Human motion and gesture recognition receive much concern in sports field, such as physical education and fitness for all. Although plenty of mature applications appear in sports training using photography, video camera, or professional sensing devices, they are either expensive or inconvenient to carry. MEMS devices would be a wise choice for students and ordinary body builders as they are portable and have many built-in sensors. In fact, recognition of hand gestures is discussed in many studies using inertial sensors based on similarity matching. However, this kind of solution is not accurate enough for human movement recognition and cost much time. In this paper, we discuss motion recognition in sports training using features extracted from distance estimation of different kinds of sensors. To deal with the multivariate motion sequence, we propose a solution that applies Max-Correlation and Min-Redundancy strategy to select features extracted with interclass distance similarity estimation. With this method, we are able to screen out proper features that can distinguish motions in different classes effectively. According to the results of experiment in real world application in dance practice, our solution is quite effective with fair accuracy and low time cost.

1. Introduction

Recognition of human motions and gestures is commonly studied in sports field so as to serve movement analysis, guidance, and evaluation. Since direct observation costs too much human resources, information techniques are introduced to assist the tasks. Video and photography are used to record human movement, and there are quite a lot of researches in analyzing the visual information [1]. For more precise analysis, 3D modeling technology [2, 3] is also used to build 3-dimensional model of movement of human body. These methods provide accurate report of human motions and gestures, but for ordinary students and exercisers, they may not afford such devices. Meanwhile, the application of these visual devices is not convenient due to the restriction of space. As for the development of smart mobile terminals, wearable sensors are popularized in many fields, such as medical and health services [4–6] as well as sports domain [7, 8]. Take running, for instance, we are able to detect the running speed, trajectories, and the calories we burn during the exercises. Most of these sensing data are obtained by microelectromechanical systems (MEMS), which is the technology of very small devices. It merges at the nanoscale into nanoelectromechanical systems (NEMS) and nanotechnology [9]. MEMS sensors can be embedded in smart phone, watch, wristband, and other wearable objects, even clothes and shoes. These sensors are portable and low cost, which is quite acceptable for the public. Therefore, to assist sports training of students, we collect the motion data with MEMS sensors for certain action recognition. Although quite a lot of contributions have been made in this domain, most of them focus on regular activity recognition with repeated motions. Recent studies in gestures recognition usually involve time consuming similarity comparison in
testing phase that might be a problem in recognizing long time complex motions.

In this paper, we attempt to find a fast and accurate way to recognize certain sports movements with wearable sensors. Our job includes motion data collection and preprocessing with multiple motion sensors embedded in MEMS; feature extraction and selection; and class determination of the unlabeled data. In this process, we may face the following difficulties and problems:

(i) Selection and fusion of multiple sensors.
(ii) Similarity matching between multivariant motion series.
(iii) How to improve the efficiency for online detection.

As there is no clear guidance for sensor selection, the choices are various in different studies. Normally, activity recognition would choose more sensors and related statistics to generate different features, while recognition based on distance metric usually chooses few key sensor data obtained after dimensional reduction for the comparison. In our study, we will discuss different situations with single or multiple sensors. Since the data we obtain come from different sensors and DoF, we need to find an effective way to deal with this multidimensional data. Moreover, finding an efficient solution for online motion recognition based on distance metric that considers the mutual distance relationship information will be another important task. As a result, we propose a flexible and interpretable solution using interclass distance features from the mutual distance between testing data and different classes. With Max-Correlation and Min-Redundancy strategy, we choose the nonredundant and distinguishable features and apply them in traditional classifiers. After careful observation of related experiments, we find that DTW based on SAX would be a good trade-off between time and accuracy as the distance metric. This solution is applied in physical education using smart phone to detect and analyze basic movements in dance practice.

The rest of the paper is organized as follows: related works are discussed in Section 2; data collection and basic preprocessing are discussed in Section 3; in Section 4, we describe our solution starting from feature extraction and selection to data classification; discussion of related experiments is arranged in Section 5; and finally comes the conclusion of the study in Section 6.

2. Related Work

Wearable sensors are usually used to detect different activities, hand gestures, and specific motions. Activity recognition [10, 11] is used to distinguish different states of human, such as running, walking, going upstairs, and going downstairs. These states consisted of repeated motions, which is a special characteristic that can be used in pattern recognition. Hand gestures recognition in man-machine interaction is also a hot topic that attracts much attention [12, 13]. Similarity comparison between the obtained data and standard model is a common way to solve the problem. There are also studies that focus on specific movement detection, such as fall [14], smoking [5], and body shaking [15]. In some cases, multiple sensors on different parts of human body would improve the recognition accuracy; thus wearable sensor groups are built to detect human status [16, 17]. At present, beside the body sensor network with different types of sensors on different part of the body that can be used in medical field, cluster of the same type of sensors on different part of human body is mainly discussed in activity and motion analysis. Professional training may prefer high accuracy, and multiple sensors would be a better choice. As for ordinary practice, sensors on a single part of the body would be wiser, considering the trade-off between accuracy, cost, and convenience. Most studies choose choose acceleration information to judge human state [18], and some of them also take gyroscope and magnetometer into consideration [19]. Sensor fusion is discussed in activity recognition based on feature extraction in time and frequency domains [20]. According to the observation of related experiments, sensor selection and fusion have close relations with the discussed activities, placing position of the sensor, type of the classifier, and the extracted features. It is difficult to make the choice beforehand. Therefore, how to make full use of information from different types of sensors efficiently without redundancy is a significant job to do.

