Computer Vision and Image Understanding 000 (2016) 1–14

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

Improving posture classification accuracy for depth sensor-based human activity monitoring in smart environments

Edmond S.L. Ho[a,c,*], Jacky C.P. Chan[a], Donald C.K. Chan[a], Hubert P.H. Shum[b], Yiu-ming Cheung[a], Pong C. Yuen[a]

[a] Department of Computer Science, Hong Kong Baptist University, Hong Kong SAR
[b] Faculty of Engineering and Environment, Northumbria University, Newcastle upon Tyne, UK
[c] Science Faculty, HKBU Institute of Research and Continuing Education, Shenzhen, Guangdong, China

ARTICLE INFO

Article history:
Received 17 April 2015
Accepted 29 December 2015
Available online xxx

Keywords:
Smart environments
Monitoring systems
Posture classification
Max-margin classification
Depth camera
Reliability estimation

ABSTRACT

Smart environments and monitoring systems are popular research areas nowadays due to its potential to enhance the quality of life. Applications such as human behavior analysis and workspace ergonomics monitoring are automated, thereby improving well-being of individuals with minimal running cost. The central problem of smart environments is to understand what the user is doing in order to provide the appropriate support. While it is difficult to obtain information of full body movement in the past, depth camera based motion sensing technology such as Kinect has made it possible to obtain 3D posture without complex setup. This has fused a large number of research projects to apply Kinect in smart environments. The common bottleneck of these researches is the high amount of errors in the detected joint positions, which would result in inaccurate analysis and false alarms. In this paper, we propose a framework that accurately classifies the nature of the 3D postures obtained by Kinect using a max-margin classifier. Different from previous work in the area, we integrate the information about the reliability of the tracked joints in order to enhance the accuracy and robustness of our framework. As a result, apart from general classifying activity of different movement context, our proposed method can classify the subtle differences between correctly performed and incorrectly performed movement in the same context. We demonstrate how our framework can be applied to evaluate the user's posture and identify the postures that may result in musculoskeletal disorders. Such a system can be used in workplace such as offices and factories to reduce risk of injury. Experimental results have shown that our method consistently outperforms existing algorithms in both activity classification and posture healthiness classification. Due to the low cost and the easy deployment process of depth camera based motion sensors, our framework can be applied widely in home and office to facilitate smart environments.

1. Introduction

One of the main purposes of smart environments and monitoring systems is to enhance the quality of life. On one hand, by understanding the needs and intention of the user, smart systems can provide the appropriate support. On the other hand, by monitoring the movement behavior of the user, these systems can alert the user in dangerous situations, such as performing movement that would result in injury. In particular, according to the Health and Safety Executive Annual Statistics Report for Great Britain [1], more than 1.1 million cases of work-related ill health were reported between 2011 and 2012, in which more than 39% belongs to musculoskeletal disorders. A smart environment with an automatic posture monitoring system is a potential solution to save the high cost of workplace injury and ill health.

One major challenge of a smart environment is to understand what the user is doing, in order to decide how to react properly to the user's behavior. Motion capturing is a traditional method to obtain the user's posture [2]. However, most of the existing techniques such as the optical motion capturing system require careful setup and calibration. These systems usually require the user to wear special devices on the body, making it difficult to be deployed and used in daily life environments. Alternatively, identifying human posture with traditional 2D video cameras can be performed using computer vision techniques [3]. However, because of the lack of details in the source video, as well as the 3D information of joints, only bigger limbs such as the body trunk and the

* Corresponding author.
E-mail address: edmond@comp.hkbu.edu.hk (E.S.L. Ho).

http://dx.doi.org/10.1016/j.cviu.2015.12.011

legs can be identified and evaluated. This greatly reduces the accuracy especially for evaluating subtle posture differences.

Recently, motion sensor with depth camera such as the Microsoft Kinect has shown its effectiveness in tracking 3D human posture in real-time [4]. Its advantage is that it can track 3D human posture without requiring the user to wear any special equipment. The low cost of the hardware camera, as well as the easy setup of the tracking system, also make it preferable to be used in daily indoor environment such as office and home. By processing the captured depth image, it becomes possible to identify depth-based edge extraction and ridge data, which are used to track human body parts [5]. However, unsupervised approaches require careful algorithm design and may not be easily generalized. To solve the problem, anatomical landmarks trained by sample data using random forests are used. The body skeleton is recognized by analyzing the depth silhouettes of the user and locating the anatomical landmarks [6]. However, run-time detection of such landmarks is not always accurate, which results in degrading the activity recognition accuracy. Similarly, utilizing the skeleton recognized by Kinect for action recognition suffer from the same problem, as the recognized joint can be different from the trained data due to occlusions, which results in noisy skeletons [7]. Previous motion analysis algorithms that assume a reliable input stream do not work well with Kinect, as the tracked joints returned by the depth camera could be wrong [8]. The main focus of this work is to propose new methods to account for the accuracy of the skeleton, such that activity recognition can be more accurate.

We propose a new posture classification framework for Kinect, which has an improved accuracy over previous algorithms. To cope with the noisy input posture, we design a set of reliability measurement [9] to evaluate how reliable the tracked joints are. The more reliable joints then contribute more in a max-margin classification system, which is used to classify postures of different context. Our framework allows a smart environment to understand what the user is doing from the noisy data obtained by Kinect. Due to the improved accuracy, the system can even classify the subtle difference between healthily and unhealthily performed postures, such as operating equipment with postures that may lead to injury. This facilitates automatic posture monitoring for workplace, which can alert the user whenever an unhealthy posture is performed. Since our method is robust, affordable and easily deployable, it is a preferable solution for smart environments and monitoring systems.

To facilitate further research in the field, the posture healthiness database created in this research will be made available to the public. Up to now, such a kind of database is not openly available. The comprehensive database consists of more than 8000 3D postures for different behaviors such as working at an office desk in sitting and standing postures, together with the source 3D depth images and color images obtained from the depth camera. It is also carefully annotated with information of the behavior, such as the nature of the movement and the potential health risks.

1.1. Contributions

There are three major contributions in this paper:

- We propose a new framework to monitor and classify user postures. It evaluates the reliability of the observed joints from Kinect, and applying such reliability as weights in a customized max-margin classifier to robustly classify noisy posture data. Our system can accurately distinguish the subtle differences between healthy and unhealthy postures.
- We propose a set of new reliability measurement terms on top of those presented in [9] to enhance the accuracy of joint reliability estimation. Apart from the traditional kinematic-based reliability measurements, we make use of the color and depth images from Kinect to identify joint that are wrongly tracked or corrupted by noise.
- We implement the first open access motion database targeting at posture healthiness. The database includes correctly and incorrectly performed postures for different work purposes, annotated posture information, as well as depth and color images obtained from the depth camera.

1.2. Outline

In the rest of this paper, we will first review the related work in Section 2. An overview of our proposed method will be given in Section 3. Next, we explain how to evaluate the reliability of each tracked joint by our proposed reliability measurements (Section 4). A max-margin classification framework which takes into account the reliability of each joint will be introduced in Section 5. We then explain how our motion database is constructed (Section 6) and present experimental results in Section 7. Finally, we conclude this paper in Section 8.

2. Related work

In this section, we review how human motion is obtained using traditional methods, and point out why these methods cannot be applied efficiently for smart environments. We also review depth camera based systems for motion tracking, and describe their weakness on noise control. We finally review works that evaluate posture based on the motion capture input, focusing the discussion on how they perform with depth cameras.

2.1. Wearable activity recognition

In computer animations and games, 3D human postures are usually captured using wearable motion capture systems. Lara and Labrador [10] provide a comprehensive survey on using wearable sensors for activity recognition. In a smart environment, wearable sensors can provide information to log the emotional status of the user [11]. Using different streams from smartphone such as audio and accelerometer can identify different activities for the purpose of life logging [12].

Different wearable systems come with different strengths and weaknesses. The optical motion capturer gather the user's 3D posture using a set of reflective markers attached on the user's body [2]. However, successful captures require the markers to be visible by the cameras, which is difficult when the user is partly occluded by surrounding objects. The accelerometer-based [13,14] and the magnetic-based [15] motion capturers overcome this constraint. By applying linear discriminant analysis (LDA) on a training action database, one can recognize the contextual meaning of the captured action using signals from accelerometers and gyroscopes [16]. By introducing audio signals captured from microphones on top of accelerometers, the action recognition accuracy can be improved [17].

Nevertheless, in these systems, the user has to wear the sensors and the system requires careful calibration before actual usage, which is not suitable for autonomous motion monitoring. On the other hand, video-based activity recognition serves as an alternative that utilizes an easier setup process, which will be reviewed in next section.

2.2. Video activity recognition

Traditional video activity recognition is performed by analyzing 2D color images captured by video cameras and identifying moving objects [18]. By tracking the non-deformable parts of a human body, 2D human postures in the video can be recognized [19]. It

Please cite this article as: E.S.L. Ho et al., Improving posture classification accuracy for depth sensor-based human activity monitoring in smart environments, Computer Vision and Image Understanding (2016), http://dx.doi.org/10.1016/j.cviu.2015.12.011

is then possible to gather high level information such as human–object interaction [20] and scene geometry [21]. The problem of these color image based algorithms is the relatively low precision for smaller body parts and the lack of 3D support, making them unsuitable for analyzing the fine details of complex human movement.

Depth camera based motion tracking system such as the Microsoft Kinect has become popular in recent years. It obtains a depth image using structured infrared light. Human posture can then be tracked by training a decision tree using a depth image database to identify different human joints [22,23]. Another class of tracking technique is to fit a skeleton structure into the detected human point cloud [24,25]. Using depth camera, tracking can be performed without requiring the user to wear any equipment, which is by definition a natural user interface to capture human motion in real-time [26].

Apart from tracking body postures, a popular research direction is to apply depth cameras to identify high level activities using different features such as 3D point cloud with relative location descriptors [27] and depth silhouettes [28,29]. To enhance recognition accuracy, skin joint features that use body skin color to identify human body parts are suggested [30]. Shape features with movement information that are represented and silhouette history information with silhouettes motion variation data are also proposed [31]. Hybrid features that combines different features including tracked joint movement and surface shape take advantage on the diversity of features to improve the system performance [32]. Utilizing translation and scaling invariant features can enhances the robustness of the activity recognition system [33]. To better handle occlusions between joints, rigid body parts features that consist of binary edge extraction and ridge data are used [5].

