Smart Cushion: A Practical System for Fine-grained Sitting Posture Recognition

Guanqing Liang, Jiannong Cao, Xuefeng Liu
Department of Computing
The Hong Kong Polytechnic University
Email: csgliang@comp.polyu.edu.hk, csjcao@comp.polyu.edu.hk, csxfliu@comp.polyu.edu.hk

***Abstract*—Poor sitting postures influence one's health and can cause upper limb and neck disorder. Current solutions for siting posture recognition, however, are impractical due to intrusiveness, high cost or low generalization capability. Particularly, most of the existing solutions are chair-dependent, which are highly coupled with certain types of chairs. In this paper, we design Postureware, a smart cushion, which is a low-cost, non-intrusive and general sitting posture recognition system. In particular, Postureware incorporates very thin pressure sensors to offer non-intrusive experience, an effective sensor placement solution to reduce cost, a set of user-invariant features and an ensemble learning classifier to improve generalization ability. We implement a prototype system and conduct extensive experiments. The results show that Postureware can classify fifteen fine-grained postures with high accuracy.**

I. Introduction

In modern lifestyle, many people spend prolonged periods of time sitting. Research shows that long periods of physical inactivity raise the risk of heart disease, diabetes, cancer, and obesity. Moreover, poor sitting postures such as leaning forward cause upper limb and neck pain [1]. To reduce risky sitting behaviours, monitoring people's sitting posture is greatly needed, which can raise people's awareness of sitting behaviours. In addition, the rehabilitation prognosis of stroke patients can be conducted by measuring the sitting imbalance [2].

Current solutions for siting posture recognition, however, are impractical due to intrusiveness, high cost or low accuracy. A wearable sensor-based method attaches inertial sensors on a user's back to collect motion data and identify the sitting posture [3] [4]. This approach is intrusive and brings discomfort to users, since users are required to wear or attach those sensors. Vision-based methods use cameras to capture user sitting postures [5] [6]. Such method requires line-of-sight condition and may raise privacy concerns. To achieve a non-intrusive and privacy-preserving solution, researchers have deployed pressure sensors on the chair to infer sitting posture [7][8]. But the main limitation of these solutions are the high cost (around 3000 USD), since they rely on high-fidelity pressure sensor array with more than two hundred sensors [9]. Note that paper [10] presents a solution to deploy pressure sensors on an ergonomic chair to recognize sitting postures based on the sitting pressure. However, this solution highly depends on certain types of chair and difficult to be generalized to other user scenarios.

The objective of this paper is to enable a practical sitting posture recognition technique that is accurate, non-intrusive, low-cost and real-time. We introduce a smart cushion called Postureware, a sitting posture recognition system that can identify fifteen representative sitting postures accurately in real-time using pressure sensors that are deployed inside a seat cushion.

The main contributions of the paper are as follows:

- We design and implement a non-intrusive system for sitting posture recognition, which can accurately recognize sitting posture in real-time (8 Hz) with low cost (150 USD).
- We propose an information-theoretic sensor placement solution, enabling the system to achieve the same recognition accuracy using much fewer number of sensors.
- We design an accurate sitting posture recognition model, which incorporates pre-processing, user-invariant feature extraction and AdaBoost classification, being able to resolve the challenges posed by user diversity and thus improving generalization accuracy. Specifically, the system can recognize 15 categories of sitting postures with 98% ten-fold cross validation accuracy with 10 sensors. The system can achieve 85% generalization accuracy with 10 sensors.

II. System Overview

This section discusses the technical considerations that underpins the design of Postureware. Specially, the considerations include sitting posture set and system requirement.

A. Sitting Posture Set

Specifically, we target at fifteen sitting postures performed by the subjects, including: (1) sitting upright; (2) slouching; (3) leaning back; (4) leaning forward (angle<30 degrees); (5) leaning forward (angle>45 degrees); (6) leaning left (angle<10 degrees); (7) leaning left (angle>20 degrees); (8) leaning right (angle<10 degrees); (9) leaning right (angle>20 degrees); (10) left leg crossed in ankle; (11) left leg crossed in knee; (12) right leg crossed in ankle; (13) right leg crossed in knee; (14) left leg crossed, leaning right; (15) right leg crossed, leaning left.