As for motion recognition, there are generally two ways to handle the data. The first one is to extract features in time and frequency domains in each fragment divided by sliding window. And then, perform dimensional reduction on these features. After that, ordinary classifiers can be used to predict motion categories. This method is commonly used in activity recognition that contains repeated motions [21]. In such cases, the probability and statistics information is more efficient and useful. However, in recognizing specific motions such certain sports gestures and movements, the differences between various motion types might exist in shape of certain location. And the statistical features might fail in reflecting these differences. Another solution of motion recognition is based on distance metric. Distances between the testing data and training data can be used to build a K-NN (1-NN + DTW is supposed to be quite effective) classifier for further prediction [22, 23]. But the comparison in the testing phase might take too much time. For some high time cost distance measurements such as DTW, this may affect the efficiency, especially for online motion recognition. Therefore, study in [24] proposed an application of Nearest Centroid Classifier using average series based on DTW [25] as centroids. Also, template matching is quite popular. By directly comparing the testing motion data with a few prototype template generated from different classes called motifs, we are able to tell the type of motion the testing data belongs to. The main idea of the previous related researches focus on finding the optimized samples that can better represent the features of the categories they belong to [26–31]. Some studies also consider the separability between different classes [32].

Motion data can be treated as time series, and there are many related studies around the topic of time series similarity estimation. First of all, we can divide the existing methods into two categories: estimations based on global similarity and local similarity. Solutions belonging to the first category
use the global matching cost/distance to be the similarity measurement, while, for data with noise, local similarity may be more accurate. The structure of related works are shown below:

(i) Global matching similarity:

(1) Similarity based on different distance metric:
   (a) Dynamic time warping (DTW) [33, 34]:
      (I) DTW with constraints [35, 36],
      (II) Dynamic Manifold Warping (DMW) [37],
      (III) TWED [38].
   (b) Traditional distance:
      (I) Manhattan distance.
      (II) Euclidean distance.
      (III) Mahalanobis distance.

   (2) Symbolic based measurement:
      (a) Edit distance.
      (b) Symbolic Aggregate approximation (SAX) [39].

(ii) Local matching similarity:

   (1) Longest Common Subsequence (LCS) [40].
   (2) Similarity based on shapelet [41].

Dynamic time warping (DTW) is most commonly used in real-time generated time series like motion and voice signal. Unlike Euclidean distance, this method tolerates offsets and time shifting during the comparison, which is quite suitable for our situation. But the time consumption of DTW is quite high, and it is one of the challenges we need to deal with. There are many attempts in improving DTW in different aspects. For example, different kinds of constraints are set for comparison, so as to reduce the time consumption [42]. Symbolic comparison is another common way to measure the similarities between time series, such as Edit distance and LCS. These methods are more suitable for symbolic comparison like string matching. As for continuous time series, they need to combine with other methods. For example, TWED use Edit distance to replace Euclidean distance in the matching process of DTW. Before the direct matching, transformation of raw data or related operation should be defined. SAX is a simple method that converts continuous time series into discrete symbolic series. We can also use SAX to compute the lower bound distance efficiently. Meanwhile, the process of discrete symbolic representation is also a special way of dimensional reduction. Direct comparison with one labelled sample from each class is fast and easy, but its accuracy cannot compare with classifiers. In some circumstances, the shape of the time series with noise might be quite similar globally, while local features are various according to different categories. Shapelet was proposed to solve such problem. Shapelet is a subsequence taken from one of the time series, whose distance to different time series can be used to distinguish time series from different categories. In order to work with commonly used classifiers, the solution called shapelet transform [43, 44] was brought up. The distances between a time series and all shapelets are treated as the selected features of the time series that can be handled in all kinds of classifiers.

Most of the existing distance based methods determine the categories of the unlabeled data according to their closest motifs or samples selected by different mechanisms. The guidance information of the classification is limited in the distance between testing data and the nearest model, no matter how superior these prototypes are. Information of the mutual distance relationship between the testing data and different classes is left out. Thus, how to make rational use of the mutual distance between different classes is the key problem we need to solve. In our solution, we consider the information of mutual distance between samples and different classes and use them as features for classification. In this way, we can speed up the recognition while maintaining high accuracy.

3. Data Collection

According to the situation and requirement of motion practice in physical education, we choose sensors embedded in smart phone to obtain the data of movements and gestures. Nowadays a smart phone carries more and more types of sensors, including accelerometer, gyroscope, gravity sensor, and magnetometer. These sensors reflect the status of the phone from different aspects. 3D accelerometer measures the acceleration on three mutual perpendicular directions. This acceleration is affected by gravity, so we need to separate the linear acceleration from the original data. 3D gyroscope obtains the rotation rate on 3 coordinates. Based on the readings of the sensors, we are able to obtain the attitude information of the smart phone. Magnetometer measures magnetic density and direction, so that we can detect the orientation of the smart phone.

In our case, we use iPhone 6 to sense human motion. The system editions after iOS 4 provide us with Core Motion Framework that can read the data of all kinds of sensors and conduct essential preprocessing. Specifically, with the class called CMDDeviceMotion, we are able to get information about movement and attitude of the smart phone. This class mainly consists of four parts: attitude information calculated with reading of the embedded sensors; gravity data obtained by accessing gravity sensor; acceleration that has removed the influence of gravity and finished the filtering; rotation rate that has removed the bias of gyroscope. By calculating the reading of the different sensors, attitude of the smart phone is described as a quaternion and three-dimensional Euler angles: yaw, roll, and pitch.

