Comparative study on classifying human activities with miniature inertial and magnetic sensors

3 authors, including:

Kerem Altun
Istanbul Kemerburgaz University
17 PUBLICATIONS   **238** CITATIONS

SEE PROFILE

Billur Barshan
Bilkent University
140 PUBLICATIONS   **2,558** CITATIONS

SEE PROFILE

Comparative study on classifying human activities with miniature inertial and magnetic sensors

Kerem Altun, Billur Barshan *, Orkun Tunçel

Department of Electrical and Electronics Engineering, Bilkent University, Bilkent, TR-06800 Ankara, Turkey

ARTICLE INFO

ABSTRACT

This paper provides a comparative study on the different techniques of classifying human activities that are performed using body-worn miniature inertial and magnetic sensors. The classification techniques implemented and compared in this study are: Bayesian decision making (BDM), a rule-based algorithm (RBA) or decision tree, the least-squares method (LSM), the *k*-nearest neighbor algorithm (*k*-NN), dynamic time warping (DTW), support vector machines (SVM), and artificial neural networks (ANN). Human activities are classified using five sensor units worn on the chest, the arms, and the legs. Each sensor unit comprises a tri-axial gyroscope, a tri-axial accelerometer, and a tri-axial magnetometer. A feature set extracted from the raw sensor data using principal component analysis (PCA) is used in the classification process. A performance comparison of the classification techniques is provided in terms of their correct differentiation rates, confusion matrices, and computational cost, as well as their pre-processing, training, and storage requirements. Three different cross-validation techniques are employed to validate the classifiers. The results indicate that in general, BDM results in the highest correct classification rate with relatively small computational cost.

© 2010 Elsevier Ltd. All rights reserved.

1. Introduction

Inertial sensors are self-contained, nonradiating, nonjammable, dead-reckoning devices that provide dynamic motion information through direct measurements. Gyroscopes provide angular *rate* information around an axis of sensitivity, whereas accelerometers provide linear or angular velocity rate information.

For several decades, inertial sensors have been used for navigation of aircraft [1,2], ships, land vehicles, and robots [3–5], for state estimation and dynamic modeling of legged robots [6,7], for shock and vibration analysis in the automotive industry, and in telesurgery [8,9]. Recently, the size, weight, and cost of commercially available inertial sensors have decreased considerably with the rapid development of micro electro-mechanical systems (MEMS) [10]. Some of these devices are sensitive around a single axis; others are multi-axial (usually two- or three-axial). The availability of such MEMS sensors has opened

up new possibilities for the use of inertial sensors, one of them being human activity monitoring, recognition, and classification through body-worn sensors [11–15]. This in turn has a broad range of potential applications in biomechanics [15,16], ergonomics [17], remote monitoring of the physically or mentally disabled, the elderly, and children [18], detecting and classifying falls [19–21], medical diagnosis and treatment [22], home-based rehabilitation and physical therapy [23], sports science [24], ballet and other forms of dance [25], animation and film making, computer games [26,27], professional simulators, virtual reality, and stabilization of equipment through motion compensation.

Early studies in activity recognition employed vision-based systems with single or multiple video cameras, and this remains the most common approach to date [28–31]. For example, although the gesture recognition problem has been well studied in computer vision [32], much less research has been done in this area with body-worn inertial sensors [33,34]. The use of camera systems may be acceptable and practical when activities are confined to a limited area such as certain parts of a house or office environment and when the environment is well lit. However, when the activity involves going from place to place, camera

systems are much less convenient. Furthermore, camera systems interfere considerably with privacy, may supply additional, unneeded information, and cause the subjects to act unnaturally.

Miniature inertial sensors can be flexibly used inside or behind objects without occlusion effects. This is a major advantage over visual motion-capture systems that require a free line of sight. When a single camera is used, the 3-D scene is projected onto a 2-D one, with significant information loss. Points of interest are frequently pre-identified by placing special, visible markers such as light-emitting diodes (LEDs) on the human body. Occlusion or shadowing of points of interest (by human body parts or objects in the surroundings) is circumvented by positioning multiple camera systems in the environment and using several 2-D projections to reconstruct the 3-D scene. This requires each camera to be separately calibrated. Another major disadvantage of using camera systems is that the cost of processing and storing images and video recordings is much higher than those of 1-D signals. 1-D signals acquired from multiple axes of inertial sensors can directly provide the required information in 3-D. Unlike high-end commercial inertial sensors that are calibrated by the manufacturer, in low-cost applications that utilize these devices, calibration is still a necessary procedure. Accelerometer-based systems are more commonly adopted than gyros because accelerometers are easily calibrated by gravity, whereas gyro calibration requires an accurate variable-speed turntable and is more complicated.

The use of camera systems and inertial sensors are two inherently different approaches that are by no means exclusive and can be used in a complementary fashion in many situations. In a number of studies, video cameras are used only as a reference for comparison with inertial sensor data [35–40]. In other studies, data from these two sensing modalities are integrated or fused [41,42]. The fusion of visual and inertial data has attracted considerable attention recently because of its robust performance and potentially wide applications [43,44]. Fusing the data of inertial sensors and magnetometers is also reported in the literature [38,46,47].

Previous work on activity recognition based on body-worn inertial sensors is fragmented, of limited scope, and mostly unsystematic in nature. Due to the lack of a common ground among different researchers, results published so far are difficult to compare, synthesize, and build upon in a manner that allows broad conclusions to be reached. A unified and systematic treatment of the subject is desirable; theoretical models need to be developed that will enable studies designed such that the obtained results can be synthesized into a larger whole.

Most previous studies distinguish between sitting, lying, and standing [18,35–37,39,45,48–50], as these postures are relatively easy to detect using the static component of acceleration. Distinguishing between walking, and ascending and descending stairs has also been accomplished [45,48,50], although not as successfully as detecting postures. The signal processing and motion detection techniques employed, and the configuration, number, and type of sensors differ widely among the studies, from using a single accelerometer [18,51,52] to as many as 12 [53] on different parts of the body. Although gyroscopes can provide valuable rotational information in 3-D, in most studies, accelerometers are preferred to gyroscopes due to their ease of calibration. To the best of our knowledge, guidance on finding a suitable configuration, number, and type of sensors does not exist [45]. Usually, some configuration and some modality of sensors is chosen without strong justification, and empirical results are presented. Processing the acquired signals is also often done ad hoc and with relatively unsophisticated techniques.

In this work, we use miniature inertial sensors and magnetometers positioned on different parts of the body to classify human activities. The motivation behind investigating activity

classification is its potential applications in the many different areas mentioned above. The main contribution of this paper is that unlike previous studies, we use many redundant sensors to begin with and extract a variety of features from the sensor signals. Then, we use an unsupervised feature transformation technique that allows considerable feature reduction through automatic selection of the most informative features. We provide an extensive and systematic comparison between various classification techniques used for human activity recognition based on the same data set. We compare the successful differentiation rates, confusion matrices, and computational requirements of the techniques.

The paper is organized as follows: In Section 2, we introduce the activities classified in this study and outline the experimental methodology. Describing the feature vectors and the feature reduction process is the topic of Section 3. In Section 4, we briefly review the classification methods used in this study. In Section 5, we present the experimental results and compare the methods' computational requirements. We also provide a brief discussion on selecting classification techniques and their advantages and disadvantages. Section 6 addresses the potential application areas of miniature inertial sensors in activity recognition. In Section 7, we draw conclusions and provide possible directions for future work.

2. Classified activities and experimental methodology

The 19 activities that are classified using body-worn miniature inertial sensor units are: sitting (A1), standing (A2), lying on back and on right side (A3 and A4), ascending and descending stairs (A5 and A6), standing in an elevator still (A7) and moving around (A8), walking in a parking lot (A9), walking on a treadmill with a speed of 4 km/h (in flat and 15° inclined positions) (A10 and A11), running on a treadmill with a speed of 8 km/h (A12), exercising on a stepper (A13), exercising on a cross trainer (A14), cycling on an exercise bike in horizontal and vertical positions (A15 and A16), rowing (A17), jumping (A18), and playing basketball (A19).

Five MTx 3-DOF orientation trackers (Fig. 1) are used, manufactured by Xsens Technologies [54]. Each MTx unit has a tri-axial accelerometer, a tri-axial gyroscope, and a tri-axial magnetometer, so the sensor units acquire 3-D acceleration, rate of turn, and the strength of Earth's magnetic field. Each motion tracker is programmed via an interface program called MT Manager to capture the raw or calibrated data with a sampling frequency of up to 512 Hz.

Accelerometers of two of the MTx trackers can sense up to $\pm 5g$ and the other three can sense in the range of $\pm 18g$, where $g = 9.80665$ m/s^2 is the gravitational constant. All gyroscopes in the MTx unit can sense in the range of $\pm 1200°$/s angular velocities; magnetometers can sense magnetic fields in the range of $\pm 75\,\mu$T. We use all three types of sensor data in all three dimensions.
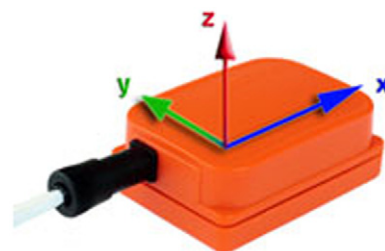

Fig. 1. MTx 3-DOF–orientation tracker (reprinted from http://www.xsens.com/en/general/mtx).

The sensors are placed on five different places on the subject's body as depicted in Fig. 2. Since leg motions in general may produce larger accelerations, two of the $\pm 18g$ sensor units are placed on the sides of the knees (right side of the right knee and left side of the left knee), the remaining $\pm 18g$ unit is placed on the subject's chest (Fig. 2(b)), and the two $\pm 5g$ units on the wrists (Fig. 2(c)).

The five MTx units are connected with 1 m cables to a device called the Xbus Master, which is attached to the subject's belt. The Xbus Master transmits data from the five MTx units to the receiver using a Bluetooth^{TM} connection. The Xbus Master, which is connected to three MTx orientation trackers, can be seen in Fig. 3(a). The receiver is connected to a laptop computer via a USB port. Two of the five MTx units are directly connected to the Xbus Master and the remaining three units are indirectly connected to the Xbus Master by wires to the other two. Fig. 3(b) illustrates the connection configuration of the five MTx units and the Xbus Master.

Each activity listed above is performed by eight different subjects (4 female, 4 male, between the ages 20 and 30) for 5 min. The subjects are asked to perform the activities in their own style and were not restricted on how the activities should be performed. For this reason, there are inter-subject variations in the speeds and amplitudes of some activities. The activities are performed at the Bilkent University Sports Hall, in the Electrical and Electronics Engineering Building, and in a flat outdoor area on campus. Sensor units are calibrated to acquire data at 25 Hz sampling frequency. The 5-min signals are divided into 5-s segments, from which certain features are extracted. In this way, $480 (=60 \times 8)$ signal segments are obtained for each activity.