978-1-5090-4338-5/17/$31.00 ©2017 IEEE

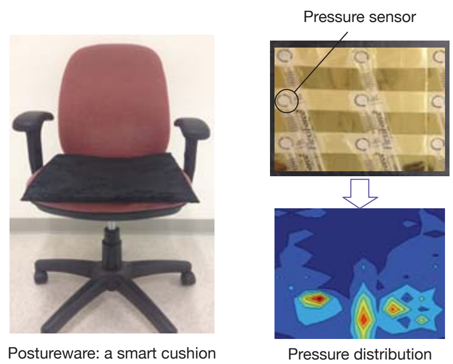

Fig. 1: Postureware overview

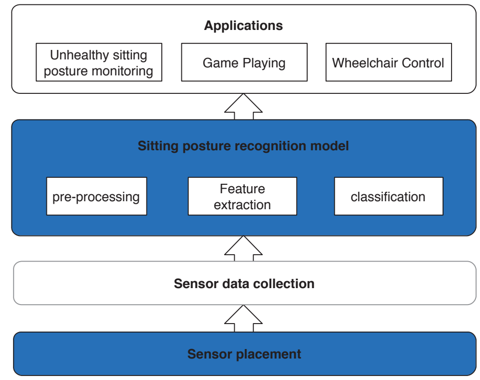

Fig. 2: Postureware architecture

Compared with the posture set in previous works [10][8], the siting posture set in this paper is more fine-grained and representative. As a result, the sitting posture recognition problem in this work is more challenging.

B. System Architecture

The delivery of Postureware system is a sensing mat that can recognize user's sitting posture. The sensing mat leverages the pressure sensors, as shown in Figure 1, to obtain the pressure distribution on top of the chair. Then the sitting posture can be inferred based on the distinctive pressure distribution pattern. Postureware contains four components: pressure sensors, sensor placement, sitting posture recognition model and application, which is shown in Figure 2. On top of sitting posture recognition, we develop three applications: first application is to monitor unhealthy sitting posture, second one is to use sitting posture to play a car racing game and the last application is to control wheelchair by changing sitting posture. In particular, Postureware contains two key components: (1) sensor placement; (2) sitting posture recognition model.

Sensor placement. To reduce system cost, we need to place as fewer number of sensors as possible, as each pressure sensor costs around 15 USD [9]. This component is designed to output a number of positions on the chair top to place sensors, aiming to minimize the number of deployed sensors while achieving the required recognition accuracy.

Sitting posture recognition model. Given the original pressure sensor data, a sitting posture recognition model is designed to infer the sitting posture accurately. Specifically, sitting posture recognition model consists of three components: pre-processing, feature extraction and classification. The first two components aim to represent sitting posture using distinctive and user-invariant patterns, whereas classification component is used to train the classifier to identify the posture based on features.

III. Postureware Framework

A. Sensor Placement

Sensor placement focuses on deploying as fewer number of sensors as possible to obtain the required accuracy of recognition. More specifically, the sensor placement problem is how to select a few number of sensors from the densely deployed sensor grid. This problem has been studied in literature [10]. The evaluation metric for placement solution is how fit the placed sensors can reconstruct the features that are computed from high-fidelity sensor array. Generally, sensors on the boundary of contact area are preferred to select under this evaluation metric, since most of the selected features are related to the shape of contact area (e.g., size, position, center of contact area, distance of contact area to the edges of the chair).