After careful observation and analysis, we decided to wear a packet on the frontal side of the waist and put the smart phone inside the packet with its screen towards the front. We set 20 Hz as the sampling rate. We show some of the data we obtain from motion A in Figure 1: stepping forward and backward in 8 beats.

Since the direction of the phone changes constantly, it is hard to determine the instant state of the phone. One common way to eliminate such influence is to add a dimension:
magnitude. As for acceleration in three dimensions, $x$, $y$, and $z$, magnitude is defined in the following:

$$\text{magnitude} = \sqrt{x^2 + y^2 + z^2}.$$  \hspace{1cm} (1)

With magnitude, we can ignore the variation of direction at a certain degree, but we fail to see the status on each dimension separately. Thus, we transform the linear acceleration and gyroscope data from coordinates of the phone to Earth coordinates. It is calculated with the equations below

$$\text{lin_acc}_\text{earth} = R \cdot (\text{acc} - \text{grav}),$$ \hspace{1cm} (2a) \hspace{1cm} \text{gyro}_\text{earth} = R \cdot \text{gyro}. \hspace{1cm} (2b)$$

Here, $R$ is the rotation matrix \[45\].

With data from the readings of each sensor, fusion of different sensors, and coordinates transformation, we finally
obtain a 31-dimensional time series that can be arranged in 9 groups. The details are shown in Table 1.

### 4. Motion Recognition Based on MCMR Interclass Distance Features

The sensor data of each motion is a fragment of unrepeated multivariate time series. For this kind of data, we usually apply methods based on distance similarity. Here, in our study, we are facing the multidimensional data from sensor fusion and the problem of matching efficiency. Inspired by shapelet transform in time series recognition, we use distances to samples from different classes to determine the class label of the testing data. Shapelet transform uses Euclidean distances to be the candidates of features for further classification. However, in our cases, this kind of measurement faces the challenge of data offsets and shifting as well as high data dimension and large time consumption. Therefore, we design a fast DTW measurement based on symbolic representation. To eliminate the influence of noise while speeding up the calculation, we use SAX to transform the original time series into shorter symbolic series before distance estimation. We are not sure what kind of sensor and which dimension is more significant in distinguishing different motions, so we use Max-Correlation and Min-Redundancy strategy to select proper candidates to participate in classification of the next step. The chosen features can be used in all kinds of classifiers. It is flexible enough for us to choose suitable classifiers with good performance according to actual situation. The procedure of the recognition solution is demonstrated in Figure 2.

Motion time series recognition based on distance metric uses the distances between time series as a kind of dissimilarity measurement and works in many similar situations, such as the NN-embedded distance and similarity measurement between testing samples and selected motifs. Unlike these methods, we try to use distances between testing motion series and sequences from different classes as values of the classification features and apply in different kinds of classifiers. Since various divergences appear in time and range of the same movement performed by different people as well as the same people in different time, we use DTW based method as the distance estimation. Meanwhile, considering the high calculation time, we reduce the dimension of the data using symbolic representation. The details of this solution will be described in the rest of this section.

Before further discussion, we summarize the notations throughout the paper in Abbreviations section.

With the data obtained in Section 3, we have the following definition of our research object.

**Definition 1** (motion time series based on sensor fusion). $S = \{s_1, s_2, \ldots, s_m\}$ is a multidimensional time series that represents a single movement with $m$-dimensional sensor data. $s_i = \{sd_{i1}, sd_{i2}, \ldots, sd_{in}\}$ indicates the data in $i$th ($1 \leq i \leq m$) dimension with length $n_i$.

In our case, $m$ is 31, and $n$ varies for different motions.

#### 4.1. Data Preprocessing and Representation

Since the same motion with similar shapes might be far away from each other due to the difference of the action range. Data normalization is needed to eliminate this divergence. For the time series in $i$th dimension $s_i$, its $Z$-normalization is shown in the following:

$$s_{i,\text{norm}} = \frac{s_i - \mu_i}{\sigma_i}.$$  \hfill (3)

Here, the mean $\mu_i$ and standard deviation $\sigma_i$ are defined as follows:

$$\mu_i = \frac{\sum sd_{ij}}{n},$$

$$\sigma_i = \sqrt{\frac{\sum (sd_{ij} - \mu_i)^2}{n}}.$$  \hfill (4)

In order to reduce the time consumption while eliminating the influence of noises, we transform the continuous time series into discrete symbolic series using SAX. The time series is divided into fragments of the same size using PAA.
The breakpoints $\beta(\beta_1, \beta_2, \ldots, \beta_k)$ divide the data range into $k + 1$ sections with the same occurrence probability under $N(0, 1)$ Gaussian distribution. All the fragments will then be transformed into corresponding symbols according to the sections they fall into. For a time series $s_i$, if we set $k$ as the number of breakpoints and $f$ as the number of fragments, we are able to obtain a symbolic series $s_{ti}[s_{t_1}, s_{t_2}, \ldots, s_{t_f}]$ where $f < n_i$ and, for each element in $s_{t_j}$, we have $s_{t_j} \in O_1, O_2, \ldots, O_{k+1}$, $1 \leq j \leq f$. $O$ is used to represent a kind of symbol such as letters, integers, or binary code according to different application. This operation is described in Figure 3.

Through this process, we are able to perform dimensional reduction and discretization. Meanwhile, as we need to compare the DTW distances, variation of the time series length may affect the results of the comparison. With the use of SAX, we can unify the length of the series obtained from different motion.