Utilizing Kinect in smart environments is a popular research topic. It can be applied in smart home to monitor older people and detect when they are likely to fall [34], to log daily activities [35–37], and to monitor residents [29]. It is also applied in smart office to evaluate the seating postures [38,39]. In the area of ergonomic, Kinect can be used for evaluating if lifting and carrying motion is detrimental to the health of workers [40]. Kinect is also applied in rehabilitation monitoring [41] and physiotherapy [42]. It is found to be suitable to assess rehabilitation performance if the error bounds are set [41]. While these researches attempt to utilize Kinect in smart environments, they do not formally handle the noisy input problem. It is pointed out that using Kinect for surveillance or monitoring applications would usually require mounting the device in high positions, which further degrades the tracking performance [43]. In this work, we propose a framework to deal with the noisy data for more accurate motion classification.

2.3. Posture evaluation

Posture evaluation is the process to understand the nature of a given posture. While geometric rules can be defined to evaluate a posture [44] and thereby to classify it [45], the rules have to be manually crafted in order to obtain the best system performance. The domain of the rules also need to be selected based on the nature of the postures to represent the posture context efficiently [46], making it inefficient to be extended to a wide variety of movement.

Data-driven approaches overcome the difficulty by evaluating the postures with prior knowledge obtained from a posture database [47]. Traditional data-driven algorithms usually assume a consistent [48] or reliable input signal [8] in order to evaluate the posture with respect to the database. However, the movement tracked by a depth camera is highly noisy due to occlusion and mis-tracking. In order to apply data-driven algorithms on depth camera based systems, it is important to assess the reliability of the input signal to identify the noise [9]. In this work, we adapt the kinematic-based reliability measurements from [9] and propose new terms utilizing the color and depth images, which enhances the overall system accuracy.

A naive method to classify an observed posture using data-driven approaches is to find a best match in the posture database [4]. However, the result will easily be affected by outliers in the database. A better approach is to search for the K nearest neighbors and do the classification based on the set of retrieved postures [49]. To avoid the high run-time cost for searching neighbors, Gaussian Process can be used to produce an abstract representation of the posture space [50].

In this work, we propose a new data-driven framework to classify Kinect postures. It includes a max-margin classification system that takes into account the reliability of the input data. Different from [9], which applies reliability measurements with a lazy learning algorithm to reconstruct the observed posture, this work utilizes the reliability measurements to enhance posture classification accuracy from noisy input data.

3. Overview

Fig. 1 shows the overview of our proposed system. Since the posture from Kinect is noisy and inaccurate, we introduce a set of reliability measurement to evaluate the reliability of the captured joints (Section 4). The reliability measurement is computed according to the consistency of the (1) joint displacement, (2) bone-length, image pixels around the joint in (3) RGB image, and (4) depth image over consecutive frames. Such reliability estimations are then integrated with the captured posture data into a max-margin classifier for posture classification (Section 5). Our proposed classification framework will learn the weighting for each reliability term to maximize the discriminative power of the classifier. During run-time, we monitor and analyze the user's posture in real time by computing the reliability measurements from the captured pose and classify it using our proposed max-margin classification framework. Depending on the application, our system can be used to classify different types of movement, or even

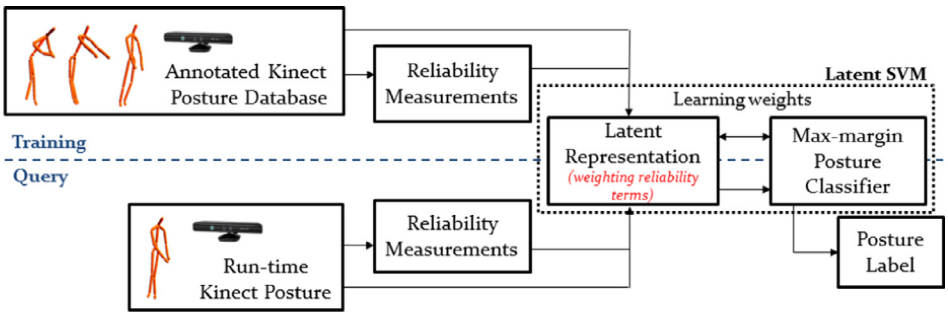


Fig. 1. The overview of our proposed framework for robust posture classification.

the healthiness status of a posture. Finally, we collect annotated human motion data using Kinect and create a motion database (Section 6) for training the classifier.

4. Reliability measurement

While Kinect can capture 3D skeletal information in real-time, the tracked human motion data are too noisy to be used in serious applications such as health monitoring systems. Therefore, it is necessary to identify the unreliable joints in order to improve the classification accuracy.

The reliability of the source data can be measured by a set of heuristics. On top of the existing behavior and kinematics reliability terms that evaluate the movement behavior and the segment length of the skeleton, respectively [9], we design two new terms that utilize the color and depth image to evaluate extra features.

4.1. Behavior reliability term

The behavior reliability term evaluates abnormal behavior of a tracked part, which is defined based on the amount of high frequency vibration of the detected joint position.

Kinect detects the user posture with the acquired depth image. The position of a joint is determined based on the depth pixels that are classified to it using a decision tree based algorithm [22]. As a result, when some joints are occluded, or when they are incorrectly recognized, the detected positions of the parts become unstable due to the lack of expected features. By evaluating the high frequency vibration of the tracked joints, we can model their respective reliabilities.

Assuming $p_i(f)$, $p_i(f+1)$ and $p_i(f+2)$ to be the 3D position of a tracked joint i in three successive frames, we can calculate the displacement vectors of the joint in frame f and $f+1$ as:

$$d_i(f) = p_i(f+1) - p_i(f) \tag{1}$$

$$d_i(f+1) = p_i(f+2) - p_i(f+1) \tag{2}$$

Since human movements are smooth in nature, the displacement vectors of a joint over consecutive frames should be similar and consistent. The inconsistency between the displacement vectors of a joint will result in high frequency of vibration and it can be evaluated by the acute angle calculated by the dot product between the two displacement vectors in consecutive frames:

$$\theta_i(j) = \begin{cases} \arccos\left(\dfrac{d_i(f)\cdot d_i(f+1)}{||d_i(f)||||d_i(f+1)||}\right) & \text{if } ||d_i(f)|| > d_{\min} \text{ and } ||d_i(f+1)|| > d_{\min} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where $d_{\min}$ is the minimum length of an acceptable displacement vector, and is set to 3 cm in our experiment. It is used to avoid getting a large angle change when the joint position is almost steady.

The behavior term is defined as:

$$Rb_i(f) = 1 - \frac{\max\left(\min\left(\frac{\sum_{f=0}^{f_b}\theta_i(f)}{f_b}, \theta_{\text{roof}}\right) - \theta_{\text{floor}}, 0\right)}{\theta_{\text{roof}} - \theta_{\text{floor}}} \tag{4}$$

where $Rb_i(f) \in [0.0, 1.0]$, f_b is the total number of frames we consider to detect vibration, θ_{floor} is an acceptable amount of rotation for each frame, θ_{roof} is the amount of rotation we consider to be the most unacceptable. Empirically, we found that setting $f_b = 3$, $\theta_{\text{floor}} = 90^\circ$, and $\theta_{\text{roof}} = 135^\circ$ gives a good result.

Notice that Kinect works best when the user is 6 feet away from the camera and is facing directly to it. In many workspace environments, it is impossible to have such a setup due to the limitation of space. We found that the postures obtained by Kinect when the camera is too far/close, or shooting the user in an angle, usually result in a higher level of noise. The behavior term described in this section can detect such noise to enhance the usability of the system.

4.2. Kinematics reliability term

The kinematics term evaluates the reliability of joints based on their kinematics correctness, which is defined with the consistency of segment length.

Kinect recognizes joints individually when determining their position, and does not explicitly maintain the kinematic correctness of the resultant postures. As suggested in [51], the length of each body limb needs to be constant over time during a real human movement. Therefore, when the position of a joint is incorrectly determined, the corresponding segment length will be changed. Here, we evaluate the reliability of a joint based on the corresponding segment length difference with respect to the reference value.

A pose initialize process is usually required to obtain reference values of body dimensions [5,52]. In [9], the reference segment length is obtained by requesting the user to perform predefined postures, such as a T-pose, in order to accurately recognize all joints. However, for anonymous tracking, it is impossible to ask individual user for initializing the system. Also, because of the space limitation, the depth camera may be setup to look at the user in an angle, making it difficult to accurately obtain the positions of all joints. Here, inspired by Jalal et al. [52] in which torso area is initialized using left and right extremes values, we propose to utilize the distance between the left and right shoulder joints detected by Kinect to estimate the body segment length, as the shoulders can be tracked accurately in a wide range of shooting angles. Based on the shoulder width, we evaluate the length of other segments with the segment length proportion described in [53].

In each pose, a joint can connect to multiple segments depending on the skeleton structure, such as the hips connecting to three segments. Assuming the joint i is connected to $s_{\text{part\_total}}$ body segments, for each connecting segment s, the segment difference ratio at frame f is calculated as:

$$d_s(f) = \min\left(\frac{\text{abs}(l_s(f) - l_{\text{s\_ref}})}{l_{\text{s\_ref}}}, 1\right) \tag{5}$$

where $l_{\text{s\_ref}}$ is the reference segment length and $l_s(f)$ is the current segment length for segment s at frame f.

The kinematics reliability value of a joint is defined as the mean segment different ratio for all connecting segments:

$$Rk_i(f) = 1 - \frac{\sum_{s=1}^{s_{\text{part\_total}}} d_s(f)}{s_{\text{part\_total}}} \tag{6}$$

where $Rk_i(f) \in [0.0, 1.0]$. The whole kinematic terms calculation process is summarized in Algorithm 1.

4.3. Color image reliability term

The color image term evaluates the reliability of joints based on their closeness of gradient features between two adjacent frames in the RGB color video.

Since human movements are continues in nature, the appearance of the joints in adjacent frames as shown in the color video should be visually similar. Dissimilar joint appearance across frames usually indicates mis-tracked joint in at least one of the frames. In our system, the color image reliability of a joint is computed by extracting a square patch of pixels centered at the joint from the color image, and evaluate the difference in color across frames. We convert the RGB pixel into gradient representation to

Please cite this article as: E.S.L. Ho et al., Improving posture classification accuracy for depth sensor-based human activity monitoring in smart environments, Computer Vision and Image Understanding (2016), http://dx.doi.org/10.1016/j.cviu.2015.12.011

Algorithm 1 Computing the kinematics reliability term.