Algorithm 1: Information-theoretic sensor placement algorithm

input:
dataset $D = \langle (s_1, y_1), ..., (s_m, y_m) \rangle$ with sensor data vector s_i and corresponding posture labels $y_i \in Y = \{1, ..., n\}$.
total sensor set S.
deployed sensor number N.
output:
selected sensor set t S_S.
Initialization:
selected sensor set $S_S = \emptyset$;
unselected sensor set $S_U = S$;
for $t = 1, ..., N$ **do**
  for $s_i \in S_U$ **do**
    $H(D) = -\sum_i P(D_i) \log P(D_i)$;
    compute $H(D, s_i) = H(D) - H(D|s_i)$;
  end
  $s_m = \arg\max_{s_i} H(D, s_i)$;
  $S_S = S_S \cup \{s_m\}$;
  $S_U = S_U \backslash \{s_m\}$;
end

However, we argue that this sensor placement solution can be further improved due to the following reasons. First, due to diversity of people, features such as position and size of contact area between people and the chair varies, making the system difficult to generalize. Second, we found that discriminative sensors are normally located inside rather than on the boundary of contact area through observing the pressure distribution of sitting postures. This observation implies that an effective sensor placement solution does not necessarily reconstruct the features well.

In order to place sensors in the positions that capture the distinctive data, we design an information-theoretic sensor placement algorithm. At a high level, this algorithm works like a greedy algorithm. First, it evaluates the performance of all the unselected sensors based on the information gain. Information gain of a sensor is measured by the mutual information between its sensory data and the data from all the unselected sensors. Then the algorithm will include the sensor with the highest information gain into the final sensor placement solution. The details of the algorithm is presented in algorithm 1. The rationale of this method is that larger information gain of a sensor indicates it contains more distinctive information, and thus improving the classification accuracy. This placement solution has two advantages: first, it is fast to compute; second, it is able to extract informative sensors, which is independent of the selected features and classification algorithms.

B. Pre-processing

Pre-processing is the first module of sitting posture recognition model. Once the pressure sensors are deployed, the pressure sensor data will be collected. However, the original data cannot be directly used for learning due to three main reasons. Firstly, the data is not calibrated, since the initial value of pressure sensor changes when no force is applied. Secondly, the data may not reflect user's sitting posture because user might not sit on top of all the deployed sensors. Thirdly, the data value varies due to user's diverse weight. In order to address the above issues, we design a pre-processing module, aiming to calibrate, extract and standardize the original data.

Data calibration. To calibrate the sensor data, we first record the initial value of sensors when no force is applied. Then data can be calibrated simply by using the sensor data to subtract the initial value of corresponding sensor.

Contact area data extraction. Contact area data refers to the data of sensors that are activated by users while sitting on top of the chair. Contact area data extraction is to crop the sensors where covers all the data from the contact area between the subject and the chair. To extract contact area data, we adopt a threshold-based method that determines the sensors contacted by users.

Data normalization. Data needs to be further nomalized to mitigate the effect caused by various weights of users. In this work, we transform the data into standard normal distribution.

C. Feature Extraction

Feature extraction is to extract a set of features from the sensor data to represent sitting posture, which is critical to the recognition accuracy. Most of the features proposed in previous works [8][10][11] are chair-dependent, as the data are collected from the deployed pressure sensors in top and back of a certain type of chair. The major difference in this work is that we purely rely on a cushion which will be placed on top of a chair. Therefore, the features should be distinctive regardless of chair types.

Category	Feature
position-based feature	Gravity, Top-k, Bottom-k
ratio-based feature	X-axis block ratio, Y-axis block ratio

TABLE I: Feature category

The goal of feature extraction is identify both user-independent and chair-independent features. The philosophy of feature extraction in this work is based on two principles. First principle is that feature should be computed in a relative manner. The insight of this principle is to enable the features to be user-invariant. For instance, the positions of top-k sensors will be considered as the relative positions of top-k sensors with respect to a reference point. Second principle requires the feature to indicate the pressure redistribution triggered by the sitting posture. This principle is inspired by the observation that different postures in essence results in the pressure redistribution of sensors covered by human body.

Our features mainly represent two kinds of information as shown in Table I: (1) position-based feature, which represents the relative position of certain points. (2) ratio-based feature, which measures the pressure ratio between regions of points. To make the analysis more convenient, the seat cushion is considered as a two-dimension Cartesian coordinate system with x, y axes. The center of seat cushion is considered as origin, while the direction of y axis is set to the front direction of chair. If the location (x, y) has been deployed with a sensor, then it is denoted as $s(x, y) = 1$, and the corresponding sensor value is represented as $d(x, y)$. In particular, position-based feature include the following features.