4.2. Feature Extraction Based on Distance Metric. The distance metrics that are commonly used in similarity estimation of time series include symbolic matching distances, different types of traditional distances, and DTW. Generalization of Euclidean distance in SAX representation provides us a lower bound distance based on the discrete symbolic data it generates. Assuming we have two symbolic series $s_{tx}[x_1, x_2, \ldots, x_{\text{low.dim}}]$ and $s_{ty}[y_1, y_2, \ldots, y_{\text{low.dim}}]$, their lower bound distance can be calculated with (5). It is fast and effective in many application scenarios of time series mining, but it is only used in one-dimensional time series:

\[
\text{SAX}_\text{MIN}_\text{DIST}\left(s_{tx}, s_{ty}\right) = \sqrt{\frac{n_x + n_y}{2 \times \text{low.dim}}} \sqrt{\sum_{i=1}^{\text{low.dim}} \min\text{dist}(x_i, y_i)^2}.\]
Figure 4: Demonstration of matching process of DTW: for two time series X and Y, their matching path is restricted in the diagonal belt area on the right, and the matching outside of this area will be ignored. This is a common way to speed up the calculation of this algorithm.

\[
\text{min}\_\text{dist} \text{ is calculated with} \\
\text{min}\_\text{dist} (x_i, y_j) = \begin{cases} 
0, & \text{if } |x_i - y_j| \leq 1 \\
\beta_{\max(x_i,y_j)} - \beta_{\max(x_i,y_j)} & \text{otherwise.}
\end{cases} 
\]  

(6)

\[b_w\] is the band width of the constraint. Normally, Euclidean distances are used to measure the similarity between data points \(x_i\) and \(y_j\) during the matching process as follows:

\[
d(i, j) = \sqrt{\sum_{k=1}^{\text{dim}} (x_{ik} - y_{jk})^2}. 
\]  

(8)

\[
D(i, j) = \begin{cases} 
d(i, j) + \min (D(i-1, j-1), D(i, j-1), D(i-1, j)), & i > 0, j > 0, i - j < b_w \\
d(i, j) + D(i, j-1), & i = 0, 0 < j < b_w \\
d(i, j) + D(i-1, j), & j = 0, 0 < i < b_w \\
0, & i = 0, j = 0.
\end{cases} 
\]  

(7)

4.3. Feature Selection Based on MCMR Strategy. As the selected series are treated as feature candidates, we decide to apply feature selection strategy in finding the most representative ones [47, 48]. In this section, Max-Correlation and Min-Redundancy (MCMR) strategy is proposed to screen out the features that best distinguish different classes and be independent from each other. It mainly consists of two steps: first of all, we find out the feature candidates with strong ability to distinguish samples from different classes; secondly, we need to eliminate the redundant candidates from the chosen ones. In the first step, we introduce the feature selection algorithm ReliefF to estimate the distinguished ability of each feature. By setting a certain threshold \(\eta\) for the number of candidates, we are able to filter out the features with strong correlations to the categories. In the next step,
the candidates with weaker class correlation than their similar candidates are abandoned. Therefore, we need to sort the candidates obtained in the former step in descending order and then successively select the candidates with low correlation to the previous chosen ones until the termination condition is satisfied.

Definition 2 (RF weight: feature weight based on ReliefF). RF weight is a value of a feature candidate that indicates its ability to distinguish different categories from each other. We can compute this weight with ReliefF.

\[ w(F_i) = w(F_i) - \frac{k}{m} \left( \frac{\text{diff}(F_i, R, H_j)}{mk} + \sum_{\text{of class}(R)} \left[ \frac{(p(C)/(1-p(\text{class}(R)))) \sum_{j=1}^{k} \text{diff}(F_i, R, M_j(C))}{mk} \right] \right). \]  

The diff function is defined in

\[ \text{diff}(F_i, R_1, R_2) = \begin{cases} \frac{|R_1(F_i) - R_2(F_i)|}{\max(F_i) - \min(F_i)}, & \text{if } F_i \text{ is continuous} \\ \frac{1}{0}, & \text{if } F_i \text{ is discrete and } R_1(F_i) = R_2(F_i) \\ \frac{1}{1}, & \text{if } F_i \text{ is discrete and } R_1(F_i) \neq R_2(F_i). \end{cases} \]

\( F \) is a collection of feature candidates chosen from labelled data of different classes. \( w(F_i) \) is the RF weight of the \( i \)th feature candidate; \( H_j \) is the \( j \)th nearest hits; \( M_j(C) \) is the \( j \)th nearest misses in class \( C \). \( \max(F_i) \) and \( \min(F_i) \) represent the maximum and minimum values of \( F_i \).

After a certain time of iterative computations, features that can better distinguish samples from different classes will obtain higher weights. However, in the process above, feature candidates are supposed to be independent from each other, and they do not consider the correlation and redundancy between features. Therefore, we need to eliminate the redundant candidates. Here, we first sort the features in descending order based on their RF weights and remove the candidates with weights less than a certain threshold. Then, for the rest of the candidates, we should check whether they are correlated with the chosen ones, successively. If one candidate is supposed to have close relationship with one of the chosen features, it should be abandoned. Otherwise, it can be kept as one of the chosen features. This process will continue until we have collected enough number of features.

In previous studies, various measurements are used to estimate feature redundancy, such as information gain, mutual information, and correlation coefficient. We notice that redundant candidates usually have similar class separability, which means their RF weights have higher linear correlation with each other. Here we use Pearson correlation coefficients based on the RF weights to indicate the correlation between the candidates, which is indicated in the following:

\[ \rho_{ab} = \frac{\sum w_a w_b - \sum w_a \sum w_b / N}{\sqrt{(\sum w_a^2 - (\sum w_a)^2 / N)(\sum w_b^2 - (\sum w_b)^2 / N)}}. \]

We use \( w_a \) to represent the RF weight matrix of a chosen feature and \( w_b \) indicates the RF weight matrix of a candidate. \( \rho_{ab} \) is the Pearson correlation coefficient between a feature that has already been chosen and a candidate. If this correlation coefficient is higher than the threshold, then we should eliminate this candidate.