1: Given a data set D which contains skeletal data, the kinematics reliability values associated with each joint are extracted from each frame (Section 4.2)
2: **for** `each body segment` **do**
3: estimate reference body segment length based on the shoulder width
4: **end for**
5: **for** `each joint` **do**
6: **for** `each connecting body segment` **do**
7: compute the segment difference ratio (Eq. (5))
8: **end for**
9: compute the kinematics reliability value as the mean segment difference ratio of all connecting segments (Eq. (5))
10: **end for**

isolate color changes from lighting condition differences. We also quantize the computed gradient into eight bins to avoid the effect of small color difference error. Example frames are shown in Fig. 2, in which the left elbow and left wrist are not correctly tracked in the middle column.

For each tracked joint i at frame f, the color patch is represented by a vector

$$cpatch_{i,f} = [g_1, g_2, ..., g_{\text{patch\_size}}] \tag{7}$$

which concatenate the binned gradient g_1 to $g_{\text{patch\_size}}$ computed from each pixel within the patch. The color image reliability term of joint i is calculated as the cosine distance between two corresponding patches extracted from two consecutive frames:

$$Rc_i(f) = \left(1 - \frac{cpatch_{i,f} \cdot cpatch_{i,f+1}}{\|cpatch_{i,f}\| \|cpatch_{i,f+1}\|}\right) \tag{8}$$

where $Rc_i(f) \in [0.0, 1.0]$, $cpatch_{i,f}$ and $cpatch_{i,f+1}$ are the patches extracted at joint i in frame f and $f+1$, respectively.

The size of the color patch is set according to the size of the skeleton in pixel with respect to the screen resolution. Under a typical setup, that is, an adult user facing the Kinect and standing 6 m away from it, a patch size of 27 by 27 pixel works very well in the resolution of 640 by 480. Such a size can be dynamically adjusted based on the camera angle and position.

4.4. Depth image reliability term

The depth image term evaluates the reliability of joints based on their closeness of gradient features between two adjacent frames in the depth image sequence.

The idea of the term is to evaluate if there is any sudden change of depth at the detected joint position across two frames, which usually indicates that the joint is mis-tracked. Similar to the color image reliability term, we extract a patch of depth image *dpatch* centered at a given joint and compare such a patch in consecutive frames. Again, the gradients are quantized into eight bins and *dpatch* is composed by concatenating the binned gradient values of the pixels within the patch. The depth image reliability term of joint i is then computed by:

$$Rd_i(f) = \left(1 - \frac{dpatch_{i,f} \cdot dpatch_{i,f+1}}{\|dpatch_{i,f}\| \|dpatch_{i,f+1}\|}\right) \tag{9}$$

where $Rd_i(f) \in [0.0, 1.0]$, $dpatch_{i,f}$ and $dpatch_{i,f+1}$ are the patches extracted at joint i in frame f and $f+1$, respectively.

The advantage of introducing the color and depth image terms on top of the behavior and kinematics terms, is enabling the system to evaluate the reliability of a joint from the raw data point of view. The major weakness of the behavior and kinematics terms is that they cannot distinguish a correct but unstable joint from a mis-tracked joint. Unstable joints contains some usable information, but mis-tracked ones as shown in Fig. 2 should not be used. The proposed color and depth image terms fill the gap by analyzing low level image-based information, in which we evaluate if a joint resembles similar features across frames. Notice that since mis-tracked joints are usually highly unstable in Kinect, the image terms only compare two consecutive frames. If the mis-tracked joints would remain at a fix position in other tracking systems, a longer time window should be considered.

5. Max-margin classification with reliability measurement

In this section, we explain our proposed posture classification algorithm that considers both the skeletal features (e.g., joint positions, relative joint positions) and the respective reliability terms. Since the reliability of the joint is taken into account, our classifier is more robust than existing methods especially for noisy data.

Fig. 2. Examples of image patches (shown in red squares) extracted around the body joints for computing the color and depth images reliability terms. Mis-tracked joints such as the left elbow (in the middle column) result in large difference in the patches. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

We adapt the max-margin learning framework [54] as our classifier as it can directly classify data in which some of the features are unavailable in each data instance. Traditional max-margin systems formulate the learning process as maximizing the worst-case instance margin in the training data. In particular, the calculation of the margin of each instance is based on the availability of the features, meaning that absent features do not contribute to the classification process. This process allows instances with incomplete features to be compared and classified directly.

The problem of applying traditional max-margin framework to our problem is that joint positions detected by Kinect may be available but incorrect due to sensor error. Furthermore, the noise level of different joints is different according to the type of the motion performed, making it difficult to applying predefined threshold to filter joint with low reliability. We therefore formulate the instance margin calculation as a feature weighting process according to the corresponding reliability measurement. This enables the system to determine the importance of a joint based on its reliability in order to achieve high system robustness.

Here, we first review the max-margin classification framework for data with absent features [54] in Section 5.1. We then point out how we adapt it to classify data with different reliability in Section 5.2. Finally, due to the reliability measurements we introduced, our max-margin framework has more system parameters than existing ones. We explain how we design a solver that solves the system effectively in Section 5.3.

5.1. Max-margin classification with absent features

Classifying data with absent features with a max-margin framework [54] is based on a classical support vector machine (SVM) approach [55]:

$$\begin{aligned} \min_{w,\xi,b} \quad & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i \\ \text{subject to} \quad & y_i\left(wx_i + b\right) \geq 1 - \xi_i, \quad i = 1 \ldots n \end{aligned} \tag{10}$$

where x_i and y_i are the features and label of instance i, C is the tradeoff parameter between model complexity and accuracy, b is a threshold and ξ are slack variables for handling training instances that are linearly non-separable. In particular, w is learned by maximizing the margin $\rho \equiv \min_i y_i(wx_i + b)/\|w\|$.

When handling instances with missing features, however, the whole feature vector x_i will contribute to the margin calculation in the classifier training process without ignoring the absent features (usually the missing features will be replaced by predicted values or simply zeros). As a result, the performance of the learned classifier will be degraded. In order to classify data with absent features, Chechik et al. [54] treat each instance in its own subspace of the full feature space by calculating the *instance margin* $\rho(i)$:

$$\rho(i) = \frac{y_i w^{(i)} x_i}{||w^{(i)}||} \tag{11}$$

where $w^{(i)}$ contains a subset of entries in w that are correspond to the valid (i.e., non-absent) features in x_i. The geometric margin of the classifier is represented by the minimum instance margin:

$$\max_w \left(\min_i \frac{y_i w^{(i)} x_i}{||w^{(i)}||} \right) \tag{12}$$

The readers are referred to [54] for further details.

An important design in Eq. (12) is that the score (i.e., $y_i w^{(i)} x_i$) is normalized according to the availability of features (i.e., $\|w^{(i)}\|$) of the instance, allowing the system to classify instances with incomplete features. The equation implicitly increases the weight of the present features, and absent features would not contribute to the margin calculation.

5.2. Max-margin classification with reliability measurement

Here, we exploit the feature weighting design of traditional max-margin classifier such that it can be adapted to features of different reliability. We formulate our classifier learning problem as maximizing the discriminative power by weighting the features according to the reliability measurements.

In our framework, the vector of weight t_i has the same dimension with the feature vector in an instance i (i.e., a posture), $t_{i,j}$ is the weight of a skeletal feature j and it is calculated as a weighted sum of the corresponding reliability measurements:

$$t_{i,j} = \alpha_{b,i,j} Rb_{i,j} + \alpha_{k,i,j} Rk_{i,j} + \alpha_{c,i,j} Rc_{i,j} + \alpha_{d,i,j} Rd_{i,j} \tag{13}$$

where $Rb_{i,j}$, $Rk_{i,j}$, $Rc_{i,j}$, $Rd_{i,j}$ are the reliability values of feature j in instance i, and α is vector contains the coefficients of the reliability terms. Using a single value to represent the weight allows an efficient coupling of weights and features. Here, we learn a set of α for each sample when training a classifier.

The instance margin is then calculated as:

$$y_i w \frac{t_i}{\|t_i\|} x_i \tag{14}$$

in which the weight vector t_i is normalized by $\|t_i\|$. As a result, features with higher reliability values contribute more in the instance margin calculation.

Finally, the classifier can be learned by maximizing the discriminative power of the max-margin classifier to separate two different classes:

$$\begin{aligned} \max_{w,\alpha,b} \quad & \frac{1}{\|w\|} \\ \text{subject to} \quad & y_i\left(w\frac{t_i}{\|t_i\|}x_i + b\right) \geq 1, \\ & t_{i,j} = \alpha_{b,i,j} Rb_{i,j} + \alpha_{k,i,j} Rk_{i,j} + \alpha_{c,i,j} Rc_{i,j} + \alpha_{d,i,j} Rd_{i,j}, \\ & 0 \leq \alpha_{\{b,k,c,d\},i,j} \leq 1, \alpha_{\{b,k,c,d\},i,j} \in \alpha, \\ & 0 \leq t_{i,j} \leq 1. \end{aligned} \tag{15}$$

where t_i contains the reliability measurements of instance i. The objective function in Eq. (15) is equivalent to minimizing $\|w\|^2$ without the slack variables.

With the solved values of the support vector w and the coefficient vector α, the label of an instance can be predicted by computing the sign of the decision score using:

$$\text{sign}\left(w\frac{t_i}{\|t_i\|}x_i + b\right) \tag{16}$$

The classifier explained above is a binary classifier. For multiclass classification, the framework learns multiple binary classifiers and select the predicted label with highest score as the final results.

5.3. Max-margin solver

Given the max-margin classification with reliability measurement formulated in Section 5.2, both w and α need to be optimized. However, finding the global optimum is a hard problem since the objective function is non-convex because of the dependency of the α values on w. Here, we propose a block based optimization algorithm that iteratively optimize w and α [56] to maximize the discriminative power. To further improve the classification performance, we formulate the final representation of each instance as latent variables which will be computed when learning a max-margin classifier using Latent SVM [56]. The details of our proposed method will be given below.