3. Feature extraction and reduction

After acquiring the signals as described above, we obtain a discrete-time sequence of N_s elements that can be represented as an $N_s \times 1$ vector $\mathbf{s} = [s_1, s_2, \ldots, s_{Ns}]^T$. For the 5-s time windows and the 25-Hz sampling rate, $N_s = 125$. The initial set of features we use before feature reduction are the minimum and maximum values, the mean value, variance, skewness, kurtosis, autocorrelation sequence, and the peaks of the discrete Fourier transform (DFT) of $\mathbf{s}$ with the corresponding frequencies. These are calculated as follows:

$$\text{mean}(\mathbf{s}) = \mu_\mathbf{s} = E\{\mathbf{s}\} = \frac{1}{N_s} \sum_{i=1}^{N_s} s_i$$

$$\text{variance}(\mathbf{s}) = \sigma^2 = E\{(\mathbf{s} - \mu_\mathbf{s})^2\} = \frac{1}{N_s} \sum_{i=1}^{N_s} (s_i - \mu_\mathbf{s})^2$$

$$\text{skewness}(\mathbf{s}) = \frac{E\{(\mathbf{s} - \mu_\mathbf{s})^3\}}{\sigma^3} = \frac{1}{N_s \sigma^3} \sum_{i=1}^{N_s} (s_i - \mu_\mathbf{s})^3$$

$$\text{kurtosis}(\mathbf{s}) = \frac{E\{(\mathbf{s} - \mu_\mathbf{s})^4\}}{\sigma^4} = \frac{1}{N_s \sigma^4} \sum_{i=1}^{N_s} (s_i - \mu_\mathbf{s})^4$$

$$\text{autocorrelation} : R_{\mathbf{ss}}(\Delta) = \frac{1}{N_s - \Delta} \sum_{i=0}^{N_s - \Delta - 1} (s_i - \mu_\mathbf{s})(s_{i-\Delta} - \mu_\mathbf{s}),$$
$$\Delta = 0, 1, \ldots, N_s - 1$$

$$\text{DFT} : S_{\text{DFT}}(k) = \sum_{i=0}^{N_s - 1} s_i e^{-j 2\pi k i / N_s}, \quad k = 0, 1, \ldots, N_s - 1$$

In these equations, s_i is the ith element of the discrete-time sequence $\mathbf{s}$, $E\{\cdot\}$ denotes the expectation operator, $\mu_\mathbf{s}$ and σ are the mean and the standard deviation of $\mathbf{s}$, $R_{ss}(\Delta)$ is the unbiased autocorrelation sequence of $\mathbf{s}$, and $S_{\text{DFT}}(k)$ is the kth element of the 1-D N_s-point DFT. In calculating the first five features above, it is assumed that the signal segments are the realizations of an ergodic process so that ensemble averages are replaced with time averages. Apart from those listed above, we have also considered using features such as the total energy of the signal, cross-correlation coefficients of two signals, and the discrete cosine transform coefficients of the signal.

Since there are five sensor units (MTx), each with three tri-axial devices, a total of nine signals are recorded from every sensor unit. Different signal representations, such as the time-domain signal, its autocorrelation function, and its DFT for two selected activities are given in Fig. 4. In parts (a) and (c) of the figure, the quasi-periodic nature of the walking signal can be observed.

When a feature such as the mean value of a signal is calculated, $45 (=9 \text{ axes } \times 5 \text{ units})$ different values are available. These values from the five sensor units are placed in the feature vectors in the order of right arm, left arm, right leg, torso, and left leg. For each one of these sensor locations, nine values for each feature are calculated and recorded in the following order: the x, y, z axes' acceleration, the x, y, z axes' rate of turn, and the x, y, z axes' Earth's magnetic field. In constructing the feature vectors, the above procedure is followed for the minimum and maximum values, the mean, skewness, and kurtosis. Thus, $225 (=45 \text{ axes } \times 5 \text{ features})$ elements of the feature vectors are obtained by using the above procedure.

After taking the DFT of each 5-s signal, the maximum five Fourier peaks are selected so that a total of $225 (=9 \text{ axes } \times 5 \text{ units } \times 5 \text{ peaks})$ Fourier peaks are obtained for each segment.
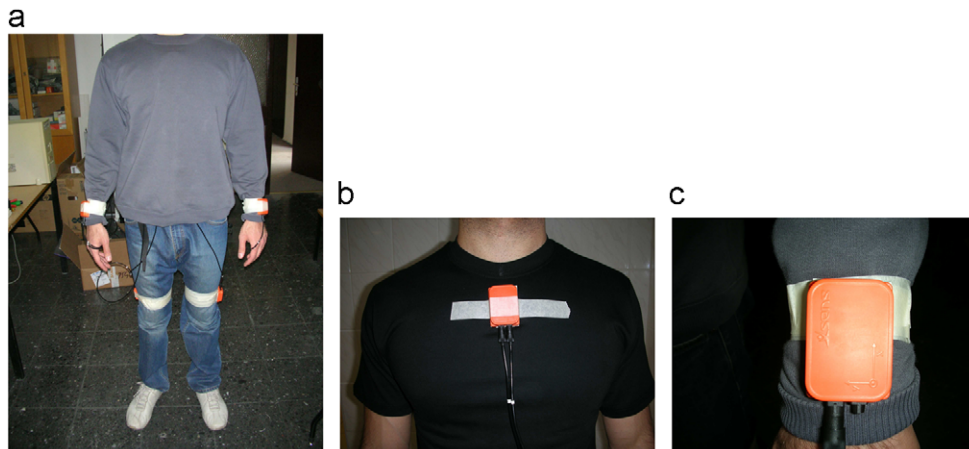
Fig. 2. Positioning of Xsens sensor modules on the body.
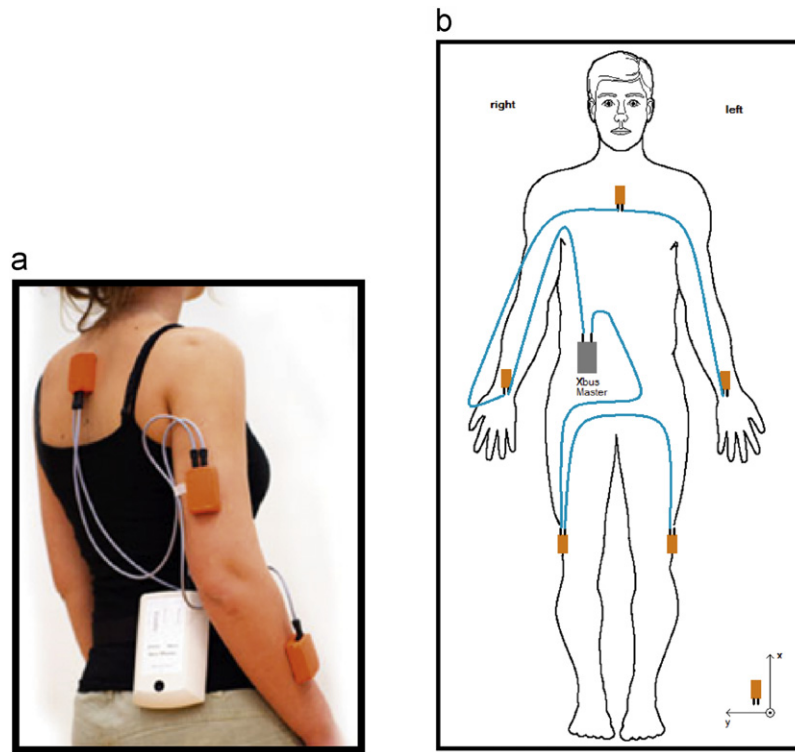
Fig. 3. (a) MTx blocks and Xbus Master (reprinted from http://www.xsens.com/en/movement-science/xbus-kit), (b) connection diagram of MTx sensor blocks (body part of the figure is from http://www.answers.com/body breadths).

Each group of 45 peaks is placed in the order of right arm, left arm, right leg, torso, and left leg, as above. The 225 frequency values that correspond to these Fourier peaks are placed after the Fourier peaks in the same order.

Eleven autocorrelation samples are placed in the feature vectors for each axis of each sensor, following the order given above. Since there are 45 distinct sensor signals, 495 ($=45$ axes $\times 11$ samples) autocorrelation samples are placed in each feature vector. The first sample of the autocorrelation function (the variance) and every fifth sample up to the fiftieth are placed in the feature vectors for each signal.

As a result of the above feature extraction process, a total of 1170 ($=225+225+225+495$) features are obtained for each of the 5-s signal segments so that the dimensions of the resulting feature vectors are 1170×1. All features are normalized to the interval [0,1] so as to be used for classification.

Because the initial set of features was quite large (1170) and not all features were equally useful in discriminating between the activities, we investigated different feature selection and reduction methods [55]. In this work, we reduced the number of features from 1170 to 30 through principal component analysis (PCA) [56], which is a transformation that finds the optimal linear combinations of the features, in the sense that they represent the data with the highest variance in a feature subspace, without taking the intra-class and inter-class variances into consideration separately. The reduced dimension of the feature vectors is determined by observing the eigenvalues of the covariance matrix of the 1170×1 feature vectors, sorted in Fig. 5(a) in descending order. The 30 eigenvectors corresponding to the largest 30 eigenvalues (Fig. 5(b)) are used to form the transformation matrix, resulting in 30×1 feature vectors. Although the initial set of 1170 features do have physical meaning, because of the matrix transformation involved, the transformed feature vectors cannot be assigned any physical meaning. Scatter plots of the first five transformed features are given in Fig. 6 pairwise. As expected, in the first two plots or so (parts (a) and (b) of the figure), the features for different classes are better clustered and more distinct.

We assume that after feature reduction, the resulting feature vector is an $N \times 1$ vector $\mathbf{x}=[x_1,\dots,x_N]^T$.

4. Classification techniques

The classification techniques used in this study are briefly reviewed in this section. More detailed descriptions can be found in [14,57] and in the given references.

We associate a class ω_i with each activity type ($i=1,\dots,c$). An unknown activity is assigned to class ω_i if its feature vector $\mathbf{x}=[x_1,\dots,x_N]^T$ falls in the region Ω_i. A rule that partitions the decision space into regions $\Omega_i, i=1,\dots,c$ is called a *decision rule*. In our work, each one of these regions corresponds to a different activity type. Boundaries between these regions are called *decision surfaces*. The *training set* contains a total of $I=I_1+I_2+\cdots+I_c$ sample feature vectors where I_i sample feature vectors belong to class ω_i, and $i=1,\dots,c$. The *test set* is then used to evaluate the performance of the decision rule.