Pearson correlation coefficient estimates the linear correlation between two variables. Its value is between +1 and −1 inclusive, where 1 is total positive correlation, 0 means no correlation, and −1 is total negative correlation. The actual correlation situation is described in Table 2. According to these corresponding relations, the threshold is easy to define. And this is why we choose this method.

The pseudocode of this process is shown in Algorithm 1. With this procedure, we are able to screen out features with high class correlation and low redundancy. These features can be used to build various kinds of classifiers for motion recognition.

4.4. Motion Classification. In this stage, we can use the selected features to train different kinds of classifiers, such as 1-Nearest Neighbor [50], Bayes Net [51], and Random Forest [52]. We may obtain different results from various classifiers, and we should choose the proper one with better performance in actual application after a few trials. In our study, we apply this solution in dance action practice. The classification effect will be discussed in detail in the next section.

5. Experiments

To verify the efficiency of our solution, we discuss its application in dance practice. The motion data is collected with smartphone and recognize different movements and gestures performed by the wearers. In fact, we found large amount of such requirements as detecting human movements during the practice session in physical education. In our case, we
invite 5 students to perform 6 basic practice movements. Each movement is performed 10 times by each person. We perform 10-fold cross-validation on these data to check the classification results in MATLAB.

In this section, we are going to discuss the following problems:

(i) Feasibility and superiority of the solution.
(ii) Necessity of sensor fusion.
(iii) Parameters selection and adjustment.

More details can be found in the rest of this section.

5.1. Feasibility and Superiority Verification. In order to check the feasibility and superiority of our solution in multivariate motion series recognition, we compare its classification accuracy with the commonly used methods. For multidimensional time series, there are actually two ways to measure the distance based on DTW. We can either calculate the matching cost on each dimension separately or match the multidimensional data points all together in one time. According to [46], this decision should be based on the correlation between these dimensions. As a matter of fact, these two methods have both been applied in related studies. In our solution, we consider the dimensions independently and leave the correlation problem to feature selection process. To check the superiority of our solution, we also try the other ways, including using LB instead of DTW, three kinds of DTW matching without symbolization. In this comparison, we set lower dimension as 50, $k$ as 5, and $\eta$ as 0.5. The accuracies in different classifiers are shown in Figure 5.

At the same time, we should also take a look at the time cost of each solution in training and testing in Figure 6.

As we can see from the results above, measuring DTW distance on different dimensions separately performs better than multidimensional comparison. If we consider the dimensional fusion on each sensor, it will take too much time with ordinary performance. If we take the whole series together with all dimensions to calculate the distance,
the result will be disappointing, though it does not take much time.

It is quite obvious that the symbolic representation would not reduce the classification accuracy while reducing calculation time. In contrast, symbolic representation performs a little bit better than original series using DTW, most of the time. To overcome the time shifting and offsets of the motion data, we apply DTW on the symbolic series. As we expected, this solution obtains the best classification accuracy and takes less time than DTW comparison on the original data. As a matter of fact, lower bound distance of SAX takes the least time in training and testing. And its accuracy is acceptable. Thus, it is also a good trade-off between time and accuracy.

5.2. Discussion of Sensor Fusion. Most of the existing studies would use a single accelerometer to measure the motion state, while some may also consider the readings of gyroscope or magnetometer. Here in our situation, we are able to obtain information from various sensors. How to choose proper amount of data to achieve better performance has become one of our research goal. In one hand, we check the performances with the use of only one kind of sensor data in Figure 7.

From these results, we confirm that accelerometer is the most effective sensor among the existing sensing groups. Most sensors behave better using 1-NN, and a little worse with decision tree. If we observe the results independently, we discover that the original reading from accelerometer containing the influence of gravity performs a little better than pure linear acceleration alone. For gyroscope, the results are improved after the preprocessing such as eliminating the bias. The transformation into Euler angles obtains better performance, but the quaternion does not work so well alone. Gravity and magnetometer obtain mediocre results separately. Meanwhile the transformation into Earth coordinates does not seem to improve much. Among all these results, the best performance comes from the linear acceleration in Earth coordinates, which obtains 97% classification accuracy using 1-NN.

On the other hand, we are going to find out the situation in sensor fusion. Whether it is necessary to use data from different sensors in our situation is one question we need to answer. Therefore, we discuss the performance using the fusion data in Figure 8.

Obviously, the results of sensor fusion are much higher than single sensing data. From the overall performance viewpoint, fusion of all sensor data behaves the best, when fusion of CMDeviceMotion (CMDM) data performs better.
than fusion of four key sensors: accelerometer, gyroscope, gravity sensor, and magnetometer (Ac + Gy + Gr + Ma) in CMDeviceMotion. And the performance of four key sensors in CMDeviceMotion is better than fusion of their raw data in most of the time. The results of accelerometer and gyroscope fusion are relatively worse than the other fusions. Among all these results, we have the best performance coming from fusion of all with 99.33% classification accuracy using 1-NN.

Apparently, with our solution, sensor fusion improves the performance. With more types of sensor data, we would have more information from different points of view. But, at the meantime, we should also consider the time consumption. Our final choice should be based on the resources we have, and the trade-off between time and accuracy might be the best decision.