Please cite this article as: E.S.L. Ho et al., Improving posture classification accuracy for depth sensor-based human activity monitoring in smart environments, Computer Vision and Image Understanding (2016), http://dx.doi.org/10.1016/j.cviu.2015.12.011

5.3.1. Model inference

Given w, our method computes a latent representation of each instance by finding α to maximize the decision score. This is done by optimizing the entries in α for each reliability measurement according to a given classifier $w = [w_1, \ldots, w_q]^T$:

$$\begin{aligned} S(w, R_i, x_i) = \max_{\alpha} \quad & w\frac{t_i}{\|t_i\|}x_i \\ \text{subject to} \quad & \alpha_{b,i,j} + \alpha_{k,i,j} + \alpha_{c,i,j} + \alpha_{d,i,j} = 1, \\ & 0 \le \alpha_{\{b,k,c,d\},i,j} \le 1, \quad \alpha_{\{b,k,c,d\},i,j} \in \alpha, \\ & i = 1 \ldots n. \end{aligned} \tag{17}$$

where R_i contains the reliability values (i.e., Rb_i, Rk_i, Rc_i and Rd_i) of instance i, t_i is calculated as in Eq. (13), and x_i contains the features of instance i. We constrain the sum of the entries in α as 1 such that t_i is the normalized weighted sum of the associated reliability measurements for each feature.

5.3.2. Learning

Having presented the calculation of latent representation of each instance, we now explain how w is obtained by our proposed max-margin classification framework. Similar to conventional SVM formulation, w is solved by:

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i \\ \text{subject to} \quad & y_i(S(w, R_i, x_i) + b) \ge 1 - \xi_i, \\ & i = 1 \ldots n, \quad 0 \le \xi_i. \end{aligned} \tag{18}$$

where ξ_i is slack variable introduced for non-separable training instances, $S(w, R_i, x_i)$ (Eq. (17)) returns the decision score of instance i by multiplying the latent representation with the given w, and C is the trade-off parameter, which is set as 1 in our experiments.

By solving Eqs. (18) and (17) alternatively, the classifier and representation (i.e., the latent variable) of each instance will be updated and the classification performance will be improved. Since w is a dependent of the latent representation, poor choice of initial conditions of α in the latent representation results in local minima. To tackle this problem, the classifier learning process will be performed several times ($maxTrainNum = 20$ in our experiments) by randomly initializing α to solve Eq. (18). The classifier that produces the minimum value will be chosen as in previous work [56]. The whole classifier learning process is summarized in Algorithm 2.

Algorithm 2 Reliability-value based max-margin classification.

1: Given the training set $\mathcal{X}$, the reliability values associated with each joint are extracted from each instance (Section 4)
2: **for** i = 1 to $maxTrainNum$ **do**
3: randomly initialize α
4: **repeat**
5: compute latent variables to represent each instance (Eq. (17))
6: train classifier w using the latent variables (Eq. (18))
7: **until** no change in w
8: **end for**
9: select the classifier w which produces the minimum value from the objective function in Eq. (18)

6. Posture database creation

In this section, we explain how our posture is represented in the database, and detail what kind of posture we have included to create the database.

6.1. Posture representation and capturing

We use the Microsoft Kinect to capture posture data for the database, as it is one of the most popular depth camera based motion sensors. The Kinect SDK [57] provides the utility to record the depth and color images, and the corresponding posture is tracked by SDK function calls. We manually annotate descriptions such as the nature of the motion and the potential risk of injury for each captured sequence.

Each posture P in the database is represented by a vector of 3D points:

$$P = [p_1, p_2, \ldots, p_n] \tag{19}$$

where p_i is the 3D location of the ith joint of the user and n is the total number of joints. Each posture is normalized by removing the global 3D translation and rotation along the vertical axis, as the nature of most postures is defined by local joint movement. Examples of the captured scene and the extracted 3D skeletal information are shown in Fig. 3.

Since the training samples are extracted from motion sequences, consecutive frames tend to be similar. We filter the database by removing similar postures base on the Euclidean distances of the 3D joint locations as explained in [9]. This allows the database to cover a wide variety of representative postures while being compact. This also unifies the density of samples in the database.

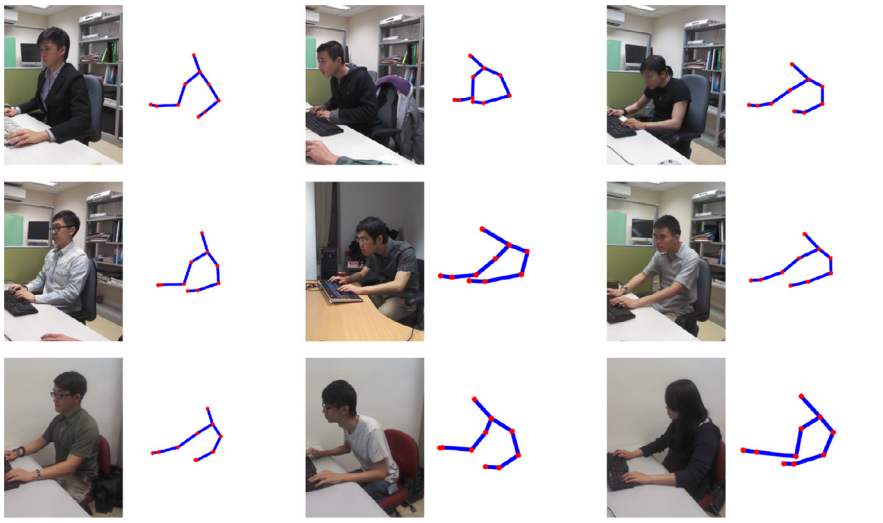
6.2. Database construction

In order to identify postures that involve health hazards, we capture both correctly and incorrectly performed postures in different working environments. We follow the guidelines produced by the European Agency for Safety and Health at Work [58] to capture movement that involves potential health risk. Both healthy and unhealthy postures of 10 participants, with ages ranged from 21 to 35, are captured. During capturing, the users are given instructions on how to perform the postures. To avoid real injury, especially when capturing unhealthy postures, extra care has been taken and the users are given time breaks during each capture. We created two databases focusing on different work environments.

The first database involves motion of standing and performing hand operations on a work bench, which is very common in field-based working environments. According to European Agency for Safety and Health at Work [58], one should prevent postures in which the joints are not in their natural position to avoid potential tendons, ligaments, and nerves damage. For a correctly performed standing posture at work, the neck should keep vertical and relaxed, the head and the back should maintain an upright position, and the shoulder should be relaxed. We follow these guidelines to capture a set of healthy postures performed by multiple people. We also design the unhealthy postures including (A-1) working on a short bench in which the user has to bend the head, neck and back, (A-2) working on a short bench that is far away from the user, and the user has to bend the back and stretch the body, (A-3) working on a work bench that is placed at the side of the user, and the user has to twist the back and raise the arms. We summarize the details of the posture classes in Table 1 to indicate the body parts are involved. The acute angles between the body part (i.e., the bone) and the vertical axis are computed from our dataset. For the torso, the angle of rotation about the vertical axis is reported. Examples of 3D pose and the corresponding RGB video are shown in Fig. 4 and different views of the standing poses are illustrated in Fig. 5.

The second database involves motion of sitting on a chair and working on a work bench, which is a usual posture for office workers. Similar to the standing posture, one should prevent bending

Please cite this article as: E.S.L. Ho et al., Improving posture classification accuracy for depth sensor-based human activity monitoring in smart environments, Computer Vision and Image Understanding (2016), http://dx.doi.org/10.1016/j.cviu.2015.12.011



(a) Healthy pose (b) B-2: The back leans forward (c) B-1: The neck leans forward

Fig. 3. Examples of postures captured in an office environment. (a) Healthy postures, (b) and (c) are considered as unhealthy postures.

Table 1
Details of the dataset of standing poses used in the experiments.

Dataset	Action class	Pose type	Body parts (angle)		
			Neck	Back	Torso
Standing	Stand straight	Healthy	Vertical (15°)	Vertical (13°)	Vertical (0°)
	(A-1) Bend back	Unhealthy	Bended (50°)	Bended (40°)	Relaxed (0°)
	(A-2) Bend and extend	Unhealthy	Relaxed (35°)	Bended (30°)	Relaxed (0°)
	(A-3) Twist body	Unhealthy	Vertical (15°)	Vertical (18°)	Twisted (15°)

Table 2
Details of the dataset of sitting poses used in the experiments.

Dataset	Action class	Pose type	Body parts (angle)	
			Neck	Back
Sitting	Straight back	Healthy	Vertical (15°)	Vertical (10°)
	(B-1) Bend neck	Unhealthy	Bended (40°)	Relaxed (15°)
	(B-2) Bend back	Unhealthy	Vertical (40°)	Bended (50°)

the head, neck and back [58]. Apart from the correctly performed postures, we capture incorrect postures including (B-1) bending the neck when working, and (B-2) bending the back when working. Since the user is in a sitting pose and is working on a work bench, the lower body is usually not visible to the depth cameras. We therefore only capture and evaluate the posture of the upper body in this database. The details are listed in Table 2. Again, the acute angles between the body part (i.e., the bone) and the vertical axis are computed from our dataset. Examples of 3D pose and the corresponding RGB video are shown in Fig. 3 and different views of the sitting poses are illustrated in Fig. 6.

7. Experimental results

In this section, we evaluate the effectiveness of our proposed method by classifying postures captured from two working environments and two benchmark datasets—MSR Action3D [59] and Florence 3D [60].

In our experiment, we trained max-margin classifiers explained in Section 5.2 to classify the postures into different classes. We carried out *leave-one-subject-out* cross validation, in which we used postures from one of the participants as testing data and all the rest postures as training data in our healthy pose datasets (Sections 7.3 and 7.4). The validation was repeated for all different combinations of the training datasets. For the benchmark datasets, we followed the data split as in the state-of-the-art approaches and the details will be given in Sections 7.5.1 and 7.5.2. Finally, we calculated the average accuracy, which is defined as the number of samples correctly classified divided by the total number of testing samples.

7.1. Datasets details

The details of the datasets used in the experiments are summarized in Table 3. To obtain a fair comparison with other approaches, we used the same data splitting (i.e., training and testing sets) among all approaches in each experiment.

For our healthy pose datasets, 20 and 10 joints are tracked in each frames for the standing and sitting datasets, respectively. For both the RGB and depth videos, the resolutions of each frame are both 640 × 480 pixels. As stated in Table 3, 10 subjects were invited to perform various kind of actions in an office environment. Their age range is 21–35 years old.

