Human Activity Classification with Miniature Inertial Sensors

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND

ELECTRONICS ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCES

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

Orkun Tunçel

July 2009

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Billur Barshan(Supervisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. A. Enis Çetin

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Selim Aksoy

Approved for the Institute of Engineering and Sciences:

Prof. Dr. Mehmet Baray
Director of Institute of Engineering and Sciences

ABSTRACT

Human Activity Classification with Miniature Inertial Sensors

Orkun Tunçel
M.S. in Electrical and Electronics Engineering
Supervisor: Prof. Dr. Billur Barshan
July 2009

This thesis provides a comparative study on activity recognition using miniature inertial sensors (gyroscopes and accelerometers) and magnetometers worn on the human body. The classification methods used and compared in this study are: a rule-based algorithm (RBA) or decision tree, least-squares method (LSM), k-nearest neighbor algorithm (k-NN), dynamic time warping (DTW-1 and DTW-2), and support vector machines (SVM). In the first part of this study, eight different leg motions are classified using only two single-axis gyroscopes. In the second part, human activities are classified using five sensor units worn on different parts of the body. Each sensor unit comprises a tri-axial gyroscope, a tri-axial accelerometer and a tri-axial magnetometer. Different feature sets extracted from the raw sensor data and these are used in the classification process. A number of feature extraction and reduction techniques (principal component analysis) as well as different cross-validation techniques have been implemented and compared. A performance comparison of these classification methods is provided in terms of their correct differentiation rates, confusion matrices, pre-processing and training times and classification times. Among the classification techniques we have considered and implemented, SVM, in general,

gives the highest correct differentiation rate, followed by k-NN. The classification time for RBA is the shortest, followed by SVM or LSM, k-NN or DTW-1, and DTW-2 methods. SVM requires the longest training time, whereas DTW-2 takes the longest amount of classification time. Although there is not a significant difference between the correct differentiation rates obtained by different cross-validation techniques, repeated random sub-sampling uses the shortest amount of classification time, whereas leave-one-out requires the longest.

Keywords: inertial sensors, gyroscope, accelerometer, magnetometer, human activity recognition, motion classification, pattern recognition, feature, principal component analysis, cross-validation, rule-based algorithm, decision tree, least-squares method, k-nearest neighbor, dynamic time warping, support vector machines.

ÖZET

MİNYATÜR EYLEMSİZLİK DUYUCULARI KULLANILARAK İNSAN HAREKETLERİNİN SINIFLANDIRILMASI

Orkun Tunçel
Elektrik ve Elektronik Mühendisliği Bölümü Yüksek Lisans
Tez Yöneticisi: Prof. Dr. Billur Barshan
Temmuz 2009

Bu çalışmada, insan hareketleri vücut üzerinde belirli noktalara minyatür eylemsizlik duyucuları (jiroskop ve ivmeölçer) ve manyetometre konumlandırılarak örüntü tanıma yöntemleriyle ayırdedilmiştir. Ayırdetme işlemi için kural-tabanlı bir yöntem (karar ağacı), en küçük kareler, k-en yakın komşuluk, dinamik zaman bükmesi ve destek vektör makinesi yöntemleri kullanılmıştır. Tezin ilk kısmında bir deneğin bacağına takılan tek eksenli iki jiroskoptan elde edilen sinyallerin işlenmesiyle sekiz farklı bacak hareketi ayırdedilmiştir. İkinci kısımda denek üzerinde beş farklı noktaya konumlandırılan duyucu birimleri insan hareketlerini sınıflandırmak için kullanılmıştır. Her duyucu biriminin içerisinde birer adet üç eksenli jiroskop, üç eksenli ivmeölçer, üç eksenli manyetometre bulunmaktadır. Duyucu sinyalleri kullanılarak elde edilen öznitelikler ayırdetme işleminde kullanılmıştır. Farklı öznitelik vektör kümeleri oluşturulmuş, bu öznitelik vektörlerinin boyutu bazı durumlar için asal bileşenler analizi yöntemiyle küçültülmüştür. Üç farklı çapraz geçerlilik (çapraz doğrulama) yöntemi kullanılmış ve bunların sonuçları birbirleriyle karşılaştırılmıştır. Kullanılan ayırdetme yöntemlerinin doğru ayırdetme yüzdeleri, karışıklık matrisleri, eğitme süreleri ve sınıflandırma süreleri karşılaştırmalı olarak sunulmuştur.

Kullanılan ayırdetme yöntemleri içinde destek vektör makinesi yöntemi en yüksek ayırdetme oranını vermiştir, bunu k-en yakın komşuluk yöntemi izlemiştir. En kısa sınıflandırma süresine karar ağacı yöntemi sahiptir, ardından sırasıyla destek vektör makinesi veya en küçük kareler, k-en yakın komşuluk veya dinamik zaman bükmesi birinci yaklaşım, dinamik zaman bükmesi ikinci yaklaşım yöntemleri sıralanabilir. En uzun eğitme süresi destek vektör makinesi yöntemi için hesaplanmış olup, en uzun sınıflandırma süresine de dinamik zaman bükmesi ikinci yaklaşım yöntemi sahiptir. Kullanılan farklı çapraz doğruluk yöntemlerinin başarı yüzdeleri arasında önemli bir fark gözlemlenmemiştir. Çapraz doğruluk yöntemleri içinde yinelenen rasgele alt-örnekleme yönteminin sınıflandırma süresinin en kısa olduğu görülmüşken, bir-taneyi-dışarıda-bırak yönteminin sınıflandırma süresi en uzundur.

Anahtar Kelimeler: eylemsizlik duyucuları, jiroskop (dönüölçer), ivmeölçer, manyetometre, hareket tanıma, hareket sınıflandırma, örüntü tanıma, öznitelik, asal bileşenler analizi, çapraz geçerlilik, kural-tabanlı ayırdetme, karar ağacı, en küçük kareler, k-en yakın komşuluk, dinamik zaman bükmesi, destek vektör makinesi.

ACKNOWLEDGMENTS

I would like to express my gratitude to my supervisor Prof. Dr. Billur Barshan for her guidance support, and encouragement throughout the development of this thesis.

I would like to express my special thanks and gratitude to Prof. Dr. A. Enis Çetin and Assist. Prof. Dr. Selim Aksoy for showing keen interest in the subject matter and accepting to read and review the thesis.

I would also like to thank my friends Kerem Altun, Hakan Tuna, İbrahim Onaran for their sincere help.