5.3. Parameter Setting. To obtain better performance, we need to choose proper parameters. And one of the common ways to achieve this goal is to decide through experiments. In our solution, we will discuss three important factors. Firstly, we should mind the choice of target dimension of discrete segmentation. If this size is too small, it will affect the accuracy. If it is too large, it will take too much time. Here, we compare the accuracy in different classifiers with different size of the target dimensions in Figure 9. And the average of the accuracies of different classifiers is shown in Figure 10, which indicates the overall trend of performance with different size of the target dimensions.

Meanwhile, we should also take a look at the time cost of these situations. And the results are shown in Figure 11.

From these results, we can see that for most classifiers, 50 would be a better choice as it takes less time while obtaining fairly well accuracy. Secondly, the threshold of Pearson correlation coefficient \( \eta \) is another parameter we need to observe. According to our intuition, candidates with high correlations to the chosen features might be redundant and should be eliminated. Thus, we should find a threshold to filter out those candidates. As we have mentioned above, this correlation coefficient is defined in a small range, and we can find out the correlation situation based on the interval this value falls in. Now, let us see where we should put this...
threshold. Since the range of this value is quite narrow, we can easily cover the whole positive range. The performance in different situations is shown in Figures 12 and 13.

According to this result, we learn that medium correlations around 0.5 work for our case. This fact actually accords with logic and our expectation. Finally, we take a look at the choice of $k$ in RF weight calculation, which represents the number of nearest candidates we need to consider in different classes. It is commonly believed that the nearest few samples play a significant role in estimating class discrimination ability as they are easier to be confused. So, we check the performance with different choices of $k$, and the results are shown in Figures 14 and 15.

As we can see from the results, no significant change appears in situations with different $k$, only a slight decline in the trend of average accuracy. That means the performance is insensitive to $k$ in this range. It takes less energy in setting this parameter.

6. Conclusion and Future Work

In this paper, we propose a flexible and efficient solution for motion recognition. This solution measures the human movements and gestures using smart phone embedded sensors and applies Max-Correlation and Min-Redundancy strategy to select proper features based on interclass symbolic distance metric. In order to decide what kinds of sensors are needed, we test the effect of different sensors individually as well as combinations of various sensors. The final choice should be a trade-off between time and accuracy based on the requirement and resources. If we pay more attention to efficiency, we can choose less sensor data with accuracy around 94% in average, whereas we can obtain about 98% accuracy using more sensor information. In general, fusion
of a few key sensors would be a better option in most cases. In our situation, we believe fusion of accelerometer and gyroscope with 6 dimensions reaching accuracies around 95.67% in average and 98.33% at best would be a better trade-off between time and accuracy. In data recognition phase, the symbolic representation is able to reduce the dimension of the time series that speeds up the calculation while eliminating the noise and random disturbances. Dynamic time warping on this low dimensional discrete data overcomes the influences of data offsets and time shifting and, in the meantime, improves the classification accuracy. To find out the significant distance measurements and use them as classification features, we define RF weights based on ReliefF algorithm to estimate their class correlation and remove the redundant ones based on Pearson correlation coefficient. This solution is proved to be quite effective in real world application of motion detection in dance practice. In fact, the methods we used in the structure of our solution can be replaced with other related methods based on actual needs, which is quite flexible.

This study is mainly a preliminary exploration of the feasibility of the method. Our future job is to serve more applications by training our solution for more types of movements in dance classes as well as other disciplines such as martial art and gymnastics.

**Abbreviations**

| Symbol | Description |
|--------|-------------|
| $S$ | A multidimensional motion time series |
| $s_i$ | Time series on $i$th dimension of $S$ |
| $m$ | Number of dimensions of the sensor data |
| $n_i$ | Length of the time series in $i$th dimension |
| $s_{i,norm}$ | Z-normalization result of $s_i$ |
| $\mu_s, \sigma_s$ | Mean and standard deviation of $s_i$ |
| $\beta$ | Breakpoints of SAX |
| $s_t$ | Symbolic representation of $s_i$ |
| $O$ | Symbol set of representation |
| low_dim | Length of the symbolic series |
| $D(i, j)$ | DTW matching cost at $(i, j)$ |
| $d(i, j)$ | Distance between data points of different time series |
| DTW$_i$ | DTW matching on each dimensions independently |
| DTW$_D$ | DTW matching on multidimensional data |
| DOF | Degree of freedom |
| MCMR | Max-Correlation and Min-Redundancy |
| $w(F_i)$ | RF weight of feature $F_i$ |
| $\rho_{a,b}$ | Pearson correlation coefficients between feature $a$ and $b$ |
| $k$ | Number of nearest samples chosen in ReliefF |
| $\eta$ | Pearson correlation coefficients threshold for redundancy determination |

**Competing Interests**

The authors declare that there are no competing interests regarding the publication of this paper.

**Acknowledgments**

This research is sponsored by National Natural Science Foundation of China (nos. 61171014, 61371885, 61401029, 61472044, 61472403, and 61571049) and the Fundamental Research Funds for the Central Universities (nos. 2014KJJC32 and 2013NT57) and by SRF for ROCS, SEM, and by the Youth Talents Project of Beijing (YETP1711).

**References**

[1] S.-R. Ke, H. L. U. Thuc, Y.-J. Lee, J.-N. Hwang, J.-H. Yoo, and K.-H. Choi, “A review on video-based human activity recognition,” Computers, vol. 2, no. 2, pp. 88–131, 2013.