7.2. Experimental settings

To fully evaluate the performance of different parts of our framework, we design four setups as below:

Baseline classification: The baseline posture classification method does not consider the reliability of the captured 3D skeletal information, which is comparable to existing motion classification algorithms. In other words, the feature vectors is defined as the positions of all joints (i.e., joint positions) and the relative positions between every pairs of joints (i.e., relative joint positions)

Please cite this article as: E.S.L. Ho et al., Improving posture classification accuracy for depth sensor-based human activity monitoring in smart environments, Computer Vision and Image Understanding (2016), http://dx.doi.org/10.1016/j.cviu.2015.12.011

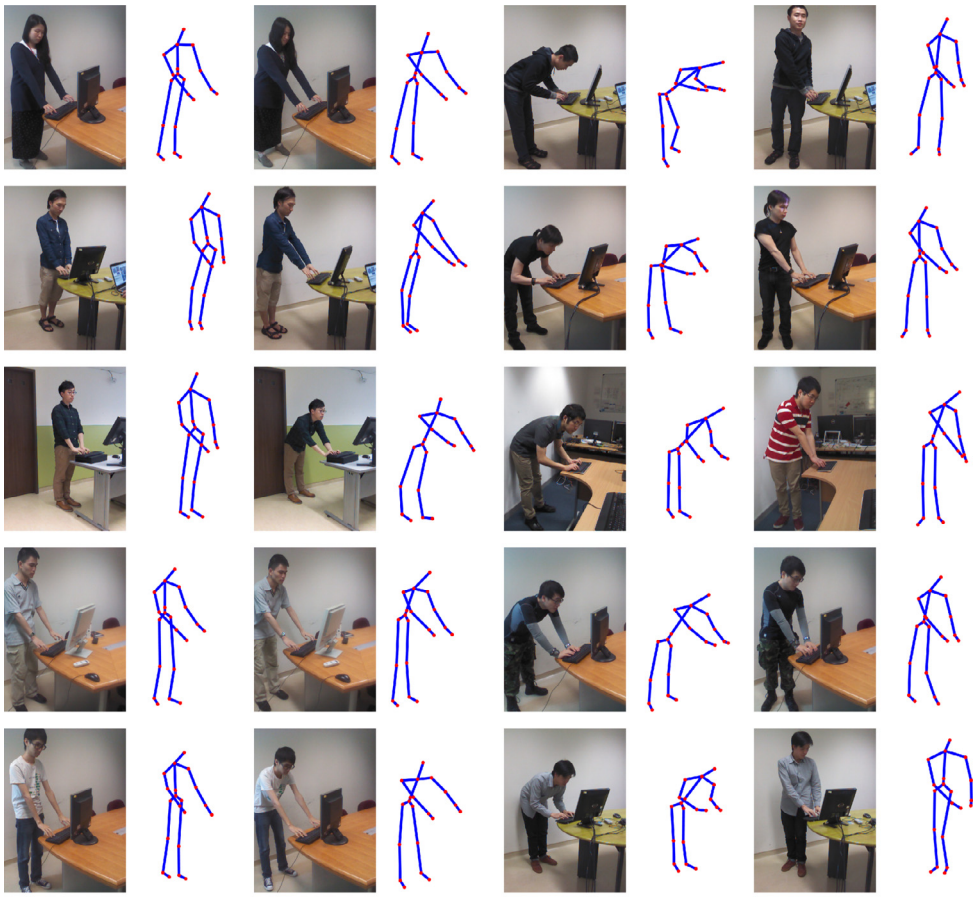


Fig. 4. Examples of postures captured in an office environment. (a) is a healthy pose, and (b)–(d) are considered as unhealthy poses.

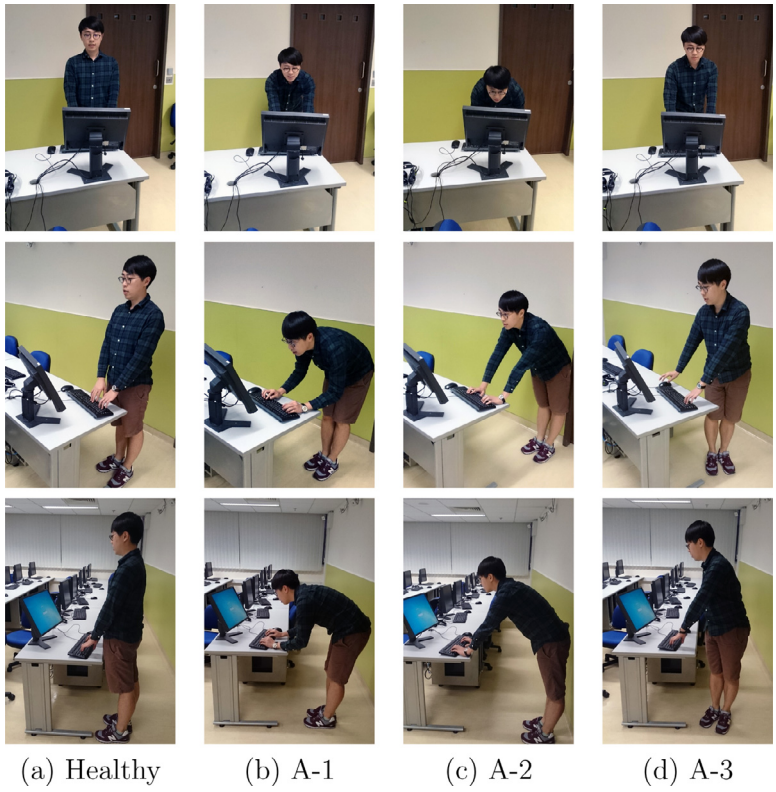
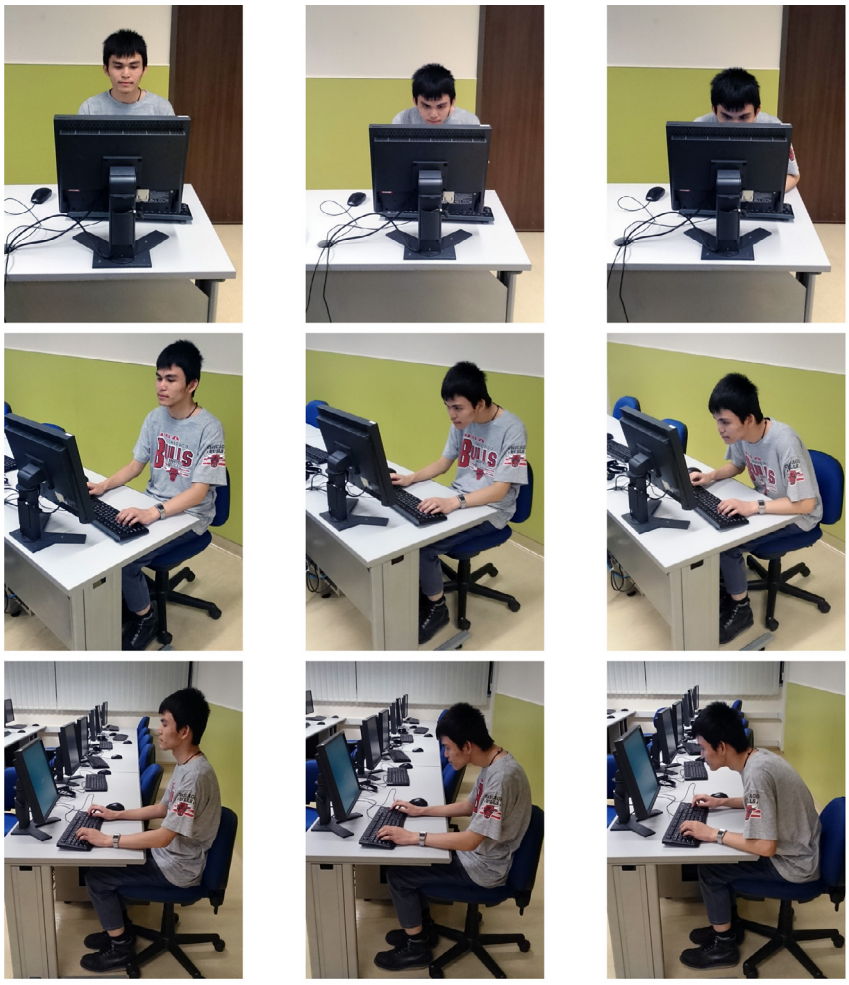


Fig. 5. Showing the captured standing poses in different view angles.

Please cite this article as: E.S.L. Ho et al., Improving posture classification accuracy for depth sensor-based human activity monitoring in smart environments, Computer Vision and Image Understanding (2016), http://dx.doi.org/10.1016/j.cviu.2015.12.011



(a) Healthy pose (b) B-1: The neck leans forward (c) B-2: The back leans forward

Fig. 6. Showing the captured sitting poses in different view angles.

Table 3
Details of all the datasets used in the experiments.

Dataset	Number of subjects	Number of classes	Size		Time duration (min) (approx.)
			Training	Testing	
Standing	10	4	1722 poses	2869 poses	6
Sitting	10	3	1621 poses	2702 poses	5
MSR Action3D [59]	10	20	284 motions	273 motions	25
Florence 3D [60]	10	9	109 motions	106 motions	4

as used in [61]. Comparing the proposed method to the baseline method can demonstrate the accuracy improvement by using reliability measurements.

Individual reliability terms classification: To show the performance of individual reliability measurement, we train the four max-margin classifiers by using the reliability term independently. The classification is performed by:

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i \\ \text{subject to} \quad & y_i\left(w\frac{R_i}{\|R_i\|}x_i + b\right) \geq 1-\xi_i, \\ & i = 1\ldots n, \quad 0 \leq \xi_i. \end{aligned} \tag{20}$$

where R_i contains one reliability term (i.e., *Rb*, *Rk*, *Rc* or *Rd*) of all features in instance *i*.

