edge detection algorithms is based on a basic operator to calculate the significant local variation of the images. Nowadays, various operators such as Gradient operator, Sobel operator, Laplace operator and Canny operator, have widely used for detection and recognition applications. This paper adopts the Sobel operator for detection [16], in which we firstly locate a 20*20 block centered at each joint, and then conduct the edge detection.

The Sobel operator is based on two filter operators to calculate the horizontal and vertical edges of original image.

-1	0	1
-2	0	2
-1	0	1

Ga

1	2	1
0	0	0
-1	-2	-1

Gb

Fig. 3 The Sobel convolution operators. Ga and Gb are respectively used for horizontal and vertical directions.

Suppose X denotes an image block, G_x and G_y represent the calculated vertical and horizontal results. We conduct the calculation by followings:

$$G_x = G_a * \mathrm{X}$$
$$G_y = G_b * \mathrm{X}$$
$$|G| = |G_x| + |G_y|$$
$$\theta = \tan^{-1}(\frac{G_y}{G_x})$$

Finally, if $|G|$ is more than a threshold, we consider point (x, y) as the edge point and set its grayscale to 255, or else set to 0.

Relative topological characters. Tab.1 reports the 6 extracted parameters, in which the angle parameters are calculated using the following formula, where (P_x, P_y) is the coordinate of P, (Q_x, Q_y) denotes the coordinate of Q.

$$\theta(\mathrm{P},\mathrm{Q}) = \tan^{-1}\left(\frac{|P_y - Q_y|}{|P_x - Q_x|}\right)$$

Table 1 Relative topological parameters.

Parameters	Definition
H1	The depth of head top joint A
H2	The depth of head center joint B
H3	The depth of body trunk joint D
L1	The horizontal angle of line "joints A- joints B"
L2	The horizontal angle of line "joints C2- joints D"
L3	The vertical angle of line "joints A- joints B"

Feature fusion. Recent feature fusion methods can be divided into two categories: 1) firstly conducting feature fusion and then inputting the fused feature into classifiers, and 2) separately conducting classification using different features and then making a comprehensive judgment. This paper adopts the second fusion method, in which we firstly exploit each feature to obtain a classification result, and then make a voting to obtain a comprehensive ranking, lastly the highest ranking results would be selected as the classification.

D. Detection based on KNN

Recent machine learning algorithms can be divided into two types: supervised learning method and unsupervised learning method. Compared with unsupervised learning, the supervised learning requires a learning process of samples, which can output a more stable and credible performance. Nowadays, numerous supervised learning algorithms such as SVM, KNN, HMM, random forest and ad-boost have been applied in object detection, tracking and recognition applications. Each algorithm has its own advantages, where KNN is a very simple and effective method for object detection, moreover, it is very suitable for small samples detection applications [17]. So, this paper adopts the KNN for detection task. The KNN exploits the distance to measure the similarity of two samples[18], as list in the following formula:

$$L(\phi_1, \phi_2) = \sum_{i=1}^{n} \sqrt{(\phi_1(i) - \phi_2(i))^2}$$

where ϕ_1 and ϕ_2 denote two samples, and n is the dimension. The k smallest distances are calculated as the k probable neighbors, finally, the nearest neighbors are selected as the category of the unknown samples.

IV. EXPERIMENT AND ANALYSIS

We test the proposed algorithm on our own dataset, which includes 18 individuals, and each with 9 sitting postures. In the experiments, we randomly select one third frames from each posture (about 600 frames) for training, and then utilize the rest samples for testing.

Tab.2 reports the average values of 4 different parameters, in which we firstly extract each frames' value, and then compute the average results. As Tab.2 shows, a significant difference exists between different sitting postures. Such as the H2 parameter between upright and head up, the L1 parameter between left partial and right partial. To make a further analysis, this paper carries out the classification. Given the statistics results of each frame, we firstly conduct a reprocessing using a median filter, and then regard them as models. In the classification, each sample is compared with the models using KNN, and the model with the least distance would be regarded as the judgment. Tab.3 reports the final results. As Tab.3 shows, each posture obtains a high accuracy (>85%), while postures such as left partial, right partial, body left oblique and body right oblique perform a little worse. Note that the head partial and body oblique may exist an unclear boundaries, which may lead a confused classification. Furthermore, this judgment is limited to camera location, e.g. parameters H2 and H3. So it is not complete to conduct detection only using the relative topological characters.

Tab.3 reports the classification results using local features. Compared with the results of using topological characters, postures such as head partial and body oblique acquires a higher accuracy (>88%), which represents the local features can help distinguish these confusions. While the performance of postures such as upright and head down are even worse. Yet note that each individual may have different body size, which may results these errors.

Tab.3 also presents the results by using the fused features, which shows a significant improvement on classification. However, we find the accuracy of head down posture is higher than body oblique. Note that during data collection, we just tell the participants to perform different postures, yet each one may have his own understanding and make a different action. Furthermore, as shown in Fig.6, there are some similarities in postures such as body oblique and head partial, which makes them easily confused. This is which we called the error of samples, so it is urgent to build a larger and more robust dataset for depth analysis. Furthermore, the proposed method costs only 58.30ms to detect a frame, which is basically meet the real-time requirement.

Additionally, this paper also makes a comparison with the state-of-the-art methods. As listed in Tab.4, the proposed method outperforms other methods: 1) the proposed method detects more sitting postures, and 2) by using the fusion of joints features based on depth image, the method obtains a higher average accuracy. Though our method performs a slightly worse than Huang et.al, yet note that our method tests more postures, which is acceptable in practical applications.