[2] H.-T. Pham, J.-J. Kim, T. L. Nguyen, and Y. Won, “3D motion matching algorithm using signature feature descriptor,” Multimedia Tools and Applications, vol. 74, no. 3, pp. 1125–1136, 2015.

[3] R. Slama, H. Wannous, and M. Daoudi, “3D human motion analysis framework for shape similarity and retrieval,” Image and Vision Computing, vol. 32, no. 2, pp. 131–154, 2014.

[4] A. Keerthikha and R. Ganesan, “Pervasive health care system for monitoring oxygen saturation using pulse oximeter sensor,” in Proceedings of the IEEE Conference on Information & Communication Technologies (ICT ’13), pp. 819–822, IEEE, April 2013.

[5] A. Parate, M.-C. Chiu, C. Chadowitz, D. Ganesan, and E. Kalogerakis, “RisQ: recognizing smoking gestures with inertial sensors on a wristband,” in Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys ’14), pp. 149–161, ACM, Bretton Woods, NH, USA, June 2014.

[6] S. K. Tasoulis, C. N. Doukas, V. P. Plagianakos, and I. Maglogiannis, “Statistical data mining of streaming motion data for activity and fall recognition in assistive environments,” Neurocomputing, vol. 105, pp. 87–96, 2013.

[7] A. Ahmadi, E. Mitchell, F. Destelle et al., “Automatic activity classification and movement assessment during a sports training session using wearable inertial sensors,” in Proceedings of the 11th International Conference on Wearable and Implantable Body Sensor Networks (BSN ’14), pp. 98–103, IEEE, Zürich, Switzerland, June 2014.

[8] J. Chardonnens, J. Favre, F. Cuenget, G. Gremion, and K. Aminian, “Measurement of the dynamics in ski jumping using a wearable inertial sensor-based system,” Journal of Sports Sciences, vol. 32, no. 6, pp. 591–600, 2014.

[9] https://en.wikipedia.org/wiki/Microelectromechanical_systems

[10] S. Althloothi, M. H. Mahoor, X. Zhang, and R. M. Voyles, “Human activity recognition using multi-features and multiple kernel learning,” Pattern Recognition, vol. 47, no. 5, pp. 1800–1812, 2014.

[11] Ö. D. Lara and M. A. Labrador, “A survey on human activity recognition using wearable sensors,” IEEE Communications Surveys and Tutorials, vol. 15, no. 3, pp. 1192–1209, 2013.

[12] M.-C. Liu, L. Yu, and Q. Yuan, “Hand gesture detection,” US Patent 8,792,722, 2014.

[13] T. Park, J. Lee, I. Hwang, C. Yoo, L. Nachman, and J. Song, “E-gesture: a collaborative architecture for energy-efficient gesture
recognition with hand-worn sensor and mobile devices,” in Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems, pp. 260–273, ACM, Seattle, Wash, USA, November 2011.

[14] M. Mubashir, L. Shao, and L. Seed, “A survey on fall detection: principles and approaches,” Neurocomputing, vol. 100, pp. 144–152, 2013.

[15] K. Niazmand, K. Ton, A. Kalaras et al., “A measurement device for motion analysis of patients with parkinson’s disease using sensor based smart clothes,” in Proceedings of the 5th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth ’11), pp. 9–16, Dublin, Ireland, May 2011.

[16] M. Field, D. Stirling, Z. Pan, M. Ros, and F. Naghdy, “Recognizing human motions through mixture modeling of inertial data,” Pattern Recognition, vol. 48, no. 8, pp. 2394–2406, 2015.

[17] A. Yurtman and B. Barshan, “Automated evaluation of physical therapy exercises using multi-template dynamic time warping on wearable sensor signals,” Computer Methods and Programs in Biomedicine, vol. 117, no. 2, pp. 189–207, 2014.

[18] D. Fuentes, L. Gonzalez-Abril, C. Angulo, and J. A. Ortega, “Online motion recognition using an accelerometer in a mobile device,” Expert Systems with Applications, vol. 39, no. 3, pp. 2461–2465, 2012.

[19] Y.-L. Hsu, C.-L. Chu, Y.-J. Tsai, and J.-S. Wang, “An inertial pen with dynamic time warping recognizer for handwriting and gesture recognition,” IEEE Sensors Journal, vol. 15, no. 1, pp. 154–163, 2015.

[20] M. Shaal, S. Bosch, O. Durmaz Incel, H. Scholten, and P. J. M. Havinga, “Fusion of smartphone motion sensors for physical activity recognition,” Sensors, vol. 14, no. 6, pp. 10146–10176, 2014.

[21] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, “Activity recognition using cell phone accelerometers,” ACM SIGKDD Explorations Newsletter, vol. 12, no. 2, pp. 74–82, 2011.

[22] Z. Shao and Y. Li, “Integral invariants for space motion trajectory matching and recognition,” Pattern Recognition, vol. 48, no. 8, pp. 2418–2432, 2015.

[23] H. Shan and Y. Liu, “Feature recognition of body dance motion in sports dancing,” Metallurgical & Mining Industry, vol. 7, no. 7, pp. 290–297, 2015.

[24] F. Petitjean, G. Forester, G. I. Webb, A. E. Nicholson, Y. Chen, and E. Keogh, “Dynamic time warping averaging of time series allows faster and more accurate classification,” in Proceedings of the 14th IEEE International Conference on Data Mining (ICDM ’14), pp. 470–479, Shenzhen, China, December 2014.

[25] F. Petitjean, A. Ketterlin, and P. Gançarski, “A global averaging method for dynamic time warping, with applications to clustering,” Pattern Recognition, vol. 44, no. 3, pp. 678–693, 2011.