Equal weight reliability terms classification: To show the accuracy improvement of optimizing the weight for the reliability terms in Section 5.3, we setup a naive system of using all four reliability terms with the same weight:

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i \\ \text{subject to} \quad & y_i\left(w\frac{\text{Rall}_i}{\|\text{Rall}_i\|}x_i + b\right) \geq 1-\xi_i, \\ & i = 1\ldots n, \quad 0 \leq \xi_i \\ \text{where} \quad & \text{Rall}_i = 0.25Rb_i + 0.25Rk_i + 0.25Rc_i + 0.25Rd_i \end{aligned} \tag{21}$$

Variable weight reliability terms classification: Finally, we show the performance of our proposed method to find optimal

Please cite this article as: E.S.L. Ho et al., Improving posture classification accuracy for depth sensor-based human activity monitoring in smart environments, Computer Vision and Image Understanding (2016), http://dx.doi.org/10.1016/j.cviu.2015.12.011

Table 4
Details of our healthy posture datasets used in the experiments.

Dataset	Action class	Pose type	Size (poses)
Standing	Stand straight	Healthy	459
	(A-1) Bend back	Unhealthy	469
	(A-2) Bend and extend	Unhealthy	521
	(A-3) Twist body	Unhealthy	463
Sitting	Straight back	Healthy	669
	(B-1) Bend neck	Unhealthy	602
	(B-2) Bend back	Unhealthy	531

Table 5
Accuracy in classifying postures in the standing to work experiment.

Method		Average % accuracy
Joint positions		80.84
Relative joint positions (RJP) [61]		86.32
Lie group representation [62]		84.90
Moving pose [63]		81.79
Moving pose [63] with pose normalization and noise removal		81.04
Proposed	RJP with *Rb* only	85.72
	RJP with *Rk* only	86.32
	RJP with *Rc* only	86.44
	RJP with *Rd* only	85.34
	RJP with *Rb*, *Rk*, *Rc* and *Rd*—equal weight	85.61
	RJP with *Rb*, *Rk*, *Rc* and *Rd*—variable weight	**88.67**

weights for the reliability terms to improve the classification performance by alternatively solving Eqs. (18) and (17).

7.3. Standing to perform hand operations on a work bench

Here, we perform *leave-one-subject-out* classification on our standing to work motion database, which includes healthy, A-1, A-2, and A-3 postures as explained in Section 6.2. Example postures are shown in Fig. 4 and details of the data used in the experiment can be found in Table 4. On average, 1722 and 2869 postures were used as training and testing data in each classification trial. The feature vector size of the joint position and relative joint position features are 60-d and 570-d, respectively. The average classification accuracies are shown in Table 5.

According to the results:

- The variable weight classifier with RJP features outperforms the classifier with the RJP feature by 2.35%. This shows that the use of reliability measurements can enhance classification accuracy.
- The variable weight classifiers with RJP features outperforms the equal weight classifiers by 3.06%. This shows that the weight optimization algorithm enhances the system accuracy.
- In all tests, the variable weight classifier performs better than all of the individual reliability term classifiers. This supports our algorithm of using multiple reliability terms.
- The variable weight classifier with RJP features outperforms the state-of-the-art approaches Lie group representation [62] and moving pose [63] by 3.77% and 6.70%, respectively. This highlights the effectiveness of our proposed variable weight classifier.

The reliability measurements are estimation of the true reliability. While they correctly evaluate the joints in general, individual terms may be inaccurate under specific situations. This explains why the classification accuracy drops for some individual term classifiers comparing to the classifier using relative joint position only. Our proposed method has the strength of combining multiple reliability terms, such that we can tolerance errors in individual terms and produce consistent results.

Table 6
Accuracy in classifying postures in the sitting to work experiment.

Method		Average % accuracy
Joint positions		66.67
Relative joint positions (RJP) [61]		70.58
Lie group representation [62]		71.41
Moving pose [63]		69.94
Moving pose [63] with pose normalization and noise removal		68.55
Proposed	RJP with *Rb* only	71.72
	RJP with *Rk* only	72.57
	RJP with *Rc* only	71.57
	RJP with *Rd* only	72.25
	RJP with *Rb*, *Rk*, *Rc* and *Rd*—equal weight	72.60
	RJP with *Rb*, *Rk*, *Rc* and *Rd*—variable weight	**79.45**

7.4. Sitting on a chair and working on a work bench

Here, we perform evaluation on the sitting to work posture database, which includes healthy, B-1 and B-2 postures as explained in Section 6.2. Example postures can be found in Fig. 3 and details of the data used in the experiment can be found in Table 4. On average, 1621 and 2702 postures were used as training and testing data in each *leave-one-subject-out* classification trial. The feature vector size of the joint position and relative joint position features are 30-d and 135-d, respectively. The average classification accuracies are shown in Table 6.

According to the results:

- Our variable weight classifier with RJP features has made a significant improvement over the classifier with RJP features only. Accuracy is enhanced by 8.87%.
- The variable weight classifier outperforms equal weight classifier by 6.85%, supporting our weight optimization algorithm.
- The variable weight classifier outperforms all single reliability term classifiers in both tests, supporting our algorithm of using all four terms.
- All of the single reliability term classifiers with RJP features perform better than the classifier with RJP features only. This shows that accuracy is enhanced by reliability measurement in general. More discussion about this can be found in Section 8.
- The variable weight classifier and all of the individual reliability term classifiers outperform the state-of-the-art approaches Lie group representation [62] and moving pose [63] by 0.16%–8.04% and 3.02%–9.51%, respectively. This highlights the effectiveness of our proposed method.

7.5. Postures of different semantic meaning from benchmark datasets

Here, we show that our proposed algorithm can enhance the accuracy of movement semantic classification. We utilize the 3D skeletal data in the MSR Action3D dataset [59] and Florence 3D Actions dataset [60] in Sections 7.5.1 and 7.5.2, respectively.

7.5.1. MSR Action3D dataset

The dataset contains 20 action classes and each action is performed by 10 subjects with 2–3 trials, and 557 motion sequences were used in the experiment as in [61]. We follow [61] to conduct a cross subject test by classifying motions from 20 action classes: *high arm wave, horizontal arm wave, hammer, hand catch, forward punch, high throw, draw x, draw tick, draw circle, hand clap, two hand wave, side-boxing, bend, forward kick, side kick, jogging, tennis swing, tennis serve, golf swing, pickup and throw*. The motions of half of the subjects are used in training and the rest are used as testing data.

We classify the motions by training the proposed binary classifier in a one-versus-all manner. Since the length of the motions

Table 7
Accuracy in classifying postures in the MSR Action3D [59] dataset with 20 action classes.

Method		Average % accuracy
Joint positions		87.74
Relative joint positions (RJP) [61]		88.23
Bag of 3D points [59]		74.70
Histogram of 3D joints [65]		78.97
Shape and motion features [66]		82.10
EigenJoints [67]		82.30
Joint angle similarities [68]		83.53
Actionlet ensemble [61]		88.20
Spatial and temporal part-sets [69]		90.22
Covariance descriptors on 3D joint locations [70]		90.53
Random forests [71]		90.90
Moving pose [63]		91.70
Lie group representation [62]		92.46
Proposed	RJP with *Rb* only	89.88
	RJP with *Rk* only	90.70
	RJP with *Rd* only	88.81
	RJP with *Rb*, *Rk* and *Rd*—equal weight	90.39
	RJP with *Rb*, *Rk* and *Rd*—variable weight	**93.36**

Table 8
Accuracy in classifying postures in the Florence 3D [60] dataset with nine action classes.

Method		Average % accuracy
Protocol of [62]—Half–half data split		
Joint positions		85.44
Relative joint positions (RJP) [61]		89.66
Moving pose [63]		81.42
EigenJoints [67]		87.28
Lie group representation [62]		90.88
Proposed	RJP with *Rb* only	86.95
	RJP with *Rk* only	89.76
	RJP with *Rb* and *Rk*—equal weight	89.97
	RJP with *Rb* and *Rk*—variable weight	**93.29**
Protocol of [60]—Leave-one-subject-out		
Joint positions		84.69
Relative joint positions (RJP) [61]		91.42
NNBB + parts + time [60]		82.00
EigenJoints [67]		89.53
LARP+TSRVF [72]		89.50
LARP+mfPCA [72]		89.67
Elastic shape analysis [73]		89.67
Taha et al. [74]		96.20
Proposed	RJP with *Rb* only	91.08
	RJP with *Rk* only	91.75
	RJP with *Rb* and *Rk*—equal weight	91.75
	RJP with *Rb* and *Rk*—variable weight	**98.33**

are not equal, we temporally align each motion to a *class template* motion which is having the minimum variance with all other positive training motions in each class. Then, to reduce the temporal dimensionality of the motions, we extract representative keyframes (17 keyframes in our experiment) to represent the class template using Frame Decimation [64]. Next, all training data (i.e., positive and negative) are aligned to the class template by dynamic time warping (DTW) and we train a classifier using the temporally aligned training data in each class. When classifying a testing motion, we temporally align the testing motion to all class templates and compute the decision value using the trained classifier in each class. The feature vector representing each motion is created by concatenating the temporally aligned frame-based features. On average, the number of motions for training is 284 and that of testing is 273. Since only the skeletal data and depth image sequences are available in this dataset, we can only calculate three reliability terms *Rb*, *Rk*, and *Rd* in our experiments. The accuracy of the classifiers is shown in Table 7.

According to the results:

- Our variable weight classifier with RJP features has made an significant improvement over the classifier with RJP features only. Accuracy is enhanced significantly by 5.13%.
- The variable weight classifier outperforms equal weight classifier by 2.97%, showing the effectiveness of our weight optimization algorithm.
- The variable weight classifier outperforms all single reliability term classifiers by 2.66%–4.55%, supporting our algorithm of using all three terms.
- All of the single reliability term classifiers perform better than the classifier with RJP features only. This shows that accuracy is enhanced by reliability measurement in general. More discussion about this can be found in Section 8.
- Even though the state-of-the-art approaches such as Lie group representation [62] and moving pose [63] achieved very high performance in this dataset, our variable weight classifier achieves an even better result by taking into account the reliability measurement in motion classification.

When compared with the Lie group representation [62] on the MSR Action3D dataset, our proposed variable weight optimizing approach outperforms the previous method with a smaller margin than other experiments in this paper. It is because the motions are captured in higher quality in general when compare with other datasets used. In particular, all motions are recorded in a front-facing manner and the subjects are in standing pose without occlusion by other objects. As a result, the motions are in higher quality and there is less room for improvement by analyzing the joint accuracy in this dataset. Nevertheless, our method still outperforms the state-of-the-art approaches and this highlight the robustness and consistency of our proposed method.