4.1. Bayesian decision making (BDM)

In BDM, class conditional probability density functions (CCPDFs) are estimated for each class. In this study, the CCPDFs are assumed to have a multi-variate Gaussian parametric form, and the mean vector and the covariance matrix of the CCPDF for each class are estimated using maximum likelihood estimators on the training vectors. For a given test vector $\mathbf{x}$, the maximum a posteriori (MAP) decision rule is used for classification [56].
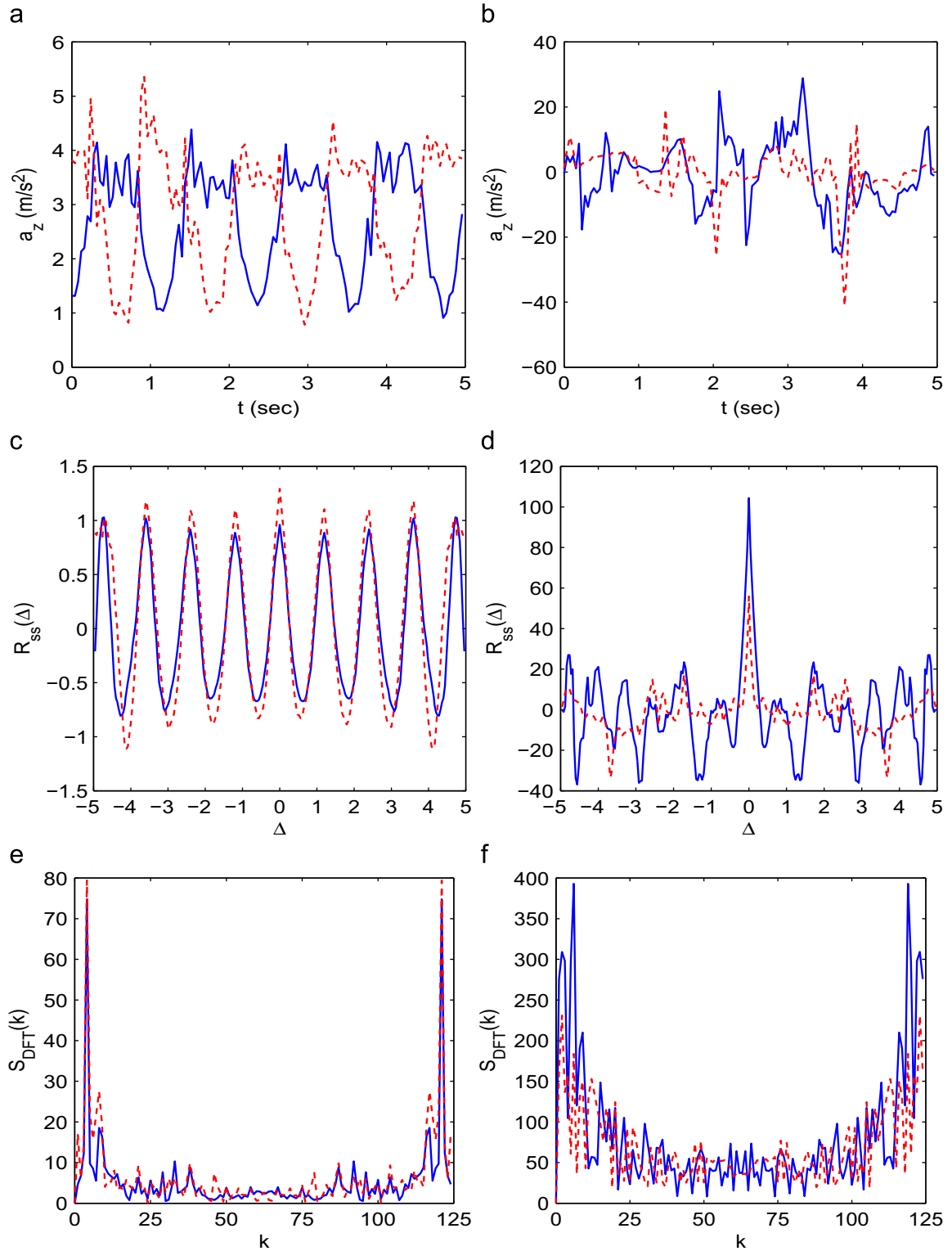
Fig. 4. (Color online) (a) and (b): Time-domain signals for walking and basketball, respectively; z-axis acceleration of the right (solid lines) and left arm (dashed lines) are given; (c) and (d): autocorrelation functions of the signals in (a) and (b); (e) and (f): 125-point DFT of the signals in (a) and (b), respectively.

4.2. Rule-based algorithm (RBA)

A rule-based algorithm or a decision tree can be considered a sequential procedure that classifies given inputs. An RBA follows predefined rules at each node of the tree and makes binary decisions based on these rules. Rules correspond to conditions such as "is feature $x_i \leq \tau_i$?," where τ is the threshold value for a given feature and $i = 1, 2, \ldots, T$, with T being the total number of features used [58].

As the information necessary to differentiate between the activities is completely embodied in the decision rules, the RBA has the advantage of not requiring storage of any reference
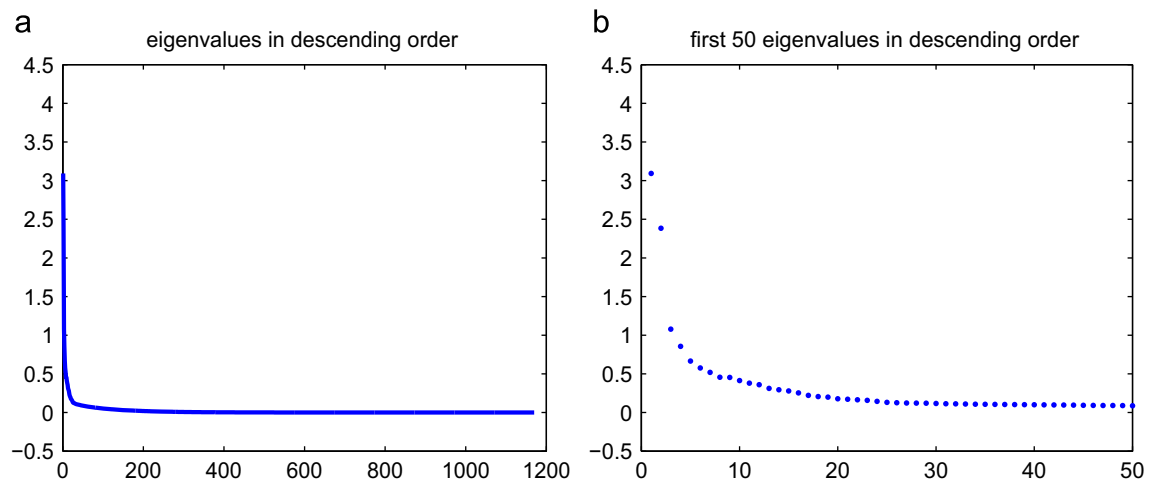
Fig. 5. (a) All eigenvalues (1170) and (b) the first 50 eigenvalues of the covariance matrix sorted in descending order.
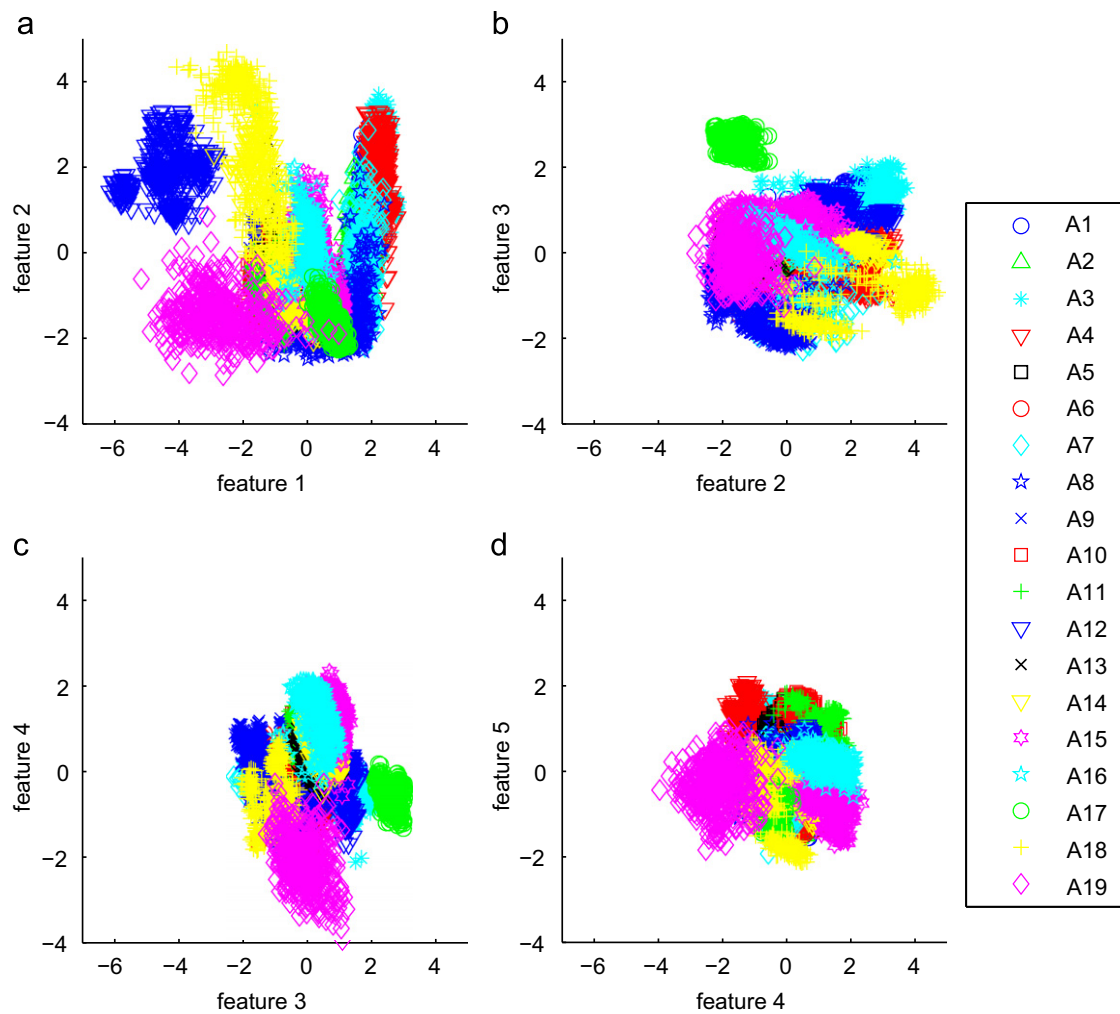
Fig. 6. (Color online) Scatter plots of the first five features selected by PCA.

feature vectors. The main difficulty is in designing the rules and making them independent of absolute quantities so that they will be more robust and generally applicable.

In this study, we automatically generate a binary decision tree based on the training data using the CART algorithm [59]. Given a set of training vectors along with their class labels, a binary tree,

and a decision rule for each node of the tree, each node corresponds to a particular subset of the training vectors where each element of that subset satisfies the conditions imposed by the ancestors of that node. Thus, a decision at a node splits the corresponding subset into two: those that satisfy the condition and those that do not. Naturally, the ideal split is expected to isolate a class from others at each decision node. Since this is not the case in practice, a decision rule is found by searching among all possible decisions that minimize the *impurity* of that node. We use entropy as a measure of impurity, and the class frequencies at each node to estimate the entropy [59]. Test vectors are then used to evaluate the classification performance of the decision tree.

4.3. Least-squares method (LSM)

In LSM, the average reference vector for each class is calculated as a representative for that particular class. Each test vector is compared with the average reference vector (instead of each individual reference vector) as follows:

$$\mathcal{D}_i^2 = \sum_{n=1}^{N}(x_n - r_{in})^2 = (x_1 - r_{i1})^2 + \cdots + (x_N - r_{iN})^2, \quad i = 1, \ldots, c \qquad (1)$$

The test vector is assigned to the same class as the nearest average reference vector. In this equation, $\mathbf{x} = [x_1, x_2, \ldots, x_N]^T$ represents a test feature vector, $\mathbf{r} = [r_{i1}, r_{i2}, \ldots, r_{iN}]^T$ represents the average of the reference feature vectors for each distinct class, and $\mathcal{D}_i^2$ is the square of the distance between these two vectors.

4.4. k-Nearest neighbor (k-NN) algorithm

In k-NN, the k nearest neighbors of the vector $\mathbf{x}$ in the training set are considered and the vector $\mathbf{x}$ is classified into the same class as the majority of its k nearest neighbors [56]. The Euclidean distance measure is used. The k-NN algorithm is sensitive to the local structure of the data. The selection of the parameter k, the number of neighbors considered, is a very important issue that can affect the decision made by the k-NN classifier. Unfortunately, a pre-defined rule for the selection of the value of k does not exist. In this study, the number of nearest neighbors k is determined experimentally by maximizing the correct classification rate over different k values.

4.5. Dynamic time warping (DTW)

Dynamic time warping is an algorithm for measuring the similarity between two sequences that may vary in time or speed. An optimal match between two given sequences (e.g. a time series) is found under certain restrictions. The sequences are "warped" nonlinearly in the time dimension to determine a measure of their similarity independent of certain nonlinear variations in the time dimension. In DTW, the aim is to find the least-cost warping path for the tested feature vector among the stored reference feature vectors [60] where the cost measure is typically taken as the Euclidean distance between the elements of the feature vectors. DTW is used mostly in automatic speech recognition to handle different speaking speeds [60,61]. Besides speech recognition, DTW has been used in signature and gait recognition, for ECG signal classification, for fingerprint verification, for word spotting in handwritten historical documents on electronic media and machine-printed documents, and for face localization in color images [62,63]. In this study, DTW is used for classifying feature vectors of different activities extracted from the signals of miniature inertial sensors.