Contents

List of Figures

List of Tables

Dedicated to my family...

Chapter 1

INTRODUCTION

Inertial sensors are self-contained, nonradiating, nonjammable, dead-reckoning devices that provide dynamic information through direct measurements. It is essential to describe, interpret, and classify the outputs of inertial sensors sufficiently accurately if the information is to be used effectively. Fundamentally, gyroscopes provide angular *rate* information about an axis of sensitivity. Similarly, accelerometers provide linear or angular velocity *rate* information. Although the rate information is reliable over long periods of time, it must be integrated to provide absolute measurements of orientation, position and velocity. Thus, even very small errors in the rate information provided by inertial sensors cause an unbounded growth in the error of integrated measurements. As a consequence, the output of inertial sensors are characterized by position errors that grow with time and distance unboundedly. One way of overcoming this problem is to periodically reset the output of inertial sensors with other absolute sensing mechanisms and so eliminate this accumulated error. Thus, techniques of fusing inertial sensor data with other sensors such as GPS, vision systems, and magnetometers have been widely adopted [1, 2, 3].

For several decades, inertial sensors have been used in various applications such as navigation of aircraft [4, 5, 6], ships, land vehicles and robots [7, 8, 9], state estimation and dynamic modeling of legged robots [10, 11], automotive industry, shock and vibration analysis, telesurgery, etc [12, 13].

Inertial sensing systems have become easy to design and carry as the size and cost of inertial sensors have decreased considerably with the rapid development of micro electro-mechanical systems (MEMS) [14]. Small, lightweight, low-cost miniature inertial sensors (gyroscopes, accelerometers, inclinometers or tilt sensors) are increasingly being made commercially available. Some of these devices are sensitive about a single axis; others are multi-axial (usually 2- or 3-axial). For example, the device illustrated in Figure 1.1 and used in the second part of this study combines miniature gyroscopes, accelerometers, and magnetometers in a small box to provide three-dimensional (3-D) drift-free acceleration (up to $18g$), rate of turn, and earth magnetic field information. For low-cost applications that utilize MEMS-based gyros, gyro calibration generally provided by high-end commercial gyros is a necessary but complicated procedure (requiring an accurate variable-speed turntable). Development of accelerometer-based systems is widely adopted because accelerometers are low cost and easily calibrated by gravity. These devices are being put into use in many different applications, human activity monitoring, recognition, and classification being one of them.

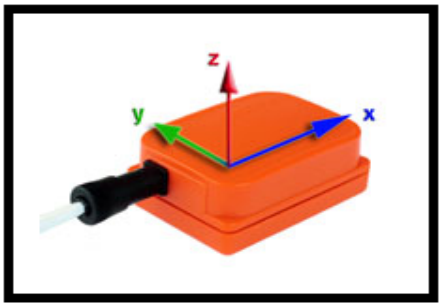


Figure 1.1: MTx 3-DOF orientation tracker (adopted from `http://www.xsens.com/en/general/mtx`).

Tracking and classification of human activities through the use of miniature inertial and magnetic sensors has a broad range of applications: Observation of

elderly people remotely by personal alarm systems [15], home-based rehabilitation and physical therapy [16], medical diagnosis [17], ergonomics [18], sports [19], ballet and dance [20], animation film making and computer games [21, 22].

1.1 Earlier Work on the Use of Inertial Sensors in Human Activity Recognition

We provide a review of the state-of-the-art in the use of body-fixed inertial sensors in activity monitoring, recognition, and classification. In a recent paper that reviews this area [23], the related applications of body-fixed motion sensors are categorized as: estimating activity level and related energy expenditure, activity monitoring, fall detection, and the assessment of balance and gait. Another reference [24] also considers the detection of postural sway, and sit-to-stand transfers as different categories. We limit our literature survey to papers published mostly in the areas of activity recognition, monitoring, and classification, and fall detection. Reference [25] reviews the use of such sensors in motion analysis, in a more general sense than our scope.

Reference [26] provides results of energy expenditure levels for various daily activities such as sitting, lying down, eating breakfast, and working at a desk. However, activity classification is not addressed in this work. The range of frequencies and amplitudes of common human body movements are provided. The study demonstrates that the integral of the signal magnitude is linearly proportional to energy expenditure.

In [27], the activity context of the user is identified. The activities considered are sitting, standing, walking, running up and down the stairs. Using an accelerometer on the wrist, activities such as writing on a board, typing on a keyboard, or shaking hands can also be identified. A naive Bayes classifier with

running mean and variance features are used. Twelve 3-D accelerometers are employed just above the ankle, just above the knee, on the hip, on the wrist, just above the elbow, and on the shoulders.

In [28], five wireless bi-axial accelerometers are used to recognize everyday activities such as free walking, walking while carrying items, working on a computer, sitting and relaxing, standing still, eating and drinking, watching TV, reading, running, bicycling, stretching, strength training, scrubbing, vacuuming, folding laundry, lying down, brushing teeth, climbing stairs, riding the elevator, riding an escalator are considered. Accelerometers are worn on the right hip, non-dominant thigh, non-dominant upper arm, dominant ankle, and dominant wrist of each subject. Acceleration data from 20 subjects are collected under laboratory and semi-naturalistic conditions. Data is labeled by the users. Windows of 512 samples, with 256 overlapping samples between consecutive windows are employed for feature extraction. The mean (average-value) of the signals is removed. The signal features considered are the mean value, correlation between acceleration signals, total energy, and frequency-domain entropy. Four different classifiers (decision table, instance-based learning, decision tree, naive Bayes classifier) are used and among these classifiers, decision trees result in the best activity recognition rate.

In [29], physical activities such as walking, standing, sitting, lying down, bicycling, ascending and descending the stairs are considered. Three single-axis accelerometers are used: one tangential and one radial on the sternum, one tangential on the thigh. To detect if the activity is static or dynamic, a high-pass filter, a rectifier, and a low-pass filter are used to process the accelerometer signals. Static activities are detected from the orientation of the sensors, whereas dynamic activities are detected using the mean, standard deviation, cycle time and signal morphology. The latter is determined from the cross-correlation coefficients with template signals.

The work reported in [30] classifies motion using artificial neural networks. Two accelerometers are used on the legs, and a bi-axial one is used on the sternum. This study suggests using a collection of selected functions on windows of 16 samples. The functions considered are different combinations of the mean, standard deviation, sine, cosine, Fourier transform, cumulative sum, norm, inner product, and outer product, from which 160 different features are generated. Six of them are selected by trial and error and used on the signal windows.