7.5.2. Florence 3D Actions dataset

In this experiment, we evaluate the accuracy of classifying motions from the skeleton data in the Florence 3D Actions dataset [60]. The dataset contains nine action classes: *wave, drink from a bottle, answer phone, clap, tight lace, sit down, stand up, read watch, bow*. Each action which is performed by 10 subjects with 2–3 trials, and 215 motion sequences were used in the experiment as in [60].

We follow [62] to classify motions from all nine action classes by using the motions of half of the subjects as training and the rest as testing and follow [60] to perform leave-one-subject-out classification, and report the average classification accuracy. Similar to Section 7.5.1, we classify the motions by training the proposed binary classifier in a one-versus-all manner. We also find the *class template* motion (with nine keyframes) and all training and testing data are aligned to the class template by DTW as explained in last section. On average, the number of motions for training is 109 and that of testing is 106. Since only the skeletal data are available in this dataset, we can only calculate two reliability terms *Rb* and *Rk* in our experiments. The results are shown in Table 8.

According to the results, in the experiments using the half–half data split setting as in [62]:

- Our variable weight classifier with RJP features has made an significant improvement over the classifier with RJP features only by 3.63%.
- The variable weight classifier significantly outperforms equal weight classifier by 3.32%, showing the effectiveness of our weight optimization algorithm.
- The variable weight classifier outperforms all single reliability term classifiers by 3.53%–6.34%, supporting our algorithm of using all two terms.

Please cite this article as: E.S.L. Ho et al., Improving posture classification accuracy for depth sensor-based human activity monitoring in smart environments, Computer Vision and Image Understanding (2016), http://dx.doi.org/10.1016/j.cviu.2015.12.011

- Our variable weight classifier out-perform the state-of-the-art approaches such as Lie group representation [62] and moving pose [63] by 2.41% and 11.87%, respectively. This highlights the effectiveness of our proposed method.

In the experiments using the leave-one-subject-out data split setting as in [60], the results also showed the same pattern as our proposed variable weight classifier outperforms all single reliability term classifier as well as existing approached. This highlight the consistency and robustness of our method across different experiment settings.

8. Discussion and conclusions

In this paper, we presented a data-driven framework that considers the reliability of the source data to classify postures captured from depth cameras. We propose new reliability terms to better evaluate the features, and present a customized max-margin classification framework that takes in the measurements. Our framework can classify the subtle different between healthy and unhealthy postures in a workplace environment. We made our motion database available to public usage in order to facilitate further research in this area.

Since the postures captured by Kinect is incomplete and noisy due to occlusion, it is proposed to reconstruct the unreliable joints using prior knowledge [9]. A traditional method of posture classification is to evaluate the reconstructed posture. However, since the reconstruction process involve modifying unreliable features, it introduces another major source of error. We opt for a max-margin classification framework, which evaluates posture considering joints with high reliability more, and do not require altering the posture.

As a common problem of data-driven approaches, if there is no posture similar to the observed one in the database, our method may fail. This is because we do not have the knowledge to accurately classify the posture. This could happen if the user has a significant different body size or segment length proportion. In the future, we would like to explore motion retargeting techniques to retarget the observed posture.

Apart from unhealthy postures, moving rapidly or keeping the body static for extensive long duration can also result in injury. To identify these kind of movements, the spatio-temporal information of the motion has to be considered. In order to efficiently classify long duration of movement, abstraction in the temporal domain may also be needed. We are interested to explore this area in the future to broaden the scope of our classification algorithm.

This research demonstrates how our framework can be applied in smart environments to identify incorrectly performed working posture. There are other motions, such as wheelchair handing, floor sweeping and window cleaning, that have a high risk of injury. As a future work, we wish to enhance the database to include a wide variety of motions. Apart from capturing data ourselves, we would like to set up a standard format for capturing different types of motion in the topic of workspace health and safety, such that interested researchers can contribute and share captured motions.

Acknowledgments

This work was supported in part by Hong Kong Baptist University Science Faculty Research Grants (FRG2/13-14/092), NSFC Young Scientist Research Grant (project no. 61302176), the Hong Kong Research Grant Council (project no. GRF210813), NSFC grant (project no. 61272366) and the Engineering and Physical Sciences Research Council (EPSRC) (Ref: EP/M002632/1).

References

[1] Health and Safety Executive, Health and safety executive annual statistics report for Great Britain 2012/2013. http://www.hse.gov.uk/statistics/overall/hssh1213.pdf, 2013. (accessed 20.10.15).
[2] V.B. Zordan, N.C. Van Der Horst, Mapping optical motion capture data to skeletal motion using a physical model, in: Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA'03), Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 2003, pp. 245–250.
[3] D.F. Fouhey, V. Delaitre, A. Gupta, A.A. Efros, I. Laptev, J. Sivic, People watching: Human actions as a cue for single-view geometry, in: Proceedings of the 12th European Conference on Computer Vision, Springer Berlin Heidelberg, 2012, pp. 732–745.
[4] H. Shum, E.S. Ho, Real-time physical modelling of character movements with microsoft kinect, in: Proceedings of the 18th ACM Symposium on Virtual Reality Software and Technology (VRST'12), ACM, New York, NY, USA, 2012, pp. 17–24.
[5] A. Jalal, Y. Kim, Dense depth maps-based human pose tracking and recognition in dynamic scenes using ridge data, in: 11th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), IEEE, 2014, pp. 119–124.
[6] A. Jalal, S. Kamal, D. Kim, Depth map-based human activity tracking and recognition using body joints features and self-organized map, in: 2014 International Conference on Computing, Communication and Networking Technologies (ICCCNT), IEEE, 2014, pp. 1–6.
[7] L. Piyathilaka, S. Kodagoda, Gaussian mixture based HMM for human daily activity recognition using 3d skeleton features, in: 8th IEEE Conference on Industrial Electronics and Applications (ICIEA), IEEE, 2013, pp. 567–572.
[8] J. Chai, J.K. Hodgins, Performance animation from low-dimensional control signals, ACM Trans. Graph. 24 (3) (2005) 686–696.
[9] H. Shum, E. Ho, Y. Jiang, S. Takagi, Real-time posture reconstruction for microsoft kinect, IEEE Trans. Cyber. 43 (5) (2013) 1357–1369.
[10] O. Lara, M. Labrador, A survey on human activity recognition using wearable sensors, IEEE Commun. Surv. Tutor. 15 (3) (2013) 1192–1209.
[11] J. Machajdik, A. Hanbury, A. Garz, R. Sablatnig, Affective computing for wearable diary and lifelogging systems: An overview, in: Workshop of the Austrian Association for Pattern Recognition, 2011.
[12] J. Hamm, B. Stone, M. Belkin, S. Dennis, Automatic annotation of daily activity from smartphone-based multisensory streams, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 110 (2013) 328–342.
[13] R. Slyper, J.K. Hodgins, Action capture with accelerometers, in: Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA'08), Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 2008, pp. 193–199.
[14] H.P.H. Shum, T. Komura, S. Takagi, Fast accelerometer-based motion recognition with a dual buffer framework, Int. J. Virtual Real. 10 (3) (2011) 17–24.
[15] J.F. O'Brien, R.E. Bodenheimer, G.J. Brostow, J.K. Hodgins, Automatic joint parameter estimation from magnetic motion capture data, in: Proceedings of Graphics Interface 2000, CRC Press, 2000, pp. 53–60.
[16] A. Yang, S. Iyengar, S. Sastry, R. Bajcsy, P. Kuryloski, R. Jafari, Distributed segmentation and classification of human actions using a wearable motion sensor network, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, IEEE, 2008, pp. 1–8.
[17] J. Ward, P. Lukowicz, G. Troster, T. Starner, Activity recognition of assembly tasks using body-worn microphones and accelerometers, IEEE Trans. Pattern Anal. Mach. Intell. 28 (10) (2006) 1553–1567.
[18] J.B. Kim, H.J. Kim, Efficient region-based motion segmentation for a video monitoring system, Pattern Recogn. Lett. 24 (1-3) (2003) 113–128.
[19] P.F. Felzenszwalb, D.P. Huttenlocher, Pictorial structures for object recognition, Int. J. Comput. Vis. 61 (1) (2005) 55–79.
[20] A. Gupta, A. Kembhavi, L. Davis, Observing human-object interactions: Using spatial and functional compatibility for recognition, IEEE Trans. Pattern Anal. Mach. Intell. 31 (10) (2009) 1775–1789.
[21] D.F. Fouhey, V. Delaitre, A. Gupta, A.A. Efros, I. Laptev, J. Sivic, People watching: Human actions as a cue for single-view geometry, in: Proceedings of the 12th European Conference on Computer Vision, Springer Berlin Heidelberg, 2012, pp. 259–274.
[22] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, A. Blake, Real-time human pose recognition in parts from single depth images, in: Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'11), IEEE Computer Society, Washington, DC, USA, 2011, pp. 1297–1304.
[23] J. Shotton, R. Girshick, A. Fitzgibbon, T. Sharp, M. Cook, M. Finocchio, R. Moore, P. Kohli, A. Criminisi, A. Kipman, A. Blake, Efficient human pose estimation from single depth images, IEEE Trans. Pattern Anal. Mach. Intell. 12 (2012) 2821–2840.
[24] M. Ye, X. Wang, R. Yang, L. Ren, M. Pollefeys, Accurate 3d pose estimation from a single depth image, in: Proceedings of the 2011 International Conference on Computer Vision (ICCV'11), IEEE Computer Society, Washington, DC, USA, 2011, pp. 731–738.
[25] A. Baak, M. Muller, G. Bharaj, H.-P. Seidel, C. Theobalt, A data-driven approach for real-time full body pose reconstruction from a depth camera, in: Proceedings of the 2011 International Conference on Computer Vision (ICCV'11), IEEE Computer Society, Washington, DC, USA, 2011, pp. 1092–1099.