4.6. Support vector machines (SVMs)

The support vector machine classifier is a machine learning technique proposed early in the 1980s [64–66]. It has been mostly used in applications such as object, voice, and handwritten character recognition, and in text classification.

If the feature vectors in the original feature space are not linearly separable, SVMs pre-process and represent them in a higher-dimensional space where they can become linearly separable. The dimension of the transformed space may sometimes be much higher than the original feature space. With a suitable nonlinear mapping $\phi(\cdot)$ to a sufficiently high dimension, data from two different classes can always be made linearly separable, and separated by a hyperplane. The choice of the nonlinear mapping method depends on the prior information available to the designer. If information is not available, one might choose to use polynomials, Gaussians, or other types of basis functions. The dimensionality of the mapped space can be arbitrarily high, however, in practice, it may be limited by computational resources. The complexity of SVMs is related to the number of resulting support vectors rather than the high dimensionality of the transformed space.

In this study, the SVM method is applied to differentiate feature vectors that belong to more than two classes (19 classes). Following the one-versus-the-rest method, c different binary classifiers are trained, where each classifier recognizes one of c activity types. A nonlinear classifier with a radial basis function kernel $K(\mathbf{x}, \mathbf{x}_i) = e^{-\gamma |\mathbf{x} - \mathbf{x}_i|^2}$ is used with $\gamma = 4$. A library for SVMs (LIBSVM toolbox) is used in the MATLAB environment [67].

4.7. Artificial neural networks (ANN)

Multi-layer ANNs consist of an input layer, one or more hidden layers to extract progressively more meaningful features, and a single output layer, each composed of a number of units called *neurons*. The model of each neuron includes a smooth nonlinearity, called the *activation function*. Due to the presence of distributed nonlinearity and a high degree of connectivity, theoretical analysis of ANNs is difficult. These networks are trained to compute the boundaries of decision regions in the form of connection weights and biases by using training algorithms. The performance of ANNs is affected by the choice of parameters related to the network structure, training algorithm, and input signals, as well as by parameter initialization [68,69].

In this work, a three-layer ANN is used for classifying human activities. The input layer has N neurons, equal to the dimension of the feature vectors (30). The hidden layer has 12 neurons, and the output layer has c neurons, equal to the number of classes. In the input and hidden layers each, there is an additional neuron with a bias value of 1. For an input feature vector $\mathbf{x} \in \mathbb{R}^N$, the target output is 1 for the class that the vector belongs to, and 0 for all other output neurons. The sigmoid function used as the activation function in the hidden and output layers is given by $g(x) = (1 + e^{-x})^{-1}$.

The output neurons can take continuous values between 0 and 1. Fully connected ANNs are trained with the back-propagation algorithm [68] by presenting a set of *training patterns* to the network. The aim is to minimize the average of the sum of squared errors over all training vectors:

$$\mathcal{E}_{\mathrm{av}}(\mathbf{w}) = \frac{1}{2I}\sum_{i=1}^{I}\sum_{k=1}^{c}[t_{ik} - o_{ik}(\mathbf{w})]^2 \qquad (2)$$

Here, $\mathbf{w}$ is the weight vector, t_{ik} and o_{ik} are the desired and actual output values for the ith training pattern and the kth output neuron, and I is the total number of training patterns. When the

entire training set is covered, an *epoch* is completed. The error between the desired and actual outputs is computed at the end of each iteration and these errors are averaged at the end of each epoch (Eq. (2)). The training process is terminated when a certain precision goal on the average error is reached or if the specified maximum number of epochs (5000) is exceeded, whichever occurs earlier. The latter case occurs very rarely. The acceptable average error level is set to a value of 0.03. The weights are initialized randomly with a uniform distribution in the interval [0,0.2], and the learning rate is chosen as 0.2.

In the test phase, the test feature vectors are fed forward to the network, the outputs are compared with the desired outputs, and the error between them is calculated. The test vector is said to be correctly classified if this error is below a threshold value of 0.25.

5. Experimental results

The classification techniques described in Section 4 are employed to classify the 19 different activities using the 30 features selected by PCA. A total of 9120 ($=60$ feature vectors $\times$ 19 activities $\times$ 8 subjects) feature vectors are available, each containing the 30 reduced features of the 5-s signal segments. In the training and testing phases of the classification methods, we use the repeated random sub-sampling (RRSS), *P*-fold, and leave-one-out (L1O) cross-validation techniques. In RRSS, we divide the 480 feature vectors from each activity type randomly into two sets so that the first set contains 320 feature vectors (40 from each subject) and the second set contains 160 (20 from each subject). Therefore, two-thirds (6080) of the 9120 feature vectors are used for training and one-third (3040) for testing. This is repeated 10 times and the resulting correct differentiation percentages are averaged. The disadvantage of this method is that some observations may never be selected in the testing or the validation phase, whereas others may be selected more than once. In other words, validation subsets may overlap.

In *P*-fold cross validation, the 9120 feature vectors are divided into $P=10$ partitions, where the 912 feature vectors in each partition are selected completely randomly, regardless of the subject or the class they belong to. One of the *P* partitions is retained as the validation set for testing, and the remaining $P-1$ partitions are used for training. The cross-validation process is then repeated *P* times (the folds), where each of the *P* partitions is used exactly once for validation. The *P* results from the folds are then averaged to produce a single estimation. The random partitioning is repeated 10 times and the average correct differentiation percentage is reported. The advantage of this validation method over RRSS is that all feature vectors are used for both training and testing, and each feature vector is used for testing exactly once in each of the 10 runs.

Finally, we also used subject-based L1O cross validation, where the 7980 ($=60$ vectors $\times$ 19 activities $\times$ 7 subjects) feature vectors of seven of the subjects are used for training and the 1140 feature vectors of the remaining subject are used in turn for validation. This is repeated eight times such that the feature vector set of each subject is used once as the validation data. The eight correct classification rates are averaged to produce a single estimate. This is similar to *P*-fold cross validation with *P* being equal to the number of subjects ($P=8$), and where all the feature vectors in the same partition are associated with the same subject.

Correct differentiation rates of the classification techniques over 10 runs and their standard deviations are tabulated in Table 1 for the three cross-validation techniques we considered. With RRSS and *P*-fold cross-validation, all of the correct differentiation rates are above 80%, with standard deviations

Table 1
Correct differentiation rates for all classification methods and three cross-validation techniques.

Method	Correct differentiation rate (%) $\pm$ one standard deviation		
	RRSS	*P*-fold	L1O
BDM	99.1 $\pm$ 0.12	99.2 $\pm$ 0.02	75.8
RBA	81.0 $\pm$ 1.52	84.5 $\pm$ 0.44	53.6
LSM	89.4 $\pm$ 0.75	89.6 $\pm$ 0.10	85.3
k-NN ($k=7$)	98.2 $\pm$ 0.12	98.7 $\pm$ 0.07	86.9
DTW$_1$	82.6 $\pm$ 1.36	83.2 $\pm$ 0.26	80.4
DTW$_2$	98.5 $\pm$ 0.18	98.5 $\pm$ 0.08	85.2
SVM	98.6 $\pm$ 0.12	98.8 $\pm$ 0.03	87.6
ANN	86.9 $\pm$ 3.31	96.2 $\pm$ 0.19	74.3

The results of the RRSS and *P*-fold cross-validation techniques are calculated over 10 runs, whereas those of L1O are over a single run.

usually lower than 0.5% with a few exceptions. From the table, it can be observed that there is not a significant difference between the results of RRSS and *P*-fold cross-validation techniques. The results of subject-based L1O are always lower than the two. In terms of reliability and repeatability, the *P*-fold cross-validation technique results in smaller standard deviations than RRSS. Because L1O cross validation would give the same classification percentage if the complete cycle over the subject-based partitions is repeated, its standard deviation is zero.

Among the classification techniques we considered and implemented, when RRSS and *P*-fold cross-validation is used, BDM gives the highest classification rate, followed by SVM and *k*-NN. RBA and DTW$_1$ perform the worst in general. In subject-based L1O cross validation, SVM is the best, followed by *k*-NN. The correct classification rates reported for L1O cross validation can be interpreted as the expected correct classification rates when data from a new subject are acquired and given as input to the classifiers. The most significant difference in the performances of the different validation methods is observed for the BDM method (Table 1). The RRSS and *P*-fold cross validation result in 99% correct classification rate, suggesting that the data are well represented by a multi-variate Gaussian distribution. However, the 76% correct classification rate of L1O cross validation implies that the parameters of the Gaussian, when calculated by excluding one of the subjects, cannot represent the data of the excluded subject sufficiently well. Thus, if one is to classify the activities of a new test subject whose training data are not available to the classifiers, SVM, *k*-NN, or LSM methods could be used.

We chose to employ the *P*-fold cross-validation technique in reporting the results presented in Tables 2–8. Looking at the confusion matrices of the different techniques, it can be observed that A7 and A8 are the activities most confused with each other. This is because both of these activities are performed in the elevator and the signals recorded from these activities have similar segments. Therefore, confusion at the classification stage becomes inevitable. A2 and A7, A13 and A14, as well as A9, A10, A11, are also confused from time to time for similar reasons. Two activities that are almost never confused are A12 and A17.

The confusion matrices for BDM and RBA are provided in Tables 2 and 3. With these methods, correct differentiation rates of 99.2% and 84.5% are, respectively, achieved. The features used in the RBA correspond to the 30 features selected by PCA and the rules change at every training cycle.

In the LSM approach, test vectors are compared with the average of the reference vectors calculated for each of the 19 activities. The confusion matrix for this method is provided in Table 4. The overall successful differentiation rate of LSM is 89.6%.

Table 2
Confusion matrix for BDM (*P*-fold cross validation, 99.2%).

True	Classified																		
	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19
A1	480	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A2	0	478	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0
A3	0	0	478	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0
A4	0	0	0	480	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A5	0	0	0	0	478	0	0	2	0	0	0	0	0	0	0	0	0	0	0
A6	0	0	0	0	0	477	0	3	0	0	0	0	0	0	0	0	0	0	0
A7	0	0	0	0	0	0	467	13	0	0	0	0	0	0	0	0	0	0	0
A8	0	0	0	0	0	0	44	435	0	0	0	0	0	0	0	0	0	0	1
A9	0	0	0	0	0	0	0	1	479	0	0	0	0	0	0	0	0	0	0
A10	0	0	0	0	0	0	0	0	0	478	2	0	0	0	0	0	0	0	0
A11	0	0	0	0	0	0	0	0	0	0	480	0	0	0	0	0	0	0	0
A12	0	0	0	0	0	0	0	0	0	0	0	480	0	0	0	0	0	0	0
A13	0	0	0	0	0	0	0	0	0	0	0	0	479	1	0	0	0	0	0
A14	0	0	0	0	0	0	0	2	0	0	0	0	0	478	0	0	0	0	0
A15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	480	0	0	0	0
A16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	480	0	0	0
A17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	480	0	0
A18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	480	0
A19	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	476

Table 3
Confusion matrix for RBA (*P*-fold cross validation, 84.5%).