Reference point. To make features user-independent, we need to set up a reference point and thus enabling the relative measurement with respect to the reference point. In our case, the center of sensor positions (o_x, o_y) is selected as the reference point. It can be calculated as follows:

$$o_x = (\sum_{\forall s(x,y)=1} x)/(\sum_{\forall s(x,y)=1} 1) \tag{1}$$

$$o_y = (\sum_{\forall s(x,y)=1} y)/(\sum_{\forall s(x,y)=1} 1) \tag{2}$$

Gravity. It refers to the center of pressure distribution on sensors. We denote the pressure data in position (x, y) as $d(x, y)$. Then the position of gravity (g_x, g_y) can be calculated based on the following equations:

$$g_x = (\sum_{\forall s(x,y)=1} x \cdot d(x, y))/(\sum_{\forall s(x,y)=1} d(x, y)) - o_x \tag{3}$$

$$g_y = (\sum_{\forall s(x,y)=1} y \cdot d(x, y))/(\sum_{\forall s(x,y)=1} d(x, y)) - o_y \tag{4}$$

Top-k sensors. Top-k sensors refer to k sensors with the highest pressure value. In particular, It extract three kinds of features: the center, the positions and variance of positions of top-k sensors. The center of top-k sensors can be computed using the same equations of gravity. The variance of positions can be measured by summing up the variance of top-k sensor positions along x-axis and y-axis.

Bottom-k sensors. Bottom-k sensors means k sensors with the lowest pressure value. Similar to top-k sensors, three kinds of features are extracted from bottom-k sensors: the center, the positions and variance of positions.

Another category of features are ratio-based feature, which includes features as below:

X-axis block ratio. This set of features R_x mainly measures the pressure difference along x-axis direction. Specifically, we measure the average value ratio among sensor blocks along with x-axis (vertical to chair front direction).

$$R_x = \{r_x | m_{x+n}/m_x, \forall_x m_x \neq 0\} n = 1, 2, ... \tag{5}$$

$$m_x = (\sum_{\forall y, s(x,y)=1} d(x, y))/(\sum_{\forall y, s(x,y)=1} 1) \tag{6}$$

where m_x represents the average pressure in the xth sensor row along x-axis direction.

Y-axis block ratio. Similar to X-axis ratio, this set of features measure the pressure difference along y-axis direction (same to chair front direction).

After feature extraction, however, some features may be redundant or even ineffective. Therefore, we need to select a set of distinctive features to best represent the posture. To perform feature selection, we adopt one commonly used method called information gain [12], which is independent of classification algorithm. The information gain of each feature with respect to the sitting posture will be calculated and feature with the highest value will be selected. Then keep repeatedly conduct previous step for the unselected features until classification accuracy has not increased.

D. Classification

The first two modules of sitting posture recognition model aim to construct representative features, while classification module is designed to train a classifier to identify the sitting posture based on the features. To further improve the generalization ability of the classifier, we adopt AdaBoost [13] as our classification algorithm. The main advantage of AdaBoost is that the error on the training dataset can be arbitrarily small and yet the generalization error is low.

Primarily, AdaBoost is an ensemble method, which combines a number of different classifiers. The main insight of AdaBoost is repeatedly running a given weak learning algorithm on various distributions over the training data, and then combine the classifiers trained in each turn into a single classifier. In this work, the input of AdaBoost algorithm is a training dataset $\langle(x_1, y_1), ..., (x_m, y_m)\rangle$ with feature vector $x_i \in X$ and corresponding posture labels $y_i \in Y = \{1, ..., n\}$; the output of the AdaBoost algorithm is a composite classifier $c_f(x)$. The details of AdaBoost algorithm is presented in Algorithm 2.

The selection of weak classifier algorithm is very critical to the performance of AdaBoost. We will evaluate the performances of several commonly used weak classifier algorithms including decision stump, naive Bayes and C4.5. The performance of different weak learning algorithms will be evaluated in the evaluation section.

- Decision stump. Decision stump is a learning model that is a one-level decision tree.
- Naive Bayes. Naive Bayes is a parametric classifier based on Bayesian Theorem with assumption that all features are independent.
- C4.5. C4.5 is a decision tree algorithm which is commonly used as a weak classifier in AdaBoost.