Please cite this article as: E.S.L. Ho et al., Improving posture classification accuracy for depth sensor-based human activity monitoring in smart environments, Computer Vision and Image Understanding (2016), http://dx.doi.org/10.1016/j.cviu.2015.12.011

[26] S. Kean, J. Hall, P. Perry, Meet the Kinect: An Introduction to Programming Natural User Interfaces, first, Apress, Berkely, CA, USA, 2011.
[27] Y. Song, J. Tang, F. Liu, S. Yan, Body surface context: A new robust feature for action recognition from depth videos, IEEE Trans. Circuits Syst. Video Technol. 24 (6) (2014) 952–964.
[28] A. Jalal, S. Lee, J. Kim, T.-S. Kim, Human activity recognition via the features of labeled depth body parts, Lecture Notes in Computer Science 7251 (2012a) 246–249.
[29] A. Jalal, N. Sarif, J.T. Kim, T.-S. Kim, Human activity recognition via recognized body parts of human depth silhouettes for residents monitoring services at smart home, Indoor Built Environ. 22 (1) (2012b) 271–279.
[30] A. Farooq, A. Jalal, S. Kamal, Dense rgb-d map-based human tracking and activity recognition using skin joints features and self-organizing map, KSIITrans. Internet Inform. Syst. 9 (5) (2015) 1856–1869.
[31] A. Jalal, S. Kamal, D. Kim, Shape and motion features approach for activity tracking and recognition from kinect video camera, in: IEEE 29th International Conference on Advanced Information Networking and Applications Workshops (WAINA), IEEE, 2015, pp. 445–450.
[32] S. Althloothi, M.H. Mahoor, X. Zhang, R.M. Voyles, Human activity recognition using multi-features and multiple kernel learning, Pattern Recogn. 47 (5) (2014) 1800–1812.
[33] A. Jalal, M. Uddin, T.-S. Kim, Depth video-based human activity recognition system using translation and scaling invariant features for life logging at smart home, IEEE Trans. Consumer Electr. 58 (3) (2012) 863–871.
[34] M. Parajuli, D. Tran, W. Ma, D. Sharma, Senior health monitoring using kinect, in: 2012 Fourth International Conference on Communications and Electronics (ICCE), IEEE, 2012, pp. 309–312.
[35] A. Jalal, S. Kamal, Real-time life logging via a depth silhouette-based human activity recognition system for smart home services, in: 11th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), IEEE, 2014, pp. 74–80.
[36] A. Jalal, S. Kamal, D. Kim, A depth video sensor-based life-logging human activity recognition system for elderly care in smart indoor environments, Sensors 14 (7) (2014) 11735–11759.
[37] A. Jalal, J.T. Kim, T.S. Kim, Development of a life logging system via depth imaging-based human activity recognition for smart homes, in: International Symposium on Sustainable Healthy Buildings, International Society of Indoor Air Quality and Climate (ISIAQ), 2012, pp. 91–95.
[38] A. Uribe-Quevedo, B. Perez-Gutierrez, C. Guerrero-Rincon, Seated tracking for correcting computer work postures, in: 2013 29th Southern Biomedical Engineering Conference (SBEC), IEEE, 2013, pp. 169–170.
[39] P. Paliyawan, C. Nukoolkit, P. Mongkolnam, Prolonged sitting detection for office workers syndrome prevention using kinect, in: 2014 11th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), IEEE, 2014, pp. 1–6.
[40] C. Martin, D. Burkert, K. Choi, N. Wieczorek, P. McGregor, R. Herrmann, P. Beling, A real-time ergonomic monitoring system using the microsoft kinect, in: IEEE Systems and Information Design Symposium (SIEDS), IEEE, 2012, pp. 50–55.
[41] W. Zhao, D. Espy, M. Reinthal, H. Feng, A feasibility study of using a single kinect sensor for rehabilitation exercises monitoring: A rule based approach, in: IEEE Symposium on Computational Intelligence in Healthcare and e-health (CICARE), IEEE, 2014, pp. 1–8.
[42] F. Cary, O. Postolache, P. Silva Girao, Kinect based system and artificial neural networks classifiers for physiotherapy assessment, in: IEEE International Symposium on Medical Measurements and Applications (MeMeA), IEEE, 2014, pp. 1–6.
[43] Z. Yang, L. Zicheng, C. Hong, Rgb-depth feature for 3d human activity recognition, China Commun. 10 (7) (2013) 93–103.
[44] M. Müller, T. Röder, M. Clausen, Efficient content-based retrieval of motion capture data, ACM Trans. Graph. 24 (3) (2005) 677–685.
[45] M. Müller, T. Röder, Motion templates for automatic classification and retrieval of motion capture data, in: Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA'06), Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 2006, pp. 137–146.
[46] E.S.L. Ho, T. Komura, Indexing and retrieving motions of characters in close contact, IEEE Trans. Visual. Comput. Graph. 15 (3) (2009) 481–492.
[47] J.K.T. Tang, H. Leung, T. Komura, H.P.H. Shum, Emulating human perception of motion similarity, Comput. Animat. Virtual Worlds 19 (3-4) (2008) 211–221.
[48] J. Tautges, A. Zinke, B. Krüger, J. Baumann, A. Weber, T. Helten, M. Müller, H.-P. Seidel, B. Eberhardt, Motion reconstruction using sparse accelerometer data, ACM Trans. Graph. 30 (3) (2011) 18:1–18:12.
[49] L. Ren, G. Shakhnarovich, J.K. Hodgins, H. Pfister, P. Viola, Learning silhouette features for control of human motion, ACM Trans. Graph. 24 (4) (2005) 1303–1331.
[50] L. Zhou, Z. Liu, H. Leung, H.P.H. Shum, Posture reconstruction using kinect with a probabilistic model, in: Proceedings of the 20th ACM Symposium on Virtual Reality Software and Technology (VRST'14), ACM, New York, NY, USA, 2014, pp. 117–125.
[51] J. Puwein, L. Ballan, R. Ziegler, M. Pollefeys, Joint camera pose estimation and 3d human pose estimation in a multi-camera setup, Lecture Notes in Computer Science 9004 (2015) 473–487.
[52] A. Jalal, Y. Kim, D. Kim, Ridge body parts features for human pose estimation and recognition from rgb-d video data, in: 2014 International Conference on Computing, Communication and Networking Technologies (ICCCNT), IEEE, 2014, pp. 1–6.
[53] H.G. Armstrong, Anthropometry and Mass Distribution for Human Analogues, vol. 1. Military Male Aviators, Defense Technical Information Center, 1988.
[54] G. Chechik, G. Heitz, G. Elidan, P. Abbeel, D. Koller, Max-margin classification of data with absent features, J. Mach. Learn. Res. 9 (2008) 1–21.
[55] V.N. Vapnik, The Nature of Statistical Learning Theory, Springer-Verlag New York, Inc., New York, NY, USA, 1995.
[56] P. Felzenszwalb, R. Girshick, D. McAllester, D. Ramanan, Object detection with discriminatively trained part-based models, IEEE Trans. Pattern Anal. Mach. Intell. 32 (9) (2010) 1627–1645.
[57] Microsoft Corporation, Kinect for windows SDK programmingguide version 1.8, 2013.
[58] European Agency for Safety and Health at Work, E-fact 45 - checklist for preventing bad working postures. https://osha.europa.eu/en/publications/e-facts/efact45, 2008. (accessed 20.10.15).
[59] W. Li, Z. Zhang, Z. Liu, Action recognition based on a bag of 3d points, in: CVPR Workshops, IEEE, 2010, pp. 9–14.
[60] L. Seidenari, V. Varano, S. Berretti, A. Del Bimbo, P. Pala, Recognizing actions from depth cameras as weakly aligned multi-part bag-of-poses, in: CVPR Workshops, IEEE, 2013, pp. 479–485.
[61] J. Wang, Z. Liu, Y. Wu, J. Yuan, Mining actionlet ensemble for action recognition with depth cameras, in: Proceedings of CVPR 2012, IEEE, 2012, pp. 1290–1297.
[62] R. Vemulapalli, F. Arrate, R. Chellappa, Human action recognition by representing 3d human skeletons as points in a lie group, in: Proceedings of CVPR'14, IEEE, 2014, pp. 588–595.
[63] M. Zanfir, M. Leordeanu, C. Sminchisescu, The moving pose: An efficient 3d kinematics descriptor for low-latency action recognition and detection, in: Proceedings of ICCV'13, IEEE, 2013, pp. 2752–2759.
[64] S. Li, M. Okuda, S. Takahashi, Embedded key-frame extraction for cg animation by frame decimation, in: IEEE International Conference on Multimedia and Expo, 2005 (ICME'05), IEEE, 2005, pp. 1404–1407.
[65] L. Xia, C.-C. Chen, J. Aggarwal, View invariant human action recognition using histograms of 3d joints, in: CVPR Workshops, IEEE, 2012, pp. 20–27.
[66] A. Jalal, S. Kamal, D. Kim, Shape and motion features approach for activity tracking and recognition from kinect video camera, in: IEEE 29th International Conference on Advanced Information Networking and Applications Workshops (WAINA), IEEE, 2015, pp. 445–450.
[67] X. Yang, Y. Tian, Eigenjoints-based action recognition using naïve-bayes-nearest-neighbor, in: CVPR Workshops, IEEE, 2012, pp. 14–19.
[68] E. Ohn-Bar, M. Trivedi, Joint angles similarities and hog2 for action recognition, in: IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), IEEE, 2013, pp. 465–470.
[69] C. Wang, Y. Wang, A. Yuille, An approach to pose-based action recognition, in: Proceedings of CVPR'13, IEEE, 2013, pp. 915–922.
[70] M.E. Hussein, M. Torki, M.A. Gowayyed, M. El-Saban, Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations, in: Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence (IJCAI'13), AAAI Press, 2013, pp. 2466–2472.
[71] Y. Zhu, W. Chen, G. Guo, Fusing spatiotemporal features and joints for 3d action recognition, in: IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), IEEE, 2013, pp. 486–491.
[72] R. Anirudh, P. Turaga, J. Su, A. Srivastava, Elastic functional coding of human actions: From vector-fields to latent variables, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2015, pp. 3147–3155.
[73] M. Devanne, H. Wannous, S. Berretti, P. Pala, M. Daoudi, A. Del Bimbo, 3-d human action recognition by shape analysis of motion trajectories on riemannian manifold, IEEE Trans. Cyber. 45 (7) (2015) 1340–1352.
[74] A. Taha, H.H. Zayed, M.E. Khalifa, E.-S. M. El-Horbaty, Human activity recognition for surveillance applications, in: The 7th International Conference on Information Technology (ICIT'15), Al-Zaytoonah University, (in Jordan), 2015, pp. 577–586.

Please cite this article as: E.S.L. Ho et al., Improving posture classification accuracy for depth sensor-based human activity monitoring in smart environments, Computer Vision and Image Understanding (2016), http://dx.doi.org/10.1016/j.cviu.2015.12.011