True	Classified																		
	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19
A1	418	6	22	4	0	0	8	2	0	0	1	1	0	5	7	2	2	0	2
A2	9	404	0	1	2	5	37	9	1	0	2	2	1	1	3	2	0	1	0
A3	12	1	439	13	0	0	2	1	1	1	1	2	3	0	2	1	1	0	0
A4	7	3	11	446	4	0	4	0	0	0	0	0	0	0	0	2	1	2	0
A5	0	2	1	1	421	1	4	11	10	4	5	1	1	4	0	6	3	3	2
A6	4	2	1	0	4	409	10	19	7	0	0	0	9	1	7	0	1	5	1
A7	8	33	2	3	1	16	360	48	0	0	0	0	1	0	2	0	1	2	3
A8	2	17	1	2	21	31	60	266	10	4	4	1	13	7	7	3	8	7	16
A9	0	3	1	1	6	5	1	4	397	20	11	4	7	8	0	9	0	2	1
A10	0	1	0	1	2	1	0	2	14	416	27	0	2	7	1	2	0	2	2
A11	0	1	1	2	2	0	0	2	13	38	404	3	1	9	0	1	0	2	1
A12	1	0	2	2	0	0	0	0	0	3	2	456	2	2	2	3	0	1	4
A13	1	1	0	0	1	0	1	4	8	1	3	4	404	35	6	6	0	1	4
A14	0	1	1	1	1	0	0	8	5	3	6	5	20	411	5	4	0	3	6
A15	3	0	2	0	1	8	0	4	2	0	0	3	4	1	432	9	9	0	2
A16	2	3	1	1	9	2	1	3	3	2	3	2	9	8	7	420	2	0	2
A17	1	0	3	0	2	7	0	1	1	0	0	0	2	0	5	1	455	2	0
A18	0	1	1	1	2	7	1	9	8	1	2	4	0	1	1	2	5	430	4
A19	1	1	1	3	1	6	1	17	2	5	2	11	12	10	0	1	7	5	394

Performance of the k-NN method changes for different values of k. A value of $k=7$ gave the best results, therefore the confusion matrix of the k-NN algorithm is provided for $k=7$ in Table 5, and a successful differentiation rate of 98.7% is achieved.

We have implemented the DTW algorithm in two different ways: In the first (DTW_1), the average reference feature vector of each activity is used for distance comparison. The confusion matrix for DTW_1 is presented in Table 6, and a correct differentiation rate of 83.2% is achieved. As a second approach (DTW_2), DTW distances are calculated between the test vector and each of the 8208 ($=9120 - 912$) reference vectors from other classes. The class of the nearest reference vector is assigned as the class of the test vector. The success rate of DTW_2 is 98.5% and the corresponding confusion matrix is given in Table 7.

In SVM, following the one-versus-the-rest method, each type of activity is assumed as the first class and the remaining 18 activity types are grouped into the second class. With P-fold cross validation, 19 different SVM models are created for classifying the vectors in each partition, resulting in a total of 190 SVM models. The number of correctly and incorrectly classified feature vectors for each activity type is tabulated in Table 8(a). The overall correct classification rate of the SVM method is calculated as 98.8%.

For ANN, since the network classifies some samples as belonging to none of the classes and output neurons take continuous values between 0 and 1, it is not possible to form a confusion matrix. The number of correctly and incorrectly classified feature vectors with P-fold cross validation is given in Table 8(b). The overall correct classification rate of this method is 96.2%. On average, the network converges in about 400 epochs when P-fold cross validation is used.

To determine which activities can be distinguished easily, we employ the receiver operating characteristic (ROC) curves of some of the classifiers [56]. For a specific activity, we consider the instances belonging to that activity as positive instances, and all

Table 4
Confusion matrix for LSM (*P*-fold cross validation, 89.6%).

True	Classified																		
	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19
A1	415	0	60	0	0	0	3	2	0	0	0	0	0	0	0	0	0	0	0
A2	4	398	0	0	0	1	72	5	0	0	0	0	0	0	0	0	0	0	0
A3	3	0	471	0	0	0	1	0	0	0	0	0	0	0	0	0	5	0	0
A4	0	0	0	478	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0
A5	0	0	0	0	480	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A6	0	0	0	0	0	448	0	32	0	0	0	0	0	0	0	0	0	0	0
A7	16	55	0	1	6	1	350	51	0	0	0	0	0	0	0	0	0	0	0
A8	1	11	0	0	9	5	57	384	9	0	0	0	0	0	0	0	0	0	4
A9	0	0	0	0	19	7	0	0	361	52	35	0	6	0	0	0	0	0	0
A10	0	0	0	0	0	0	0	0	0	414	66	0	0	0	0	0	0	0	0
A11	0	0	0	0	1	0	0	0	0	78	401	0	0	0	0	0	0	0	0
A12	0	0	0	0	0	0	0	0	0	0	0	480	0	0	0	0	0	0	0
A13	0	0	0	0	0	0	0	4	0	0	0	0	466	9	0	0	0	1	0
A14	0	0	0	0	0	0	0	1	0	0	0	0	123	347	0	0	0	9	0
A15	0	0	0	0	0	0	0	0	0	0	0	0	1	0	476	1	2	0	0
A16	0	0	0	0	16	0	0	0	0	0	0	0	0	1	0	462	0	1	0
A17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	480	0	0
A18	0	0	0	0	1	65	0	0	15	0	0	0	0	0	0	0	0	399	0
A19	0	0	0	0	0	1	0	11	0	0	0	1	2	0	0	0	0	0	465

Table 5
Confusion matrix for the *k*-NN algorithm for *k*=7 (*P*-fold cross validation, 98.7%).

True	Classified																		
	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19
A1	480	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A2	0	479	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
A3	0	0	480	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A4	0	0	0	479	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
A5	0	0	0	0	480	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A6	0	0	0	0	0	480	0	0	0	0	0	0	0	0	0	0	0	0	0
A7	1	11	0	0	0	2	446	20	0	0	0	0	0	0	0	0	0	0	0
A8	0	5	0	0	4	10	38	422	1	0	0	0	0	0	0	0	0	0	0
A9	0	0	0	0	0	1	0	0	477	1	1	0	0	0	0	0	0	0	0
A10	0	0	0	0	0	0	0	0	0	474	6	0	0	0	0	0	0	0	0
A11	0	0	0	0	0	0	0	0	0	1	479	0	0	0	0	0	0	0	0
A12	0	0	0	0	0	0	0	0	0	0	0	480	0	0	0	0	0	0	0
A13	0	0	0	0	0	0	0	0	0	0	0	0	476	4	0	0	0	0	0
A14	0	0	0	0	0	0	0	0	0	0	0	0	1	479	0	0	0	0	0
A15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	479	1	0	0	0
A16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	479	0	0	0
A17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	480	0	0
A18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	480	0
A19	0	0	0	0	0	0	0	7	0	0	0	0	1	0	0	0	0	0	472

other instances as negative instances. Then, by setting a decision threshold or criterion for a classifier, the *true positive rate* (TPR) (the ratio of the true positives to the total positives) and the *false positive rate* (FPR) (the ratio of the false positives to the total negatives) can be calculated. Varying the decision threshold over an interval, a set of TPRs and the corresponding FPRs are obtained and plotted as a ROC curve.

Fig. 7 depicts the ROC curves for BDM, LSM, *k*-NN, and ANN classifiers as examples. In BDM and *k*-NN, the decision threshold is chosen as the posterior probability. For BDM, the posterior probability is calculated using the Bayes' rule. For *k*-NN, it is estimated by the ratio $(k_i+1)/(k+c)$, where $k=7$ for our case, $c=19$ is the total number of classes, and k_i is the number of training vectors that belong to class ω_i, out of the k nearest neighbors. This gives smoother estimates than using binary probabilities. In LSM, the decision threshold is chosen as the distance between

a test vector and the average reference vector of each class; and in ANN, the norm of the difference between the desired and actual outputs. Since there are 19 activities, the number of positive instances of each class is much less than the number of negative instances. Consequently, the FPRs are expected to be low and therefore, we plot the FPR in the logarithmic scale for better visualization. It can be observed in Fig. 7 that the sensitivity of BDM classifier is the highest. A test vector from classes A2, A7, or A8 is less likely to be correctly classified than a test vector belonging to one of the other classes. It is also confirmed by the confusion matrices that these are the most confused activities. For the LSM classifier, the same can be said for A13 and A14, as well as for A9, A10, and A11 where the FPRs for a given TPR are rather high. Despite this, for a tolerable FPR such as, say, 0.1, the TPR for LSM and ANN still remains above 0.75.

Table 6
Confusion matrix for DTW$_1$ (*P*-fold cross validation, 83.2%).

True	Classified																		
	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19
A1	413	2	44	15	0	0	5	0	0	0	0	0	1	0	0	0	0	0	0
A2	14	436	1	0	0	11	16	2	0	0	0	0	0	0	0	0	0	0	0
A3	38	2	430	2	0	0	4	0	0	0	0	0	0	0	1	0	3	0	0
A4	2	0	33	442	1	0	1	0	0	0	0	0	0	0	0	0	1	0	0
A5	0	0	0	0	476	1	0	2	0	1	0	0	0	0	0	0	0	0	0
A6	0	0	0	0	0	433	2	45	0	0	0	0	0	0	0	0	0	0	0
A7	24	85	2	0	10	3	305	51	0	0	0	0	0	0	0	0	0	0	0
A8	3	13	0	0	19	4	68	355	8	3	0	0	2	1	1	0	0	1	2
A9	0	0	0	0	36	11	0	0	338	42	41	0	6	4	0	2	0	0	0
A10	0	0	0	0	4	0	0	0	2	312	136	0	5	3	0	18	0	0	0
A11	0	0	0	0	5	0	0	1	1	134	307	0	4	7	1	20	0	0	0
A12	0	0	0	0	0	0	0	0	0	0	0	480	0	0	0	0	0	0	0
A13	0	0	0	0	1	0	0	3	1	1	0	0	433	36	0	5	0	0	0
A14	0	0	0	0	0	2	0	0	1	0	3	0	167	296	0	8	0	3	0
A15	1	0	0	0	0	0	0	7	1	0	0	0	9	0	448	14	0	0	0
A16	0	1	0	0	38	0	0	1	3	0	0	0	8	2	6	421	0	0	0
A17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	480	0	0
A18	0	3	0	0	4	68	12	3	18	0	0	0	5	13	0	0	7	346	1
A19	0	0	0	0	4	2	0	12	0	2	3	3	7	5	0	0	0	3	439

Table 7
Confusion matrix for DTW$_2$ (*P*-fold cross validation, 98.5%).

True	Classified																		
	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19
A1	480	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A2	0	480	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A3	0	0	480	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A4	0	0	0	480	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A5	0	0	0	0	480	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A6	0	0	0	0	0	480	0	0	0	0	0	0	0	0	0	0	0	0	0
A7	1	7	0	0	0	1	430	41	0	0	0	0	0	0	0	0	0	0	0
A8	0	1	0	0	2	9	47	418	0	0	0	0	0	0	0	0	0	0	3
A9	0	0	0	0	0	1	0	0	478	0	1	0	0	0	0	0	0	0	0
A10	0	0	0	0	0	0	0	0	0	469	11	0	0	0	0	0	0	0	0
A11	0	0	0	0	0	0	0	0	0	5	475	0	0	0	0	0	0	0	0
A12	0	0	0	0	0	0	0	0	0	0	0	480	0	0	0	0	0	0	0
A13	0	0	0	0	0	0	0	0	0	0	0	0	479	1	0	0	0	0	0
A14	0	0	0	0	0	0	0	0	0	0	0	0	0	480	0	0	0	0	0
A15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	480	0	0	0	0
A16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	480	0	0	0
A17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	480	0	0
A18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	480	0
A19	0	0	0	0	0	0	0	5	0	0	0	0	1	0	0	0	0	0	474

5.1. Computational cost of the classification techniques

We also compared the classification techniques given above based on their computational costs. Pre-processing and classification times are calculated with MATLAB version 7.0.4, on a desktop computer with AMD Athlon 64 X2 dual core processor at 2.2 GHz and 2.00 GB of RAM, running Microsoft Windows XP Professional operating system. Pre-processing/training, storage requirements, and processing times of the different techniques are tabulated in Table 9. The pre-processing time of BDM is used for estimating the mean vector, covariance matrix and the CCPDFs that need to be stored for the test stage. In RBA, the pre-processing phase involves extracting the rules based on the training data. Once the rules are available, the vectors need not be stored and any test vector can be classified using the RBA. In LSM and DTW$_1$, the averages of the training vectors for each class need to be stored for the test phase.