Table 2 The comparison with other approaches.

Method	Postures	Feature	accuracy
WU et.al	8	Skin-Color Features	84.92%
Yuan et.al	7	Fusion of Skin-color and SURF	93.70%
Huang et.al	3	Contextual Feature of Depth Image	96.53%
Ours	9	Fusion of Joints Feature	95.80%

Overall, the proposed method using a cross-platform 3D sensor, moreover, it costs only 58.30ms to detect a frame and almost achieve the real-time detection, which can be further

applied for embedded sitting posture detection.

V. CONCLUSION

Bad sitting posture is an important factor to cause myopia and lumbar cervical disease. Based on this, this paper designs a real-time sitting posture detection method using an embedded 3D sensor. The proposed method includes: 1) conducting foreground segmentation using a Gaussian mixture model, 2) locating six body joints, 3) extracting two features including topological characters and local edge features, 4) carrying out classification using a KNN classifier.

The other contributions of this paper are: 1) we carried out a real-time detection method using a cross-platform 3D sensor, which can be quickly transplant to the embedded system, 2) we introduced a fast foreground segmentation algorithm for sitting applications, and 3) we proposed a novel extraction of the effective joints features based on the extracted body contours. The experiments show that, the proposed fused joints features method leads to a higher accuracy than other state-of-the-art methods. Moreover, the proposed method can be further applied for real-time sitting posture detecting system. Certainly, detecting sitting posture is a difficult task as the external variation influence. Our ongoing works will focus on 1) making a more robust foreground segmentation in real applications, and 2) building a larger dataset for depth analysis and 3) analyzing the relationship between sitting postures and learning efficiency.

Acknowledgments. This work is supported by the National Natural Science Foundation of China (61503398).

REFERENCES

1. W. Dankaerts, P.O. Sullivan, A. Burnett, L. Straker. Differences in sitting postures are associated with nonspecific chronic low back pain disorders when patients are subclassified[J]. Spine, 2006, 31(6):698-704.
2. A.C. Grunseit, J.Y. Chau, V. Rangul, et al. Patterns of sitting and mortality in the Nord-Trøndelag health study (HUNT)[J]. International Journal of Behavioral Nutrition & Physical Activity, 2017,14(1):8.
3. K. Kamiya, M. Kudo, H. Nonaka, J. Toyama. Sitting posture analysis by pressure sensors[J].International Conference on Pattern Recognition. IEEE, 2008:1-4.
4. J. Meyer, B. Arnrich, J. Schumm, G. Troster. Design and Modeling of a Textile Pressure Sensor for Sitting Posture Classification[J]. IEEE Sensors Journal, 2010, 10.
5. L. Martins, R. Lucena, J. Belo, R. Almeida, et al. Intelligent Chair Sensor – Classification and Correction of Sitting Posture[J]. Springer International Publishing, 2014, 41.
6. E.M. Tapia, S.S. Intille, K. Larson. Activity Recognition in the Home Using Simple and Ubiquitous Sensors[J]. International Conference on Pervasive Computing, 2004, 3001:158-175.
7. S.L. Wu, R.Y. Cui. Human Behavior Recognition Based on Sitting Postures[J]. International Symposium on Computer, communication, Control and Automation Proceedings. 2010:138 -141.
8. D.B. Yuan, Y. Dai, T.Q. Chen. Multi-feature fusion recognition of incorrect sit posture [J]. Computer Engineering and Design, 2017, 38.
9. J.Y. Huang, S.C. Su, C.L. Huang. Human upper body posture recognition and upper limbs motion parameters estimation[J].Signal and Information Processing Association Summit and Conference. 2013:1-9.
10. Raptis, Michalis, Kirovski, et al. Real-time classification of dance gestures from skeleton animation[J]. The Eurographics Association, 2011:147-156.
11. Z. Liu, L. Zhou, H. Leung, et al. Kinect Posture Reconstruction based on a Local Mixture of Gaussian Process Models[J]. IEEE Transactions on Visualization & Computer Graphics, 2016, 22.
12. L.B. Neto, F. Grijalva, et al. A Kinect-Based Wearable Face Recognition System to Aid Visually Impaired Users[J]. IEEE Transactions on Human-Machine Systems, 2016:1-13.
13. W. Fu, M. Johnston, M. Zhang. Genetic programming for edge detection: a Gaussian-based approach[J]. Soft Computing, 2016, 20(3):1231-1248.
14. H. Felle, E. Johannes. A chronology of interpolation: from ancient astronomy to modern signal and image processing[J]. Proceedings of the IEEE, 2002,90(3):319-342.
15. W. Liu, P.P. Pokharel, J.C. Principe. The Kernel Least-Mean-Square Algorithm[J]. IEEE Transactions on Signal Processing, 2008, 56(2):543-554.
16. Y.D. Qu, C.S. Cui, S.B. Chen, J.Q. Li. A fast sub-pixel edge detection method using Sobel – Zernike moments operator[J]. Image & Vision Computing, 2005, 23(1):11-17.
17. W.J. Hwang. Fast KNN classification algorithm based on partial distance search[J]. Electronics Letters, 1998, 34 (21):2062-2063.
18. J. Lu, Z.W. Wu, Y. Wang, Y. Lu. Research on Abnormal Behavior Detection Based on KNN Algorithm[J]. Computer Engineering, 2007, 33(7):133-134.