In [15], activities are classified using binary decision trees, arranged in a hierarchical structure. For example, first resting and activity are distinguished, then walking, sitting, standing, lying down etc. in a hierarchical manner. For each binary decision, algorithms such as simple thresholding and pattern matching are employed. Details on the signal processing aspects of this work appear in [31]. A single tri-axial accelerometer mounted on the waist is used on 26 healthy subjects. In this work, suggestions for developing a generic classifier is proposed. In [32], processing is done in real-time using the methods proposed in [15]. The following activities are classified: lying down, slow walking, normal walking, fast walking, sit-to-stand, stand-to-sit, lying-to-sitting, sitting-to-lying, active fall, inactive fall, chair fall. Some performance evaluation of the hardware is presented in this work as well.

In [18], recognition of daily activities such as lying, sitting, standing, walking, Nordic walking, running, rowing, and cycling are considered. Sensors such as accelerometers, magnetometers, speech sensors, and light intensity sensors are considered and accelerometers are found to be the most useful. Classifiers used are a custom-designed decision tree, an automatically-generated decision tree, and an artificial neural network. During the measurements, test people are asked to follow a scenario to perform activities at different locations in two hour measurement sessions. The best results are obtained with automatically generated decision tree. The features considered are the mean, variance, median, skewness,

kurtosis, 25 and 75 percentiles using a sliding window. Also, frequency-domain features such as spectral centroid, spectral spread, estimation of the location and the power of the frequency peak, and signal power in different frequency bands are employed. Six features are selected for classification: peak frequency of up-down chest acceleration, median of up-down chest acceleration, peak power of up-down chest acceleration, variance of back-forth chest acceleration, sum of the variances of 3-D wrist accelerations, power ratio of frequency bands 1–1.5 Hz and 0.2–5 Hz measured from left-right magnetometer on chest. After classification, a median filter is used to remove activities with short duration.

In [33], accelerometers, audio sensors, barometric pressure, humidity, and temperature sensors, visible, infrared and high-frequency light sensors, and a compass are combined in one unit. The first three sensor types have turned out to be most useful. Two subjects have worn the device for six weeks. Over 600 features have been extracted and the features have been ranked to select the top 50. Static classifiers such as naive Bayes and decision stumps have been used. Temporal smoothness is achieved by using hidden Markov models. Activities such as sitting, standing, walking, jogging, ascending and descending stairs, riding a bicycle, driving car, riding elevator down, riding elevator up have been differentiated with a correct differentiation rate of 95%. The method used in [34] is the same as in [33]. Data from four different locations on the body are considered to train a general purpose classifier. First, data from N randomly selected individuals out of 12 are used for training, and all 12 were used for testing. Then, only the unused data are used for testing where 80–85% correct decision percentage is achieved.

In [35], the authors propose an activity recognition system primarily for elderly people that can classify nine daily activities: Sit-to-stand, stand-to-sit, lying, lying-to-stand, stand-to-lying, walking, running, sitting and falling down. The activity recognition system consists of three modules: 3-D accelerometer

that is worn on the left side of the waist, a gateway for transferring sensor data to personal digital assistant (PDA) and a PDA phone that makes motion classification. A neural network classifier is used that results in 95.5% overall success rate for five male and two female subjects.

Reference [16], argues the advantages of a wearable body area network (WBAN) of physiological sensors for monitoring the human body continuously. At the first step of this WBAN system, there are sensor nodes to monitor the human body. Each wireless node has one of the following sensors: Accelerometer, gyroscope, ECG, EMG, EEG, blood pressure sensor, tilt sensor, breathing sensor, "smart sock" sensor. Their system can be used for computer-assisted orthopedic rehabilitation of cardiac patients at the recovery stage that is needed after a hip/knee operation or for home-based rehabilitation of patients for saving money and time.

In [36], the authors implement a system that has two single-axis gyroscopes and a two-axis accelerometer which is worn under the foot. They analyze human foot motion during walking and they divide a normal walking gait cycle into four different phases: stance, toe-rotation, swing and heel-rotation. These phases and the transitions between these phases are identified.

In Reference [37], two tri-axial gyroscopes that are attached to the belt of the subject are employed to classify four different actions: walking upstairs, walking downstairs, level walking and starting/stopping walking. Principal component analysis (PCA) and independent component analysis (ICA) are used for feature extraction. The features are used as input to the discrete wavelet transform. Three different data sets are composed: one by using PCA, one by using ICA and the last one is decimated data samples. After that, they compare success rates that are obtained for these three sets. The results indicate that the use of PCA and ICA in the feature generation improves recognition success rate significantly, but the difference between the results of PCA and ICA is negligible.

In [20], authors place wireless sensor modules at the wrists and ankles of three subjects (dancers). Each sensor module (node) includes three orthogonal gyroscopes, three orthogonal accelerometers and a capacitive sensor for measuring node-to-node proximity. They make preliminary experiments on three dancers by looking at cross-covariance of the inertial data throughout small-sized windows. Based on this information, they determine which dancers are synchronized, which one is leading or lagging. Also the variance of inertial data is observed again throughout small-sized windows to understand if there is an increase or decrease in the general trend of activities.

In [38], six activities (sitting, standing, walking, ascending stairs, descending stairs, running) are classified. Only one sensing platform that contains four sensors are used: a dual axis accelerometer, a light sensor, a temperature sensor and a microphone. This sensing platform is placed at different parts of body: at the belt, shirt pocket, trouser pocket, back pocket, and necklace. Each one of these six activities is repeated for every position of the sensor platform. It is found out that every one of these six positions of the sensor platform gives good results for recognition of walking, standing, sitting and running. Ascending and descending the stairs is mixed up with walking for all of the six sensor platform positions and the recognition rate is not satisfactory.

In [39], the authors propose a body-worn wireless sensor system to detect suspicious human activities for security applications such as identifying terrorist activities. The implemented system consists of two phases: The first phase of the system has a one-class SVM classifier to recognize only normal human activities. The activities that cannot be recognized as normal at the first phase are passed on to the second phase of the system. At the second phase, suspicious activities are examined by using the collection of abnormal activity models that are adapted using kernel nonlinear regression.