[26] E. Fuchs, T. Gruber, J. Nitschke, and B. Sick, “On-line motif detection in time series with SwiftMotif,” Pattern Recognition, vol. 42, no. 11, pp. 3015–3031, 2009.

[27] Y. Yang, Z. Cai, J. Li, and G. Lin, “A model-free and stable gene selection in microarray data analysis,” in Proceedings of the 5th IEEE Symposium on Bioinformatics and Bioengineering (BIBE ’05), pp. 3–10, Minneapolis, Minn, USA, October 2005.

[28] B. Hartmann, I. Schwab, and N. Link, “Prototype optimization for temporarily and spatially distorted time series,” in Proceedings of the AAAI Spring Symposium Series, Palo Alto, Calif, USA, March 2010.

[29] K. Yang, Z. Cai, J. Li, and G. Lin, “A stable gene selection in microarray data analysis,” BMC Bioinformatics, vol. 7, article 228, 2006.

[30] H. Wu, Z. Miao, Y. Wang, and M. Lin, “Optimized recognition with few instances based on semantic distance,” The Visual Computer, vol. 31, no. 4, pp. 367–375, 2015.

[31] H. Wu, Z. Miao, Y. Wang, J. Chen, C. Ma, and T. Zhou, “Image completion with multi-image based on entropy reduction,” Neurocomputing, vol. 159, no. 1, pp. 157–171, 2015.

[32] B. Hartmann and N. Link, “Gesture recognition with inertial sensors and optimized DTW prototypes,” in Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC ’10), pp. 2012–2019, Istanbul, Turkey, October 2010.

[33] D. J. Berndt and J. Clifford, “Using dynamic time warping to find patterns in time series,” in Proceedings of the Knowledge Discovery in Databases (KDD ‘94), vol. 10, no. 16, pp. 359–370, Seattle, Wash, USA, 1994.

[34] H. Sakoe and S. Chiba, “A dynamic programming approach to continuous speech recognition,” in Proceedings of the 7th International Congress on Acoustics, vol. 3, pp. 65–69, Budapest, Hungary, August 1971.

[35] F. Itakura, “Minimum prediction residual principle applied to speech recognition,” IEEE Transactions on Acoustics, Speech and Signal Processing, vol. 23, no. 1, pp. 67–72, 1975.

[36] H. Sakoe and S. Chiba, “Dynamic programming algorithm optimization for spoken word recognition,” IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 26, no. 1, pp. 43–49, 1978.

[37] D. Gong, G. Medioni, and X. Zhao, “Structured time series analysis for human action segmentation and recognition,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 36, no. 7, pp. 1414–1427, 2014.

[38] P.-F. Marteau, “Time warp edit distance with stiffness adjustment for time series matching,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 2, pp. 306–318, 2009.

[39] J. Lin, E. Keogh, S. Lonardi, and B. Chiu, “A symbolic representation of time series, with implications for streaming algorithms,” in Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD ’03), pp. 2–11, ACM, 2003.

[40] A. Abdulla-Al-Maruf, H.-H. Huang, and K. Kawagoe, “Time series classification method based on longest common subsequence and textual approximation,” in Proceedings of the 7th International Conference on Digital Information Management (ICDIM ’12), pp. 130–137, IEEE, August 2012.

[41] L. Ye and E. Keogh, “Time series shapelets: a new primitive for data mining,” in Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD ’09), pp. 947–956, ACM, July 2009.

[42] E. Keogh and C. A. Ratanamahatana, “Exact indexing of dynamic time warping,” Knowledge and Information Systems, vol. 7, no. 3, pp. 358–386, 2005.

[43] J. Lines, L. M. Davis, J. Hills, and A. Bagnall, “A shapelet transform for time series classification,” in Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD ’12), pp. 289–297, Beijing, China, August 2012.

[44] J. Hills, J. Lines, E. Baranauskas, J. Mapp, and A. Bagnall, “Classification of time series by shapelet transformation,” Data Mining and Knowledge Discovery, vol. 28, no. 4, pp. 851–881, 2014.
[45] http://developer.android.com/reference/packages.html.

[46] M. Shokoohi-Yekta, J. Wang, and E. Keogh, ”On the non-trivial generalization of dynamic time warping to the multi-dimensional case,” in Proceedings of the SIAM International Conference on Data Mining, pp. 39–48, Vancouver, Canada, April 2015.

[47] Z. Cai, R. Goebel, M. R. Salavatipour, and G. Lin, ”Selecting dissimilar genes for multi-class classification, an application in cancer subtyping,” BMC Bioinformatics, vol. 8, article 206, 2007.

[48] G. Lin, Z. Cai, J. Wu, X.-F. Wan, L. Xu, and R. Goebel, ”Identifying a few foot-and-mouth disease virus signature nucleotide strings for computational genotyping,” BMC Bioinformatics, vol. 9, no. 1, article 279, 2008.

[49] M. Robnik-Šikonja and I. Kononenko, ”Theoretical and empirical analysis of relief and relieff,” Machine Learning, vol. 53, no. 1-2, pp. 23–69, 2003.

[50] N. S. Altman, ”An introduction to kernel and nearest-neighbor nonparametric regression,” The American Statistician, vol. 46, no. 3, pp. 175–185, 1992.

[51] S. Lee and S. Shimoji, ”Bayesnet: bayesian classification network based on biased random competition using gaussian kernels,” in Proceedings of the IEEE International Conference on Neural Networks, vol. 3, pp. 1354–1359, San Francisco, Calif, USA, April 1993.

[52] L. Breiman, ”Random forests,” Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.