Note that the pre-processing times of these two methods are exactly equal. For *k*-NN and DTW$_2$, all training vectors need to be stored. For the SVM, the SVM models constructed in the training phase need to be stored for the test phase. For ANN, the structure of the trained network and the connection weights need to be saved for testing. ANN and SVM require the longest training time and SVM also has considerable storage requirements. These are followed by RBA, BDM and LSM (same as DTW$_1$). The *k*-NN and DTW$_2$ methods do not require any pre-processing.

The processing times for classifying a single feature vector are given in the same table. The classification time for ANN is the smallest, followed by LSM, RBA, BDM, SVM, DTW$_1$ and DTW$_2$ or *k*-NN methods. The latter two take the longest amount of classification time because of the nature of the classifiers and also because a comparison should be made with every training vector.

Table 8
(a) Number of correctly and incorrectly classified motions out of 480 for SVMs (*P*-fold cross validation, 98.8%); (b) same for ANN (*P*-fold cross validation, 96.2%).

True	Classified	
	Correct	Incorrect
(a)		
A1	480	0
A2	479	1
A3	478	2
A4	477	3
A5	480	0
A6	478	2
A7	445	35
A8	430	50
A9	476	4
A10	479	1
A11	479	1
A12	480	0
A13	477	3
A14	480	0
A15	480	0
A16	479	1
A17	480	0
A18	480	0
A19	473	7
(b)		
A1	471	9
A2	454	26
A3	475	5
A4	478	2
A5	473	7
A6	463	17
A7	421	59
A8	388	92
A9	457	23
A10	471	9
A11	464	16
A12	479	1
A13	467	13
A14	470	10
A15	475	5
A16	472	8
A17	479	1
A18	478	2
A19	461	19

5.2. Discussion

Given its very high correct classification rate and relatively small pre-processing and classification times and storage requirements, it can be concluded that BDM is superior to the other classification techniques we considered for the given classification problem. This result supports the idea that the distribution of the activities in the feature space can be well approximated by multi-variate Gaussian distributions. The low processing and storage requirements of the BDM method make it a strong candidate for similar classification problems.

SVM, although very accurate, requires a considerable amount of training time to construct the SVM models. Its storage requirements and processing time fall in the middle. The *k*-NN method is also very accurate, with zero pre-processing time but its processing time is one of the two largest. For real-time applications, LSM could also be a suitable choice because it is faster than BDM at the expense of a 10% lower correct classification rate. The ANN requires considerable training time but once it is trained and the connection weights are stored, classification is done very rapidly.

The performances of the classifiers do not change considerably if one or more of the sensors fail because of power cut or any other malfunction. In Table 10, we present classification results with reduced number of sensors. For example, using only the sensor on the torso, correct classification rate of 96.6% can be achieved with BDM. It can also be observed that the sensors on the legs are more discriminative than the sensors on the arms, with the left leg being more discriminative for most of the classification methods. Most of the activities performed in this study involve quasi-symmetric movement of the body with respect to the sagittal plane. That is, left and right sides of the body follow basically the same movement patterns that are either stationary (sitting, standing), in phase (jumping, rowing), or out of phase (walking, running, cycling). Exceptions are basketball and lying on the right side activities. The cycling activities involve symmetric out-of-phase movement of the legs, but not the arms. The sensor locations are symmetric as well, thus one can expect redundancy in the information acquired by the sensors. However, this redundancy can be exploited in case of the failure of one or more sensors. This can also be observed in Table 10. The correct classification rates using the left side and right side sensors are close to each other, which means that if a sensor on either side fails, its symmetric counterpart on the other side will compensate for that sensor. The torso sensor does not have a symmetric counterpart; however, its failure would result in only a slight decrease in the correct classification rate as can be seen in the table.

6. Potential application areas

Human activity monitoring and classification have applications in diverse areas. A significant application area is the remote monitoring of elderly people who live alone and are in need of additional support; emergency situations arising from falls and changes in vital signs could be detected within a short time. Similarly, remote monitoring of people with physical or mental disabilities, and children at home, school, or in the neighborhood may be of interest. Home-based rehabilitation of the elderly is another potential area of application. For example, it would be possible to check whether the patient is able to perform his/her physical therapy exercises in the correct and most efficient manner and provide feedback to enable proper performance of the exercises. Furthermore, joint processing and evaluation of sensory information from heart rate, blood pressure, and temperature monitors together with motion and position information can allow a detailed judgment of the situation and help determine whether attention is required.

Another potential area of high impact is in ergonomics, regarding the proper use of tools, devices, and instruments, important both for efficiency and for human health. Worker productivity could be improved by monitoring whether tasks are performed in the most efficient, optimal, safe, and nonexhausting manner. This would also help in the prevention of repetitive motion injury (e.g. carpal tunnel syndrome) by providing warning signals against improper motions.

Likewise, in the area of physical education, training and sports, and dance, such monitoring can be used to help trainers and individuals obtain feedback regarding their motions in terms of effectiveness and safety, as well as increasing the benefits of physical exercise, improving athletic performance, and most importantly, promoting health and preventing injuries. This application could also be useful in detecting sports-rule violations.

Recording sports performances and traditional and modern dance is an application that would be significant from a cultural heritage viewpoint, complementary to ordinary video recording. Whereas ordinary video recording provides a projection of the
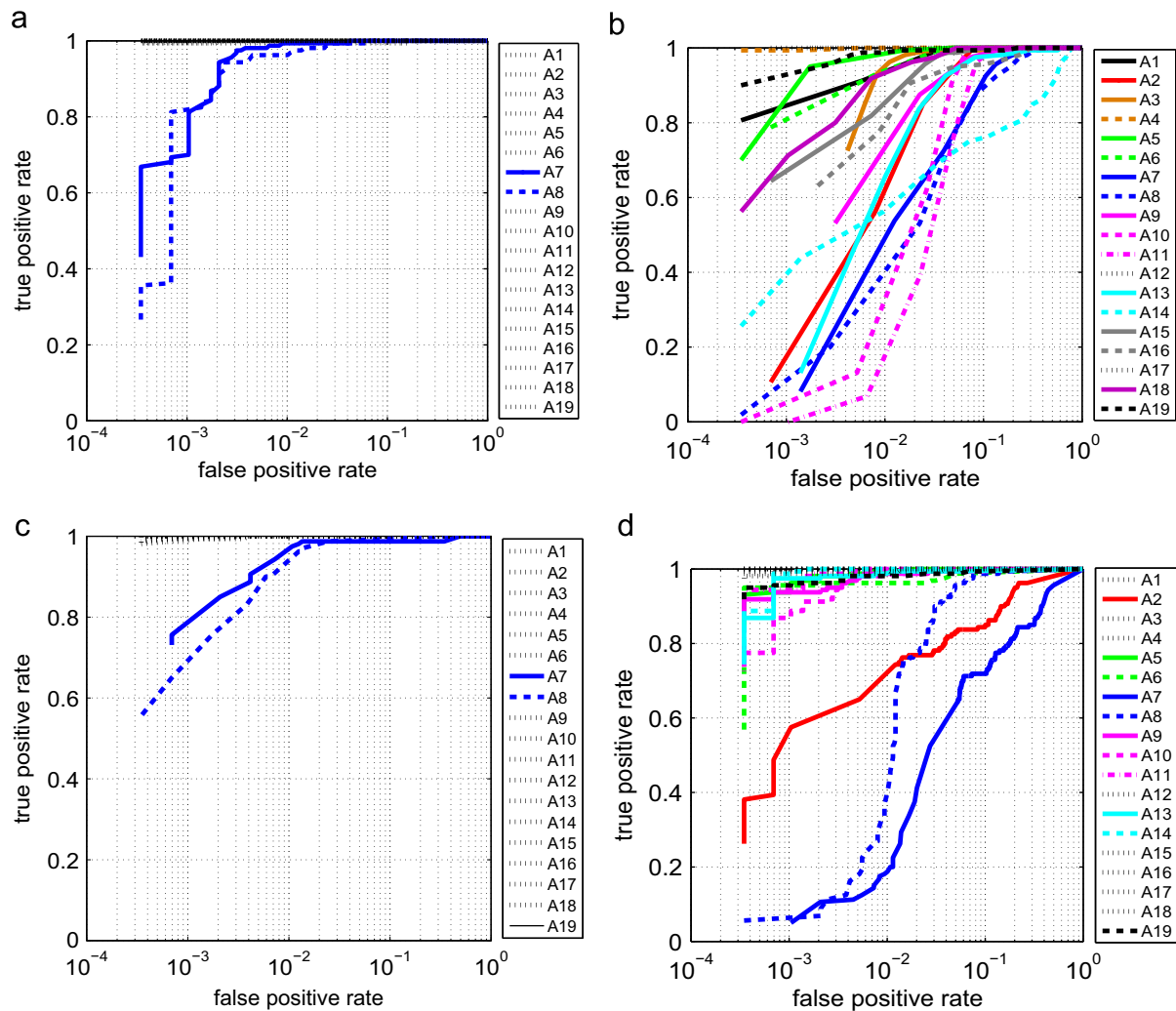
Fig. 7. (Color online) ROC curves for (a) BDM, (b) LSM, (c) *k*-NN, and (c) ANN using RRSS cross validation. In parts (a) and (c), activities other than A7 and A8 are all represented by dotted horizontal lines at the top where the true positive rate equals one.

Table 9
Pre-processing and training times, storage requirements, and processing times of the classification methods.

Method	Pre-processing/training time (ms)			Storage requirements	Processing time (ms)		
	RRSS	*P*-fold	L1O		RRSS	*P*-fold	L1O
BDM	28.98	28.62	24.70	Mean, covariance, CCPDF	4.56	5.70	5.33
RBA	2514.21	3874.78	3400.14	Rules	0.64	0.95	0.84
LSM	6.77	9.92	5.42	Average of training vectors for each class	0.25	0.24	0.21
k-NN	–	–	–	All training vectors	101.32	351.22	187.32
DTW$_1$	6.77	9.92	5.42	Average of training vectors for each class	86.26	86.22	85.57
DTW$_2$	–	–	–	All training vectors	116.57	155.81	153.25
SVM	7368.17	13,287.85	10,098.61	SVM models	19.49	7.24	8.02
ANN	290,815	228,278	214,267	Network structure and connection weights	0.06	0.06	0.06

The processing times are given for classifying a single feature vector.

motion from the perspective of the camera, recording key body motion parameters provides a structural-functional description of the motions in terms of the degrees of freedom of the subjects and their body parts, which could be considered more intrinsic than the camera image.

It is not hard to imagine applications for learning to play a musical instrument or even conducting an orchestra. Students and professionals alike can benefit from self-monitoring and

could use these techniques as an aid to overcome bad habits and improve and perfect their techniques. Motion injuries are also encountered in musicians so this application may also assist in their prevention.