IV. Evaluation

A. System Implementation

Postureware incorporates three main hardware components: a pressure sensor array, a data sampling module and a data processing board. As discussed earlier, the pressure sensor array is used to obtain sensor value, whereas a sampling module is used to collect data with certain frequency. After receiving data from the data sampling module, the data processing board executes the algorithms of sitting posture recognition and delivers the identified sitting posture to applications.

Initially, pressure sensors is deployed in 5*8 grid and the distance between neighbouring sensor is 4 cm vertical to the chair front direction and 6cm along the chair front direction. FlexiForce A201 [9] is chosen as our pressure sensor, because it's very thin (0.208 mm thick) and very stable (less than 3% linearity error), which can provide non-intrusive user experience. In data sampling unit, the pressure data is encoded as a 16-bit digital value and the data sampling rate is set as 115200 Baud/s. Data processing board is essentially an arm development board (EBD2410), where runs the sitting posture recognition algorithms and then output the recognized sitting posture. We implement the algorithms using c++ language. The communication between the development board and other devices is done with socket programming.

B. Experiment Setup

We run a pilot study to evaluate the performance of Postureware. We have 15 people to be our experiment subjects, including 13 males and 2 females. The weight of those subjects ranges from 45 kg to 85 kg, while the height is between 150 cm to 180 cm. The diversity of subjects in terms of sex, weight and height makes the dataset representative and enables the system to have good generalizability.

All subjects are instructed to perform fifteen categories of sitting postures. As discussed previously, people sit differently even for the same posture. Then subjects will perform the predefined fifteen sitting postures discussed in section II. Note that the subjects are intervened to conduct predefined set of postures, because we find that some of users can not perform some postures accurately such as sitting upright. We argue

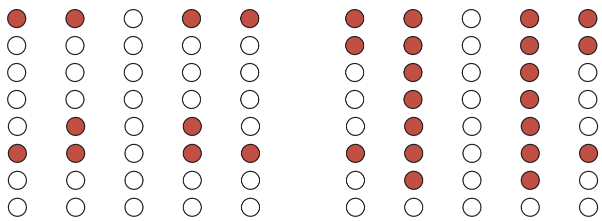

Fig. 3: Sensor placement solution: red dots represent the deployed sensors: (a) sensor number equals 10, (b) sensor number equals 20

that intervention is necessary when instructing user to hold some postures, mainly because subjects may not know what a standard healthy posture looks like.

In order to measure the performance, we use three widely used evaluation metrics: precision, recall and F-measure. Let us consider a sample of activity A_1 in the test dataset. If the predicted activity is A_1, it will be counted as a true positive (TP). Otherwise, assume the predicted activity is A_2, then it would be counted as a false positive (FP). In addition, activity A_1 will be counted as a false negative (FN) since it is missing in the prediction. F-measure reflects the overall effect of both precision and recall. The metrics can be computed based on the following equations.

To evaluate the system performance, we use 10-fold cross validation to measure the performances of classification models. In order to demonstrate the effectiveness of the proposed methods, we compare the performance of proposed methods against a set of benchmarks. The benchmarks of sitting posture recognition includes the widely adopted classifier approaches such as Naive Bayes, Support Vector Machine [14]. Those classifiers are implemented in open source machine learning software Weka 3.6 [15].

C. System Performance

This section presents the evaluation results of sensor placement solution, feature selection, sitting posture recognition and the system real-timeness.

1) Sensor Placement: We are interested in figuring out the effective solutions for sensor placement. We first deploy forty pressures sensors in 5*8 gird and then run the proposed sensor placement algorithm to select a number of sensors. Figure 3 shows the sensor placement solutions given different number of sensors. Specifically, we plot the sensor placement results when number of deployed sensors equals ten and twenty. Interestingly, with the more sensors deployed, two areas will become denser: one is close to the front boundary of a chair, while the other is on the region around the center of the chair. The results are reasonable because the pressure distribution in front boundary of chair provide more distinctive information to differentiate postures like left legs crossed in knees and left legs crossed in ankle. On the other hand, the region around the center of chair is mainly covered by the hip, offering distinctive information for different sitting postures.