Reference [24] overviews accelerometer-based systems and their application areas on monitoring of human motion. A brief summary about the type of accelerometers and the requirements about accelerometers that can be used to monitor human movements is given. Many references on the usage of accelerometer-based systems on human body at different areas are provided such as: Measurement of metabolic energy expenditure, assessment of physical activities, measurement of balance and postural sway, gait analysis, sit-to-stand transfers, falls and movement classification. Authors conclude that monitoring of human movements by using accelerometers can be used in applications such as clinical assessment, event monitoring and longitudinal monitoring.

Reference [40] presents a method to detect falls using a tri-axial accelerometer embedded in a hearing-aid housing mounted behind the ear. Experiments were performed on a single subject where the subject intentionally performed the fall. Experiments were attempted with an elderly subject for unintentional falls but during the period of the experiment, no unintentional falls occurred. This study proposes three threshold values for detecting falls: one on horizontal plane acceleration, one on 3-D velocity, and one on 3-D acceleration. The reason that velocity is also used is because acceleration triggers many false alarms, especially during posture changes. The proposed algorithm is too specific and cannot be generalized easily.

Reference [41] provides a review of fall definitions, methods of identifying falls, the details of the recorded signals and the methods of analysis. The paper concludes that there is a large variation in the literature, and suggests standardizing definitions and the details of the recorded signals.

Most of the earlier studies have focused on classification of activities in a non-systematic manner. The research undertaken by different parties are uncoordinated and exhibit a piece-wise collection of results that are difficult to

synthesize into the kind of broader understanding that is necessary to make substantial progress. Most previous studies can distinguish between sitting, lying and standing [15, 28, 29, 30, 42, 43, 44, 45, 46], since these postures are relatively easy to detect using the static component of acceleration. Distinguishing between walking, ascending and descending stairs has also been performed [28, 29, 46], although not as successfully as detecting postures. The configuration, number, and type of sensors differ widely in the different studies, from using a single accelerometer [15, 32, 47] to as many as twelve [27] on different parts of the body. To the best of our knowledge, a universally-accepted method for finding the optimal configuration, number, and type of sensors does not exist [28].

1.2 Earlier Work on the Use of Camera Systems in Human Activity Recognition

A more commonly used approach in human activity recognition and classification is the employment of single or multiple video camera systems. Vision-based analysis of human motion is one of the most fundamental problems in computer science and engineering because of its vast application areas. The applications of vision-based analysis has been classified into three groups [48, 49]:

- surveillance applications, which include people counting, crowd flux analysis, and security issues such as detection and analysis of abnormal behavior in crowded areas.
- control applications, which include motion capture applications such as games, animations and human-computer interfaces.
- analysis applications, which include clinical studies for diagnostics and rehabilitation, as well as performance analysis, evaluation and improvement for athletes.

A number of surveys about vision-based systems that accept different taxonomies for human motion analysis appear in the literature [48, 49, 50, 51]. A more general framework is presented by [50] which classifies such systems as detection, tracking, and recognition systems. This classification defines the tasks to be performed sequentially according to the natural procedure, i.e., the systems first detect the human in the images (low-level processing), then tracks the observed motion (intermediate-level processing) and then performs recognition (high-level processing) according to the tracked motion.

A classical approach used in motion recognition is template matching [52]. Bobick and Davis proposed using Motion Energy Image (MEI) and Motion History Image (MHI) to perform recognition of different aerobics motions, by comparing the extracted data with pre-stored motion templates. Wang *et al.* [53] also use the idea of template matching on still images for clustering, in order to distinguish the image context between figure skating, basketball and baseball.

State-space approaches are widely used in motion recognition, and the mostly used models are Hidden Markov Models (HMM). In previous work, HMMs are both used with low-level image features [54] or with high-level action classes [55]. In [54], a method based on entropy minimization is proposed in order to detect abnormal behavior. In [55], various behaviors in medium-resolution tennis videos are classified using high-level features such as action classes. HMM is used to model the sequence of actions which form the classified behavior. Leo *et al.* [56] use a discrete HMM to classify between four kinds of activities: walking, probing the soil with a stick, damping the ground with a tank and picking up objects from the ground. Shi *et al.* [57] implement a Dynamic Bayesian Network to distinguish two activities (reading and calling someone by phone) and compare the performance with a HMM-based model. In [58], an automatic model selection based approach is proposed to model complex activities of multiple objects such

as shopping activity and aircraft cargo loading/unloading activity. A Dynamically Multi-Linked Hidden Markov Model (DML-HMM) is developed to find out correlations among events. It is demonstrated that performance of DML-HMM is better for modeling group activities in a cluttered and noisy scene when compared with Dynamic Probabilistic Networks (DPNs), Parallel Hidden Markov Model (PaHMM) and Coupled Hidden Markov Model.

Ribeiro and Santos-Victor [59] implement a Bayesian classifier to distinguish between five classes: active, inactive, walking, running, and fighting. The likelihood functions are modeled as mixtures of Gaussians and expectation-maximization (EM) method is used for training. Different feature combinations are explored and evaluated as well.

Rittscher *et al.* [60] demonstrate the problems about the contour tracking method for marking the outline of a person in an image sequence. They represent the image sequence as a space-time x-y-t cube, and classify between running, skipping and walking using spatio-temporal features extracted from the cube.

In a recent study by Ramasso *et al.* [61], transferable belief models are used for human action recognition in athletics sports videos. The database is composed of 33 athletics videos. Three different actions (running, jumping, and falling) are distinguished in four athletic jumps (pole vault, high jump, triple jump and long jump). The proposed model is also compared to Bayesian Networks.

The use of camera systems may be acceptable and practical when the activities are confined to a limited area such as certain parts of a house or office environment and when the environment is well illuminated. When the activity involves going from place to place (such as riding a vehicle, traveling, going shopping, going outside etc.) camera systems are not so practical. Furthermore, camera systems interfere more with the privacy of the people involved and supply additional, unnecessary information. Besides activity monitoring, they also

provide unnecessary information about the surroundings, other people around, appearance, facial expression and body language, personal preferences of the person(s) involved.

When a single camera is used, the 3-D scene is projected on to a 2-D one where significant information loss occurs. For example, some points of interest (which are typically pre-identified by using special markers on the body such as light-emitting diodes (LEDs)) may be occluded by human body parts or objects in the surroundings. This is circumvented by providing multiple 2-D projections from a number of cameras positioned in the environment in order to reconstruct the 3-D scene. A major disadvantage of using camera systems is that computational complexity of processing and developing algorithms for 2-D signals is much higher than dealing with 1-D signals. 1-D signals from inertial sensors can directly provide the required information.