Generalizing from these example applications, these approaches can be used in any area where a characteristic human motion is involved and the individual subject may exhibit a distinct signature. Handwriting patterns, walking patterns, and

Table 10
All possible sensor combinations and the corresponding correct classification rates for some of the methods using *P*-fold cross-validation.

Sensors used	BDM	RBA	LSM	*k*-NN	DTW$_1$	DTW$_2$	SVM		BDM	RBA	LSM	*k*-NN	DTW$_1$	DTW$_2$	SVM
–	–	–	–	–	–	–	–	+T	96.6	67.5	79.0	92.8	62.5	92.9	93.4
RA	94.5	56.8	72.5	88.4	57.1	87.5	90.6	+T	98.1	74.9	84.0	95.8	65.0	95.7	97.7
LA	93.7	59.6	75.3	87.8	47.8	84.4	91.0	+T	98.1	69.5	87.2	95.6	67.2	94.6	97.4
RL	97.3	66.5	82.3	91.0	70.2	87.3	93.8	+T	98.4	80.8	85.4	97.2	76.3	97.6	98.0
LL	94.8	79.8	79.1	96.7	74.6	97.5	96.0	+T	98.4	80.7	85.4	97.3	75.9	97.5	97.9
RA+LA	97.6	68.8	83.4	95.5	61.6	94.8	97.0	+T	98.5	76.8	86.8	97.5	74.3	97.4	98.3
RL+LL	98.8	78.2	84.0	96.6	75.8	95.6	97.8	+T	99.0	83.5	86.3	97.7	79.5	97.7	98.4
RA+RL	98.0	75.6	84.3	96.9	73.2	95.9	97.5	+T	98.8	79.4	87.6	98.0	77.2	97.7	98.5
LA+LL	98.4	76.1	85.6	95.9	72.4	94.9	97.9	+T	98.8	80.0	88.1	97.2	76.4	97.0	98.2
RA+LL	98.5	77.0	83.6	96.3	73.9	96.4	98.0	+T	98.9	80.6	87.2	97.6	80.2	97.8	98.5
LA+RL	97.8	72.5	86.1	95.9	72.7	94.5	97.1	+T	98.7	77.4	88.9	97.5	76.3	97.2	98.4
RA+LA+RL	98.7	77.0	87.1	97.7	76.4	97.3	98.4	+T	98.9	79.2	89.0	98.3	79.4	98.3	98.7
RA+LA+LL	98.7	79.0	86.7	97.8	77.3	97.3	98.5	+T	99.0	81.6	88.9	98.4	80.7	98.2	98.6
RA+RL+LL	99.0	81.5	86.1	98.1	78.7	97.7	98.7	+T	99.1	82.4	88.3	98.4	82.3	98.6	98.8
LA+RL+LL	98.9	80.7	87.2	97.6	75.9	97.4	98.4	+T	99.0	83.7	89.3	98.2	78.5	98.2	98.5
RA+LA+RL+LL	99.0	82.5	88.0	98.3	79.0	98.4	98.8	+T	99.2	84.5	89.6	98.7	83.2	98.5	98.8

T: torso, RA: right arm, LA: left arm, RL: right leg, LL: left leg.

other such regular characteristic behavior exhibit different patterns from person to person and may be used as a generalized signature of that person for both recognition and validation purposes.

If we extend the application areas beyond recognizing and classifying human motion, there are plenty of applications in monitoring and classifying animal motion [16]. For example, changes in the behavior of groups of animals due to diseases such as avian flu or bovine encephalitis could be detected.

Other relevant areas of application are motion artifact compensation in medical imaging, stabilizing cameras and video recorders. This would require the data from motion sensors to be combined with conventionally acquired information and necessitate developing appropriate algorithms, things beyond the present state of the art in this area. For instance, a motion compensation system that relies solely on acquired images must use indirectly deduced motion parameters and sophisticated and potentially time-consuming processing. On the other hand, direct information obtained from motion sensors would potentially enable unnecessary motion artifacts to be eliminated more precisely with less computational load. Motion sensors for this purpose can be attached to the subject, the camera or the acquisition/recording device, or both. While there could be applications in which attaching a sensor to the subject is not practical, attaching motion sensors to a patient undergoing diagnostic imaging would likely not be objectionable. In cases where it is not acceptable to place the motion sensors on the subject, they could be placed on the camera or the video recorder.

In the area of animation and film making, including emerging 3-D television technology, motion sensors might not only contribute to the development of realistic animated models but also provide useful auxiliary information to the acquisition process.

Motion sensors attached to human subjects may also find use in computer games, virtual reality, and professional simulators, enabling better coupling between the displayed virtual environment and the subject's actions.

As sensors continue to get smaller and cheaper, it will become more and more convenient to integrate them in commonly used accessories such as watches, glasses, headbands, belts, hats, hearing aids, etc. We also expect the development of extremely small and thin, lightweight sensor patches that may be worn on the skin like a bandage. This will greatly expand sensor's applications; as the discomfort or burden of wearing them becomes negligible, it will be possible to consider applications in many other areas of daily life that are currently out of question because the present sensors are not light enough.

7. Conclusions and future work

We have presented the results of a comparative study where features extracted from miniature inertial and magnetometer signals are used for classifying human activities. We compared a number of classification techniques based on the same data set in terms of their correct differentiation rates, confusion matrices, computational costs, and training and storage requirements. BDM achieves higher correct classification rates in general compared to the other classification techniques, and has relatively small computational time and storage requirements. This parametric method can be employed in similar classification problems, where it is appropriate to model the feature space with multi-variate Gaussian distributions. The SVM and *k*-NN methods are the second-best choices in terms of classification accuracy but SVM requires a considerable amount of training time. For real-time applications, LSM could also be considered a suitable choice because it is faster than BDM at the expense of a lower correct classification rate.

We implemented and compared a number of different cross-validation techniques in this study. The correct classification rates obtained by subject-based L1O cross validation are usually lower. RRSS uses the shortest amount of processing time, whereas *P*-fold requires the longest. However, the main disadvantage of RRSS is that some feature vectors may never be used for testing, whereas others may be used more than once. In *P*-fold and L1O cross validation, all feature vectors are used for both training and testing.

There are several possible future research directions that can be explored:

An aspect of activity recognition and classification that has not been much investigated is the normalization between the way different individuals perform the same activities. Each person does a particular activity differently due to differences in body size, style, and timing. Although some approaches may be more prone to highlighting personal differences, new techniques need to be developed that involve time-warping and projections of signals and comparing their differentials.

To the best of our knowledge, optimizing the positioning, number, and type of sensors has not been much studied.

Typically, some configuration, number, and modality of sensors is chosen and used without strong justification.

Detecting and classifying falls using inertial sensors is another important problem that has not been sufficiently well investigated [21], due to the difficulty of designing and performing fair and realistic experiments in this area [12]. Therefore, standard definitions of falls and systematic techniques for detecting and classifying falls still do not exist. In our ever-aging population, it seems imperative to develop such definitions and techniques as soon as possible [19,20].

Fusing information from inertial sensors and cameras can be further explored to provide robust solutions in human activity monitoring, recognition, and classification. Joint use of these two sensing modalities increases the capabilities of intelligent systems and enlarges the application potential of inertial and vision systems.

Acknowledgments

References

[1] I.J. Cox, G.T. Wilfong, Section on inertial navigation, in: M.M. Kuritsky, M.S. Goldstein (Eds.), Autonomous Robot Vehicles, Springer-Verlag, New York, USA, 1990.

[2] D.A. Mackenzie, Inventing Accuracy: A Historical Sociology of Nuclear Missile Guidance, MIT Press, Cambridge, MA, USA, 1990.

[3] B. Barshan, H.F. Durrant-Whyte, Inertial navigation systems for mobile robots, IEEE Trans. Robotics Autom. 11 (3) (1995) 328–342.

[4] C.-W. Tan, S. Park, Design of accelerometer-based inertial navigation systems, IEEE Trans. Instrum. Meas. 54 (6) (2005) 2520–2530.

[5] B. Barshan, H.F. Durrant-Whyte, Evaluation of a solid-state gyroscope for robotics applications, IEEE Trans. Instrum. Meas. 44 (1) (1995) 61–67.

[6] J.G. Nichol, S.P.N. Singh, K.J. Waldron, L.R. Palmer III, D.E. Orin, System design of a quadrupedal galloping machine, Int. J. Robotics Res. 23 (10–11) (2004) 1013–1027.

[7] P.-C. Lin, H. Komsuoglu, D.E. Koditschek, Sensor data fusion for body state estimation in a hexapod robot with dynamical gaits, IEEE Trans. Robotics 22 (5) (2006) 932–943.

[8] W.T. Ang, P.K. Khosla, C.N. Riviere, Design of all-accelerometer inertial measurement unit for tremor sensing in hand-held microsurgical instrument, in: Proceedings of the IEEE International Conference on Robotics and Automation, vol. 2, September 2003, pp. 1781–1786.

[9] W.T. Ang, P.K. Pradeep, C.N. Riviere, Active tremor compensation in microsurgery, in: Proceedings of the 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, vol. 1, September 2004, pp. 2738–2741.

[10] D.H. Titterton, J.L. Weston, Strapdown Inertial Navigation Technology, second ed., IEE, UK, 2004.

[11] W. Zijlstra, K. Aminian, Mobility assessment in older people: new possibilities and challenges, Eur. J. Ageing 4 (1) (2007) 3–12.

[12] M.J. Mathie, A.C.F. Coster, N.H. Lovell, B.G. Celler, Accelerometry: providing an integrated, practical method for long-term, ambulatory monitoring of human movement, Physiol. Meas. 25 (2) (2004) R1–R20.

[13] W.Y. Wong, M.S. Wong, K.H. Lo, Clinical applications of sensors for human posture and movement analysis: a review, Prosthet. Orthot. Int. 31 (1) (2007) 62–75.

[14] O. Tunçel, K. Altun, B. Barshan, Classifying human leg motions with uniaxial piezoelectric gyroscopes, Sensors 9 (11) (2009) 8508–8546.

[15] A.M. Sabatini, Inertial sensing in biomechanics: a survey of computational techniques bridging motion analysis and personal navigation, Computational Intelligence for Movement Sciences: Neural Networks and Other Emerging Techniques, Idea Group Publishing, Hershey, PA, USA, 2006, pp. 70–100. ⟨http://www.idea-group.com⟩.

[16] F. Audigié, P. Pourcelot, C. Degueurce, D. Geiger, J.M. Denoix, Fourier analysis of trunk displacements: a method to identify the lame limb in trotting horses, J. Biomech. 35 (9) (2002) 1173–1182.

[17] J. Pärkkä, M. Ermes, P. Korpipää, J. Mäntyjärvi, J. Peltola, I. Korhonen, Activity classification using realistic data from wearable sensors, IEEE Trans. Inf. Technol. B 10 (1) (2006) 119–128.

[18] M.J. Mathie, B.G. Celler, N.H. Lovell, A.C.F. Coster, Classification of basic daily movements using a triaxial accelerometer, Med. Biol. Eng. Comput. 42 (5) (2004) 679–687.

[19] K. Hauer, S.E. Lamb, E.C. Jorstad, C. Todd, C. Becker, Systematic review of definitions and methods of measuring falls in randomised controlled fall prevention trials, Age Ageing 35 (1) (2006) 5–10.

[20] N. Noury, A. Fleury, P. Rumeau, A.K. Bourke, G.O. Laighin, V. Rialle, J.E. Lundy, Fall detection—principles and methods, in: Proceedings of the 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, August 2007, pp. 1663–1666.

[21] M. Kangas, A. Konttila, P. Lindgren, I. Winblad, T. Jämsä, Comparison of low-complexity fall detection algorithms for body attached accelerometers, Gait Posture 28 (2) (2008) 285–291.