We also want to evaluate the impact of sensor number towards the recognition accuracy. With 10 sensors placed using the proposed method, it can achieve 99.6% ten-fold cross validation accuracy for 15 sitting postures. Furthermore, increasing sensor number significantly improve the recognition accuracy when the sensor number is less than 5.

We also compare the proposed method and the uniform method. The uniform sensor placement solution is obtained based on the uniformly generated random value ranging from 1 to 40, which indicates the deployed location. Specifically, when the sensor number is eight, the proposed solution achieves 97% accuracy, while the uniform solution obtains 90%. The key takeaway from the results is as follows: when the number of deployed sensors is small, the proposed method achieve much higher accuracy than the uniform method. However, the gap between the proposed method and the uniform method narrows as the number of sensors increase. The intuition behind this observation is that the more sensors are deployed, the more both solutions overlap in terms of sensors.

2) Feature Selection: The objectives of feature selection experiment are two-folds: first objective is to rank the features in terms of the information gain metric; second objective is to evaluate the impact of feature number towards recognition accuracy. Based on information gain method, the gravity feature is considered as the most important feature out of 65 features. Interestingly, the rest top five informative features all belong to ratio-based features, indicating the pressure ratio between the left side and right side of the chair. It is because that different sitting postures essentially results in the pressure redistribution on the chair top, which can be reflected by gravity feature and ratio-based features.

We will further evaluate the impact of feature number to the recognition accuracy. Figure 4 plots the recognition accuracy with respect to number of features when sensor number is 15. Note that the recognition accuracy generally improves when the feature number increases. Specifically, when the number of features is 5, the proposed method can achieve 91% ten-fold cross validation accuracy. The high recognition accuracy with a few number of features demonstrates that the extracted features are distinctive and user invariant, which can be used to differentiate different sitting postures effectively.

Classification Algorithm	Precision	Recall	F-measure
Naive Bayes	0.59	0.54	0.56
SVM	0.88	0.86	0.86
AdaBoost(Naive Bayes)	0.61	0.56	0.56
AdaBoost(Decision stump)	0.02	0.14	0.04
AdaBoost(C4.5)	**0.98**	**0.98**	**0.98**

TABLE II: Sitting posture recognition accuracy (sensor = 10)

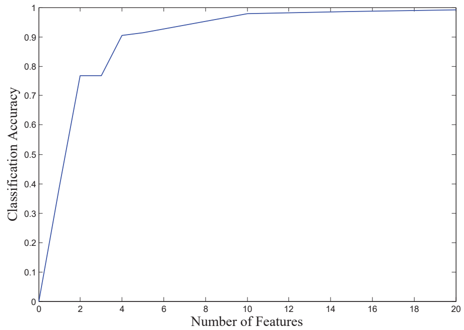

Fig. 4: The impact of feature dimension on classification accuracy

3) Sitting posture recognition result: The main objective of the this experiment is to evaluate the performance of the proposed sitting posture classification algorithm. Support vector machine and naive Bayes act as the baseline methods. Table II summarizes the results when the number of deployed sensors are 10. AdaBoost with C4.5 weak classifier achieves the best performance with 98% precision accuracy and 98% recall accuracy, while AdaBoost with decision stump weak classifier has a very inaccurate performance. It is worth emphasizing that extracted features in this paper are in essence discriminative such as the relative position and the pressure ratio, which generally prefers discriminative models like support vector machine than generative model like naive Bayes.

Generalization performance is a very important metric on how well the classification model can be applied to different users. To measure the generalization performance, we use Leave-one-out cross-validation. In particular, the data samples of certain users will be saved for validation. Figure 5 plots the generalization accuracy given different sensor number. When the sensor number is larger than 10, the generalization accuracy is quite high with precision larger than 85% and recall larger than 86%.