The approach taken in this work is the use of miniature inertial sensors positioned on different parts of the human body to provide direct measurement of motion. The use of camera systems and inertial sensors are two inherently different approaches that do not exclude each other and can be used in a complementary fashion in many situations. Examples of combining or fusing information from these two sensor modalities are provided in the next section.

1.3 Review of Earlier Work on the Joint Use of Inertial and Visual Sensors in Human Activity Recognition

In a number of studies, accelerometers are used together with video camera systems, mostly for comparison purposes. Some examples are summarized below:

One study that checks the validity of accelerometer data by comparing it with video data is reported in [42]. The activities considered are lying on the back, lying on the side, lying prone, standing, sitting, movement-related activity. The subjects are males with and without transtibial amputation. Four accelerometers are used, two on thighs, two on lower part of the sternum. The study uses the orientation of accelerometers and the gravity component of acceleration for activity detection.

In [43], the results are also compared with video monitoring. The activities considered are: lying down, standing, sitting, dynamic motion, and other. Two accelerometers are used, one being on the chest, mounted in the same direction as gravity, and one on the rear of the thigh, mounted in the direction perpendicular to gravity. The acceleration data is first low-pass filtered, then median and mean absolute deviation were calculated over 1 sec intervals using a sampling frequency of 10 Hz. It is shown that median and the mean absolute deviation are less sensitive to outliers than the mean and the standard deviation.

In [62], position and orientation estimation and tracking is studied using inertial and magnetic sensors positioned on the arms and legs. A standard Kalman filter is used for sensor data fusion. A camera system and markers are used for comparison and testing of the methodology.

In [63], it is suggested that there is a correlation between sit-to-stand, stand-to-sit transitions and fall risk in the elderly. Activities considered are: walking, lying down, and sit-to-stand, stand-to-sit, sit-to-lie transitions. Experiments are performed on 11 elderly people at a gait laboratory and their correlation with fall risk is studied. A single gyroscope is used and a camera system is employed as reference. The discrete wavelet transform (DWT) is used for signal analysis. Reference [44] reports the extension of this study where three experiments are performed. The first experiment is basically the same as in [63]. In the

second experiment, postural transitions on 24 hospitalized elderly people are detected. In the third experiment, daily physical activities of 9 elderly people are detected. One "kinematic sensor," consisting of two accelerometers and a gyroscope mounted on the chest are used. The signals are again processed using the DWT.

In some studies, visual sensors are not only used in a supplementary fashion or as a reference basis, but their data is actually integrated or fused with the inertial data. Visual and inertial sensing are two sensing modalities that can be explored to provide robust solutions in human activity monitoring, recognition, and classification. Fusion of information from these two modalities increases the capabilities of intelligent systems and enlarges the application potential of vision systems. These two sensing modalities have complementary characteristics and can cover the limitations and deficiencies of each other: Inertial sensors have large measurement uncertainty at slow motion and lower relative uncertainty at high velocities. They can measure very large velocities and accelerations. On the other hand, cameras can track features very accurately at low velocities. With increasing velocity, tracking accuracy decreases since the resolution must be reduced to accommodate a larger tracking window for the same pixel size and a larger tracking velocity. The fusion of visual and inertial sensor outputs has attracted significant attention recently, due to its robust performance and wide potential application. Two workshops on this topic have taken place in the recent years [64] and selected papers from the 2005 workshop have been published in a journal special issue [65].

In humans and animals, the vestibular system in the inner ear gives inertial information essential for navigation, orientation, body posture control and equilibrium. In humans, this sensorial system is crucial for several visual tasks and head stabilization. Neural interactions of the human vision and vestibular systems occur at a very early processing stage. The information provided by

the vestibular system is used during the execution of visual movements such as gaze holding and tracking. With the recent development of low-cost, single chip, micromachined inertial sensors, these sensors can be easily incorporated alongside the camera imaging sensor, providing an artificial vestibular system. The noise level of these miniature sensors is not suitable for inertial navigation systems, but their performance is similar to biological inertial sensors and can play a significant role in artificial vision.

One of the earliest works where the integration of inertial and visual information is investigated is [66]. Methods of extracting the motion and orientation of the robotic system from inertial information are derived theoretically but not directly implemented in a real system.

Reference [2] considers the fusion of inertial and visual information for accurate tracking of arm motions. A single tri-axial inertial sensor and a single camera are used. Inertial sensor gives hints to vision on where to search for features. Two data fusion methods are proposed for tracking where the first one is a deterministic technique for simple arm motions and the second one is a probabilistic method, based on the Extended Kalman Filter. The results are compared with commercial marker-based systems.

As already noted, the work done on activity recognition through the use of inertial sensors until now is of limited scope, and mostly unsystematic and ad hoc in nature. Usually, some configuration and some modality of sensors is chosen without strong justification, and empirical results are presented. Processing of the acquired signals is often also ad hoc and relatively unsophisticated. Furthermore, the available literature, viewed as a whole, is rather fragmented and incongruent, and the results are not directly comparable with each other; it is more like a scattered set of isolated results rather than a cumulating body of knowledge that builds on earlier work. A unified and systematic treatment of the subject is essential.

There is a lack of a systematic framework and theoretical models that will guide the research in this area and enable the design of studies and experiments such that proposed systems, methods, and obtained results can contribute and be synthesized into a larger whole. Furthermore, there needs to be theoretical models developed to move beyond the present state of piece-wise results. This would significantly facilitate advancements in this area and more importantly increase the usefulness and applicability of the research.

In this thesis, human activities are differentiated only by using miniature inertial sensors and magnetometers worn on the body. The results of our preliminary studies are published in [67] where a limited number of techniques and a limited number of features were used for motion classification. This study aims at providing a systematic comparison between several methods used for human activity recognition based on their successful differentiation rates and computational costs.

The organization of this thesis is as follows: In Chapter 2, the motions classified in this study are introduced and descriptions of two experiments in which two different data sets are acquired are given. Also feature selection and reduction process is the topic of Chapter 2. In Chapter 3, the classification methods used in this study are reviewed. In Chapter 4, experimental results are presented and discussed. In Chapter 5, conclusions are drawn, potential application areas of the work done in this study are provided and possible directions for future work are given.