[22] W.H. Wu, A.A.T. Bui, M.A. Batalin, D. Liu, W.J. Kaiser, Incremental diagnosis method for intelligent wearable sensor system, IEEE Trans. Inf. Technol. B 11 (5) (2007) 553–562.

[23] E. Jovanov, A. Milenkovic, C. Otto, P.C. de Groen, A wireless body area network of intelligent motion sensors for computer assisted physical rehabilitation, J. NeuroEng. Rehab. 2 (6) (2005).

[24] M. Ermes, J. Pärkkä, J. Mäntyjärvi, I. Korhonen, Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions, IEEE Trans. Inf. Technol. B 12 (1) (2008) 20–26.

[25] R. Aylward, J.A. Paradiso, Sensemble: a wireless, compact, multi-user sensor system for interactive dance, in: Proceedings of the Conference on New Interfaces for Musical Expression, Paris, France, 4–8 June 2006, pp. 134–139.

[26] J. Lee, I. Ha, Real-time motion capture for a human body using accelerometers, Robotica 19 (6) (2001) 601–610.

[27] T. Shiratori, J.K. Hodgins, Accelerometer-based user interfaces for the control of a physically simulated character, ACM Trans. Graphics (SIGGRAPH Asia 2008) 27 (5) (2008).

[28] T.B. Moeslund, E. Granum, A survey of computer vision-based human motion capture, Comput. Vision Image Understanding 81 (3) (2001) 231–268.

[29] T.B. Moeslund, A. Hilton, V. Krüger, A survey of advances in vision-based human motion capture and analysis, Comput. Vision Image Understanding 104 (2–3) (2006) 90–126.

[30] L. Wang, W. Hu, T. Tan, Recent developments in human motion analysis, Pattern Recognition 36 (3) (2003) 585–601.

[31] J.K. Aggarwal, Q. Cai, Human motion analysis: a review, Comput. Vision Image Understanding 73 (3) (1999) 428–440.

[32] L. Hyeon-Kyu, J.H. Kim, An HMM-based threshold model approach for gesture recognition, IEEE Trans. Pattern Anal. 21 (10) (1999) 961–973.

[33] H. Junker, O. Amft, P. Lukowicz, G. Troester, Gesture spotting with body-worn inertial sensors to detect user activities, Pattern Recognition 41 (6) (2008) 2010–2024.

[34] J.C. Lementec, P. Bajcsy, Recognition of arm gestures using multiple orientation sensors: gesture classification, in: Proceedings of the 7th International Conference on Intelligent Transportation Systems, Washington, DC, USA, 3–6 October 2004, pp. 965–970.

[35] M. Uiterwaal, E.B.C. Glerum, H.J. Busser, R.C. van Lummel, Ambulatory monitoring of physical activity in working situations, a validation study, J. Med. Eng. Technol. 22 (4) (1998) 168–172.

[36] J.B. Bussmann, P.J. Reuvekamp, P.H. Veltink, W.L. Martens, H.J. Stam, Validity and reliability of measurements obtained with an "activity monitor" in people with and without transtibial amputation, Phys. Ther. 78 (9) (1998) 989–998.

[37] K. Aminian, P. Robert, E.E. Buchser, B. Rutschmann, D. Hayoz, M. Depairon, Physical activity monitoring based on accelerometry: validation and comparison with video observation, Med. Biol. Eng. Comput. 37 (1) (1999) 304–308.

[38] D. Roetenberg, P.J. Slycke, P.H. Veltink, Ambulatory position and orientation tracking fusing magnetic and inertial sensing, IEEE Trans. Bio-med. Eng. 54 (5) (2007) 883–890.

[39] B. Najafi, K. Aminian, F. Loew, Y. Blanc, P. Robert, Measurement of stand-sit and sit-stand transitions using a miniature gyroscope and its application in fall risk evaluation in the elderly, IEEE Trans. Bio-med. Eng. 49 (8) (2002) 843–851.

[40] B. Najafi, K. Aminian, A. Paraschiv-Ionescu, F. Loew, C.J. Büla, P. Robert, Ambulatory system for human motion analysis using a kinematic sensor: monitoring of daily physical activity in the elderly, IEEE Trans. Bio-med. Eng. 50 (6) (2003) 711–723.

[41] Y. Tao, H. Hu, H. Zhou, Integration of vision and inertial sensors for 3D arm motion tracking in home-based rehabilitation, Int. J. Robotics Res. 26 (6) (2007) 607–624.

[42] T. Viéville, O.D. Faugeras, Cooperation of the inertial and visual systems, in: Traditional and Non-Traditional Robotic Sensors, 59th ed., NATO ASI Series, vol. F63, Springer-Verlag, Berlin, Germany, 1990, pp. 339–350.

[43] Proceedings of the Workshop on Integration of Vision and Inertial Sensors (InerVis), Coimbra, Portugal, June 2003; Barcelona, Spain, April 2005.

[44] Special Issue on the 2nd Workshop on Integration of Vision and Inertial Sensors (InerVis05), Int. J. Robotics Res., vol. 26, June 2007.

[45] L. Bao, S.S. Intille, Activity recognition from user-annotated acceleration data, in: Pervasive Computing, Lecture Notes in Computer Science, vol. 3001, 2004, pp. 1–17.

[46] R. Zhu, Z. Zhou, A real-time articulated human motion tracking using tri-axis inertial/magnetic sensors package, IEEE Trans. Neural Syst. Rehab. Eng. 12 (2) (2004) 295–302.

[47] X. Yun, E.R. Bachmann, H. Moore, J. Calusdian, Self-contained position tracking of human movement using small inertial/magnetic sensor modules, in: Proceedings of the IEEE International Conference on Robotics and Automation, Rome, Italy, 10–14 April 2007, pp. 2526–2533.

[48] P.H. Veltink, H.B.J. Bussmann, W. de Vries, W.L.J. Martens, R.C. Van Lummel, Detection of static and dynamic activities using uniaxial accelerometers, IEEE Trans. Rehabil. Eng. 4 (4) (1996) 375–385.

[49] K. Kiani, C.J. Snijders, E.S. Gelsema, Computerized analysis of daily life motor activity for ambulatory monitoring, Technol. Health Care 5 (4) (1997) 307–318.

[50] F. Foerster, M. Smeja, J. Fahrenberg, Detection of posture and motion by accelerometry: a validation study in ambulatory monitoring, Comput. Hum. Behav. 15 (5) (1999) 571–583.

[51] D.M. Karantonis, M.R. Narayanan, M. Mathie, N.H. Lovell, B.G. Celler, Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring, IEEE Trans. Inf. Technol. B 10 (1) (2006) 156–167.

[52] F.R. Allen, E. Ambikairajah, N.H. Lovell, B.G. Celler, Classification of a known sequence of motions and postures from accelerometry data using adapted Gaussian mixture models, Physiol. Meas. 27 (10) (2006) 935–951.

[53] N. Kern, B. Schiele, A. Schmidt, Multi-sensor activity context detection for wearable computing, in: Ambient Intelligence, Lecture Notes in Computer Science, vol. 2875, Eindhoven, The Netherlands, November 2003, pp. 220–232.

[54] Xsens Technologies B.V., Enschede, Holland, MTi and MTx User Manual and Technical Documentation, 2009 〈http://www.xsens.com〉.

[55] S.J. Preece, J.Y. Goulermas, L.P.J. Kenney, D. Howard, A comparison of feature extraction methods for the classification of dynamic activities from accelerometer data, IEEE Trans. Bio-med. Eng. 56 (3) (2009) 871–879.

[56] A. Webb, Statistical Pattern Recognition, John Wiley & Sons, New York, USA, 2002.

[57] S.J. Preece, J.Y. Goulermas, L.P.J. Kenney, D. Howard, K. Meijer, R. Crompton, Activity identification using body-mounted sensors—a review of classification techniques, Physiol. Meas. 30 (4) (2009) R1–R33.

[58] S. Theodoridis, K. Koutroumbas, Pattern Recognition, Academic Press, Orlando, FL, USA, 2006.

[59] R.A.O.L. Breiman, J.H. Friedman, C.J. Stone, Classification and Regression Trees, Chapman & Hall, CRC, Boca Raton, FL, USA, 1998.

[60] J.R. Deller, J.H.L. Hansen, J.G. Proakis, Discrete-Time Processing of Speech Signals, IEEE Press, New York, USA, 2000.

[61] T. Parsons, Voice and Speech Processing, McGraw-Hill, New York, USA, 1986.

[62] Z.M. Kovács-Vajna, A fingerprint verification system based on triangular matching and dynamic time warping, IEEE Trans. Pattern Anal. 22 (11) (2000) 1266–1276.

[63] M.F. Zanuy, On-line signature recognition based on VQ-DTW, Pattern Recognition 40 (3) (2007) 981–992.

[64] V.N. Vapnik, Estimation of Dependences based on Empirical Data, Nauka, Moscow, 1979 (in Russian, English translation: Springer-Verlag, New York, USA, 1982).

[65] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, second ed., John Wiley & Sons Inc., New York, USA, 2001.

[66] C.W. Hsu, C.C. Chang, C.J. Lin, A Practical Guide to Support Vector Classification, 2008. 〈http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide. pdf〉.

[67] C.C. Chang, C.J. Lin, LIBSVM: a library for support vector machines, 2001, Software available at 〈http://www.csie.ntu.edu.tw/~cjlin/libsvm〉.

[68] S. Haykin, Neural Networks: A Comprehensive Foundation, Macmillan Publishing, New York, USA, 1994.

[69] M.T. Hagan, H.B. Demuth, M.H. Beale, Neural Network Design, PWS, Boston, MA, USA, 1996.

About the Author—KEREM ALTUN received his B.S. and M.S. degrees in mechanical engineering from Middle East Technical University (METU) in Ankara, Turkey, in 1999 and 2002, respectively. He is currently working towards his Ph.D. degree in the Department of Electrical and Electronics Engineering at Bilkent University, Ankara. He received a Ph.D. scholarship from The Scientific and Technological Research Council of Turkey in 2006, and the Professor Güven Aksu award from METU's Mechanical Engineering Department in 1998 for outstanding performance in an undergraduate course. His research interests include intelligent sensing, sensor signal processing, ultrasonic and inertial sensing, pattern recognition, system dynamics and control, and sensor data fusion.

About the Author—BILLUR BARSHAN received B.S. degrees in both electrical engineering and physics from Boğaziçi University in Istanbul, Turkey, in 1986 and her M.S. and Ph.D. degrees in electrical engineering from Yale University in the USA in 1988 and 1991, respectively. Dr. Barshan was a research assistant at Yale from 1987 to 1991 and a postdoctoral researcher at the Robotics Research Group at the University of Oxford in the UK from 1991 to 1993. In 1993, she joined the faculty of Bilkent University in Ankara, Turkey, where she is a professor in the Department of Electrical and Electronics Engineering. Dr. Barshan is also the founder of the Robotics and Sensing Laboratory in the same department. She is the recipient of the 1994 Nakamura Prize for the most outstanding paper at the 1993 IEEE/RSJ Intelligent Robots and Systems International Conference, the 1998 TÜBİTAK Young Investigator Award, and the 1999 METU Mustafa N. Parlar Foundation Research Award. Dr. Barshan's current research interests include sensor-based robotics, intelligent sensing, ultrasonic, optical, and inertial sensing, sensor signal processing, multi-sensor data fusion, and human activity recognition and classification.

About the Author—ORKUN TUNÇEL received his B.S. and M.S. degrees in electrical and electronics engineering from Bilkent University, Ankara, Turkey, in 2006 and 2009, respectively. Among his research interests are inertial sensor signal processing, human activity classification, pattern recognition, and mobile robotics.