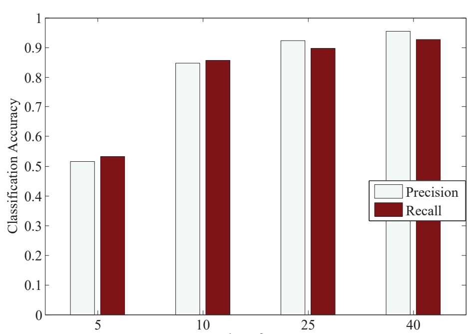

Fig. 5: Generalization accuracy

V. Conclusion

In this paper, we present Postureware, a practical solution for sitting posture recognition based on pressure sensor array. To reduce system cost, we propose an effective sensor placement solution that can achieve 98% ten-fold cross validation accuracy pressure sensors. To ensure the system generalization capability, we develop a robust sitting posture recognition framework, including user-invariant features and an ensemble learning classification model. Our system can achieve 85% generalization accuracy.

In future work, we plan to extend the sitting posture recognition system to perform inter-disciplinary researches. In particular, we are interested in three directions: affection recognition, rehabilitation and self-improvement.

Acknowledgment

The work described in this paper was fully/substantially/partially supported by the funding for Project of Strategic Importance provided by The Hong Kong Polytechnic University (Project Code : 1-ZE26). The paper is also financially supported in part under NSFC/RGC Joint Research Scheme with grant No. N_PolyU519/12 and NSFC Key Grant 61332004.

References

[1] Wikipedia, "Poor posture," http://en.wikipedia.org/wiki/Poor posture, July 2013.
[2] K. J. Sandin and B. S. Smith, "The measure of balance in sitting in stroke rehabilitation prognosis." *Stroke*, vol. 21, no. 1, pp. 82–86, 1990.
[3] L. E. Dunne, P. Walsh, S. Hermann, B. Smyth, and B. Caulfield, "Wearable monitoring of seated spinal posture," *Biomedical Circuits and Systems, IEEE Transactions on*, vol. 2, no. 2, pp. 97–105, 2008.
[4] J. F. Knight, H. W. Bristow, S. Anastopoulou, C. Baber, A. Schwirtz, and T. N. Arvanitis, "Uses of accelerometer data collected from a wearable system," *Personal and Ubiquitous Computing*, vol. 11, no. 2, pp. 117–132, 2007.
[5] G. Alexey, "Depth camera," http://www.codeproject.com/Articles/260741/Sitting-posture-recognition-with-Kinnect-sensor, Oct. 2011.
[6] B. Boulay, F. Brémond, and M. Thonnat, "Posture recognition with a 3d human model," in *Imaging for Crime Detection and Prevention, 2005. ICDP 2005. The IEE International Symposium on*. IET, 2005, pp. 135–138.
[7] W. Xu, Z. Li, M.-C. Huang, N. Amini, and M. Sarrafzadeh, "ecushion: An etextile device for sitting posture monitoring," in *Body Sensor Networks (BSN), 2011 International Conference on*, 2011, pp. 194–199.
[8] H. Z. Tan, L. A. Slivovsky, and A. Pentland, "A sensing chair using pressure distribution sensors," *Mechatronics, IEEE/ASME Transactions on*, vol. 6, no. 3, pp. 261–268, 2001.
[9] Tekscan. Flexiforce. http://www.tekscan.com/flexiforce.html.
[10] B. Mutlu, A. Krause, J. Forlizzi, C. Guestrin, and J. Hodgins, "Robust, low-cost, non-intrusive sensing and recognition of seated postures," in *Proceedings of the 20th annual ACM symposium on User interface software and technology*. ACM, 2007, pp. 149–158.
[11] Y. Li and R. Aissaoui, "Smart sensor, smart chair, can it predicts your sitting posture?" in *Industrial Electronics, 2006 IEEE International Symposium on*, vol. 4, 2006, pp. 2754–2759.
[12] L. C. Molina, L. Belanche, and À. Nebot, "Feature selection algorithms: A survey and experimental evaluation," in *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on*. IEEE, 2002, pp. 306–313.
[13] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," 1996.
[14] C. M. Bishop *et al.*, *Pattern recognition and machine learning*. springer New York, 2006, vol. 1.
[15] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *ACM SIGKDD explorations newsletter*, vol. 11, no. 1, pp. 10–18, 2009.