Chapter 2

CLASSIFIED MOTIONS, FEATURE EXTRACTION AND FEATURE REDUCTION

2.1 Classified Motions

2.1.1 Leg Motions

In the first part of this thesis, eight different leg motions are classified by using two single-axis gyroscopes that are placed on the right leg of a subject, as described below. These motions are:

1. standing position without moving the legs (Figure 2.1(a)),
2. moving only the lower part of right leg to the back (Figure 2.1(b)),
3. moving both the lower and the upper part of the right leg to the front while bending the knee (Figure 2.1(c)),

4. moving the right leg forward without bending the knee (Figure 2.1(d)),

5. moving the right leg backward without bending the knee (Figure 2.1(e)),

6. opening the right leg to the right side of the body without bending the knee (Figure 2.1(f)),

7. squatting down, moving both the upper and the lower part of the right leg (Figure 2.1(g)), and

8. moving only the lower part of the right leg upward while sitting on a stool (Figure 2.1(h)).

The two gyroscopes are piezoelectric vibratory gyroscopes Gyrostar ENV-05A manufactured by the company Murata (Figure 2.2). The Gyrostar is a small relatively inexpensive piezoelectric gyro originally developed for the automobile market and active suspension systems [68]. The main application of the Gyrostar has been in helping car navigation systems to keep track of turns for short durations when the vehicle is out of contact with reference points derived from the additional sensors. It consists of a triangular prism made of a special substance called "Elinvar", on each vertical face of which a piezoelectric transducer is placed. Excitation of one transducer at about 8 kHz, perpendicular to its face, causes vibrations to be picked up by the other two transducers. If the sensor remains still, or moves in a straight line, the signals produced by the pick-up transducers are exactly equal. If the prism is rotated around its principal axis, Coriolis forces proportionate to the rate of rotation are created.

These devices operate with a supply voltage of 8 to 13.5 VDC. and convert angular velocity information to analog voltage at their output [69]. The output voltage is proportional to the angular velocity around the principal axis of the device and varies between 0.5 to 4.5 VDC. The maximum rate that can be measured with the Gyrostar is $\pm 90^o$/s. An angular velocity of zero (no motion) corresponds to a voltage output of +2.5 VDC. At the maximum angular velocities

(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

Figure 2.1: Eight different leg motions.

of $+90°$/sec and $-90°$/sec, the output voltage becomes 4.5 VDC and 0.5 VDC, respectively. If the angular velocity is larger than the maximum value ($\pm 90°$/sec), saturation occurs at the corresponding voltage level (0.5 VDC or 4.5 VDC) so that the rate and the orientation information become erroneous and need to be reset.

Since these devices are sensitive to rotations about a single axis, the positioning of these sensors should be done by taking their sensitivity axis into account. The two gyroscopes are mounted on the right leg of the subject as illustrated in Figure 2.3. One of the gyroscopes is placed 17 cm above and the other one is placed 15 cm below the right knee. These sensors are placed at a position that their axes of sensitivity are parallel both to the ground and to the human body. By positioning sensors this way, it is expected to benefit from these sensors maximally. Throughout the motions listed above, the left leg of the subject does not move and it steps on the ground. Photos that are taken while performing these motions are given in Figure 2.1.

The block diagram of the experimental setup is given in Figure 2.4. The experimental setup contains two piezoelectric gyroscopes for sensing the leg movements, one multiplexer to multiplex the signals of the two gyros, an 8-bit analog-to-digital (A/D) converter with a sampling frequency of 2668 Hz, and a PC. Data acquired by the A/D converter is recorded on the PC through the parallel port of the computer with a simple interface program that is written in Turbo C++. After acquiring and storing this data, sensor signal processing is done by using MATLAB. Finally, the signals are downsampled by 23 to obtain 116 Hz digital signals.

The eight motions listed above are performed by a male subject in a laboratory environment. Each of the eight different leg motions is performed repetitively during a period of 72 sec. For each leg motion, this 72 sec period is repeated 8 times. At the end, each motion has been performed approximately

Figure 2.2: Murata Gyrostar ENV-05A.

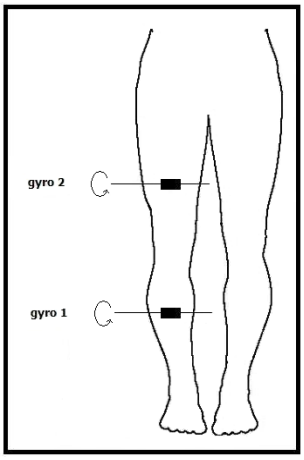


Figure 2.3: Position of the two gyroscopes (body figure is adopted from `http://www.answers.com/body breadths`).

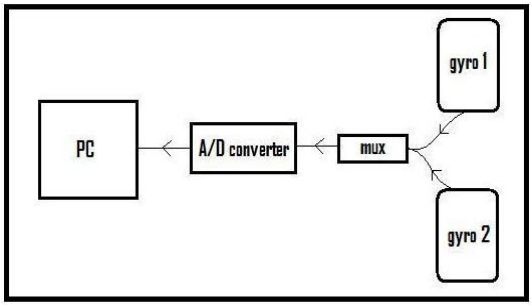
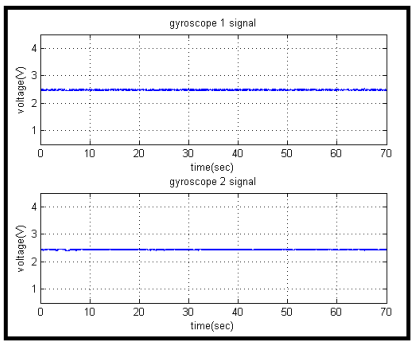


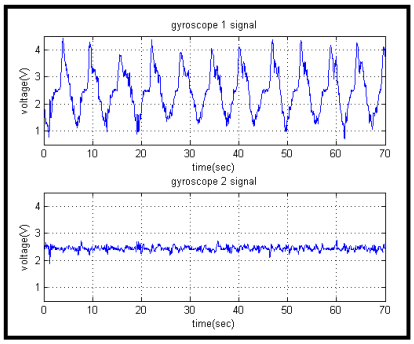
Figure 2.4: Block diagram of the experimental setup.

for about 576 sec. The last 70 sec of each 72 sec signal is used and divided into 10 sec time windows. Hence, while acquiring signals for each motion, a total of $7 \times 8 = 56$ ten second windows are recorded from each gyroscope. Since there are two gyroscopes, $56 \times 2 = 112$ signals are used for each motion.

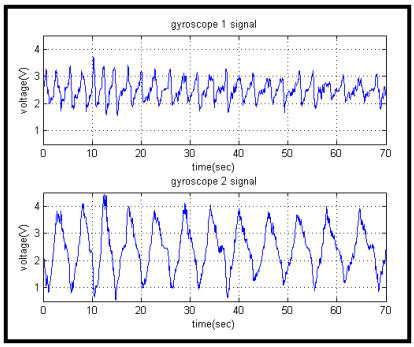
Sample gyroscope signals for eight different leg motions are shown in Figure 2.5 where the quasi-periodic nature of the signals can be observed. This is a sufficient time period to examine the signals since the period of each motion is about 5 to 7 sec.



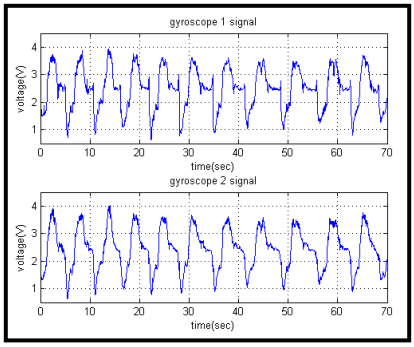
(a) M1



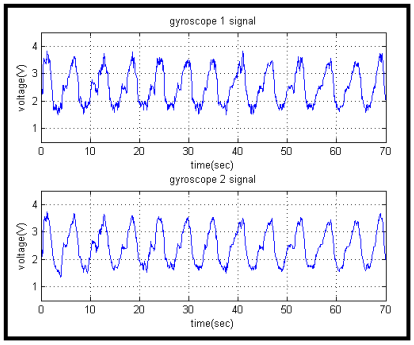
(b) M2



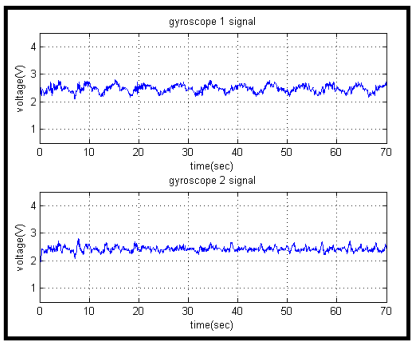
(c) M3



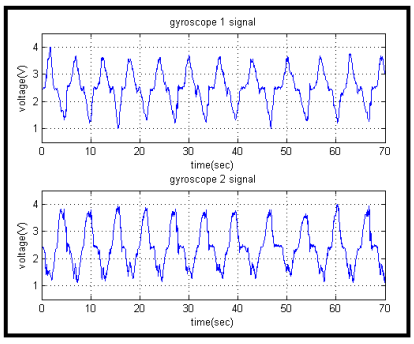
(d) M4



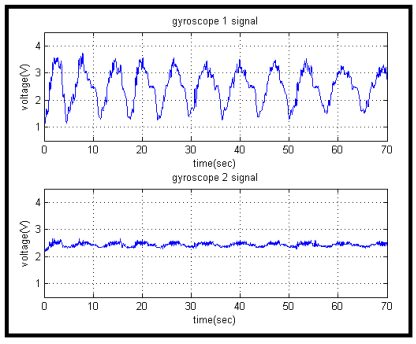
(e) M5



(f) M6



(g) M7



(h) M8

Figure 2.5: Signals of the two gyroscopes (gyro 1 and gyro 2) for the eight different leg motions.

2.1.2 Whole Body Activities

In the second part of this thesis, more complex human activities are classified by using more sophisticated and accurate sensor units. The 19 activities that are classified in this part are:

1. ascending stairs,
2. playing basketball,
3. exercising with cross trainer,
4. cycling with an exercise bike at horizontal position,
5. cycling with an exercise bike at vertical position,
6. descending stairs,
7. standing in the elevator without moving,
8. standing and moving in the elevator,
9. jumping,
10. lying down on back,
11. lying down on right side,
12. rowing,
13. running on a treadmill with 8 km/hr speed,
14. sitting,
15. standing,
16. exercising with stepper,
17. walking at parking lot,

18. walking on a treadmill with 4 km/hr speed, and

19. walking on a treadmill with 4 km/hr speed and with 15° slope.

In this part, five of MTx 3-DOF orientation trackers are used, manufactured by Xsens Technologies (Figure 1.1). Each MTx has a tri-axial accelerometer, a tri-axial gyroscope and a tri-axial magnetometer so that these sensor units produce 3-D acceleration, 3-D rate of turn, and 3-D earth-magnetic field data [70]. Each motion tracker is programmed via an interface program called MT Manager to capture data signals. Sampling frequency of the sensor signals can be chosen as 25 Hz, 50 Hz or 100 Hz.

Accelerometers of two of the MTx trackers can sense up to $\pm 5g$ and the other three can sense in the range $\pm 18g$ where $g = 9.80665$ m/s^2 is the standard gravity. All gyroscopes of the MTx unit can sense in the range ± 1200°/sec angular velocities, magnetometers can sense in the range ± 750 mGauss. We use all of these three types of sensor data in all three dimensions.

These sensors are placed at five different positions on the subject's body. 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 the left side of the left knee), the remaining $\pm 18g$ unit is placed at the chest of the subject, and the two $\pm 5g$ units to the wrists. Positions of the sensor units on the human body can be seen in Figure 2.6. The five MTx units are connected with 1 m length cables to a device called Xbus Master which is attached to the belt of the subject. Xbus Master transmits five MTx's data to the receiver by using a bluetooth connection. Xbus Master which is connected to three MTx orientation trackers can be seen in Figure 2.7. The receiver is connected to a laptop computer via a USB connection. Two of the five MTx units are directly connected to the Xbus Master. Remaining three units have an indirect connection to the XBus

Master through the other two. Figure 2.8 illustrates the connection configuration of five MTx units and the Xbus Master.

(a) (b) (c)

Figure 2.6: Location of Xsens sensor modules on the body.

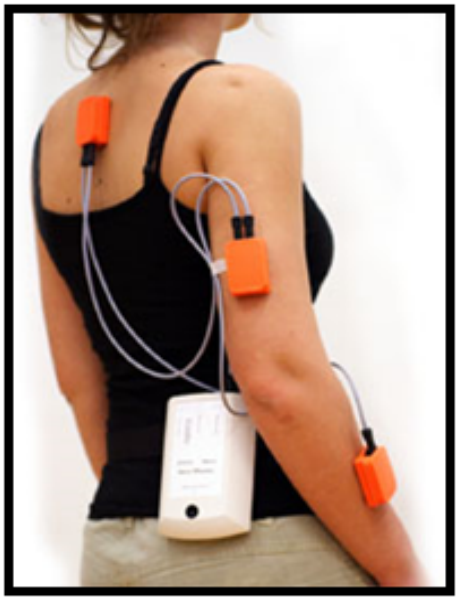


Figure 2.7: Mtx blocks and Xbus Master (adopted from `http://www.xsens.com/en/movement-science/xbus-kit`).

Each activity listed above is performed for 5 min by a male subject. Most of the activities are performed at the Bilkent University Sports Hall, some of them are performed in the Electrical and Electronics Engineering Building, and some of the data are acquired outdoors near Odeon. The 5 min long signals are divided into 5 sec intervals from which features are extracted. Thus, 60 signal

segments are obtained for each activity. Sensor units are calibrated to acquire data with 50 or 100 Hz sampling frequencies for different activities. The 50 Hz signals are downsampled by 2 and 100 Hz signals are downsampled by 4, to get 25 Hz signals. For each activity, 60 feature vectors are obtained.

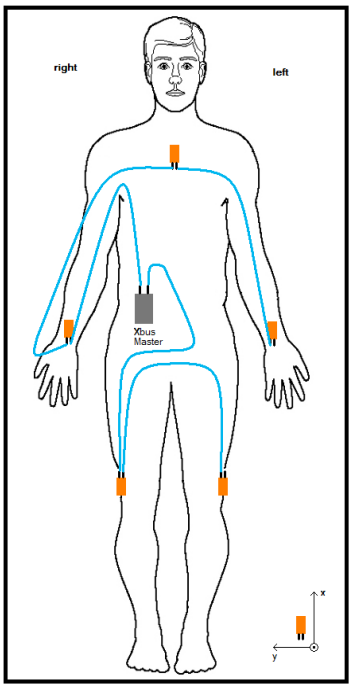
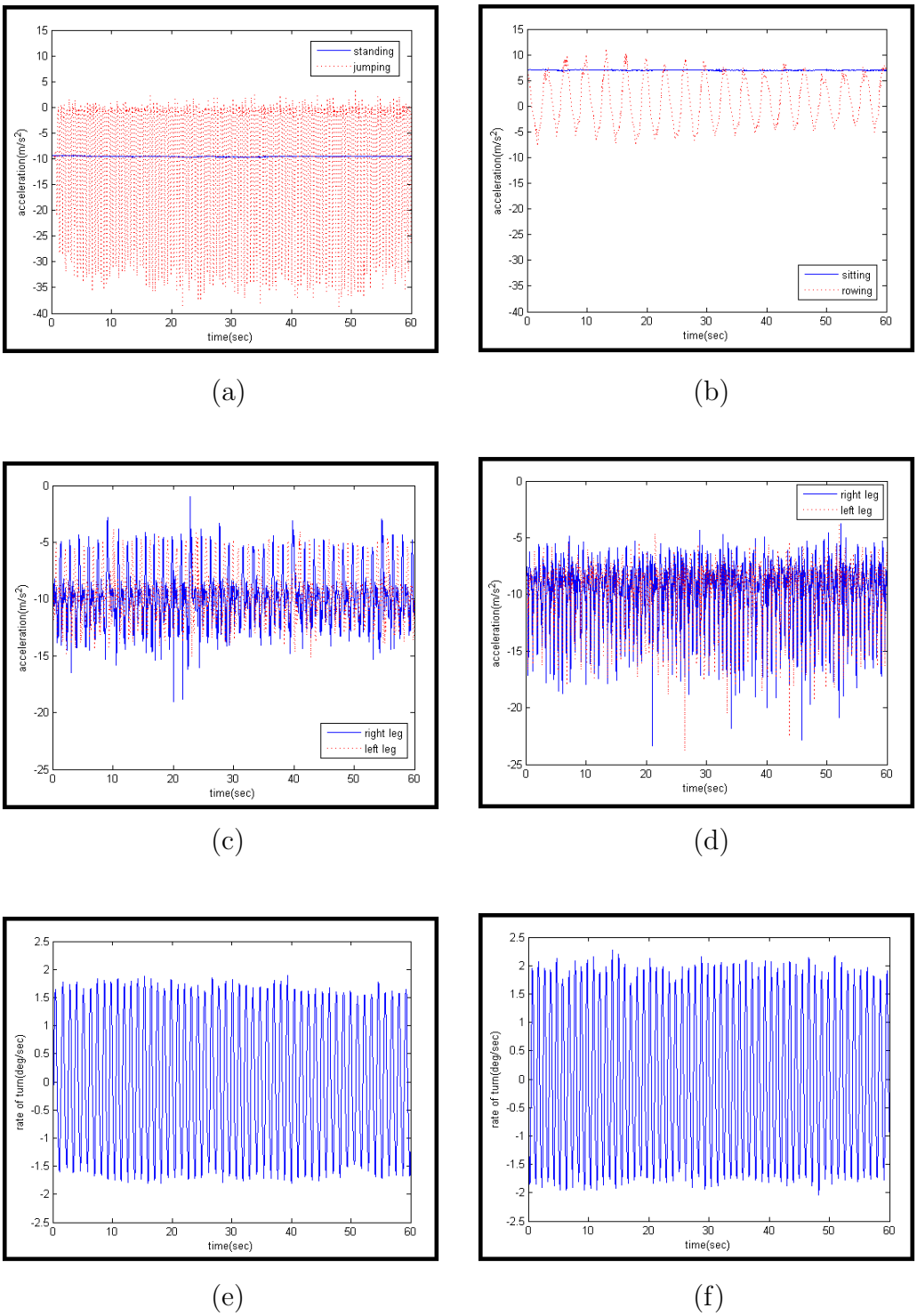


Figure 2.8: Connection diagram of MTx sensor blocks (body figure is adopted from `http://www.answers.com/body breadths`).

Some example signals for 1 min time period are given in Figure 2.9 for accelerometers and gyroscopes that are placed at different parts of the body. In Figure 2.9(a), x-axis accelerometer signal of the right leg for the motions standing and jumping is given. In Figure 2.9(b), z-axis accelerometer signal of the chest for the motions sitting and rowing can be seen. In Figure 2.9(c), x-axis accelerometer signal of the right and left leg for the motion ascending stairs is shown. In Figure 2.9(d), x-axis accelerometer signal of the right and left leg for the motion descending stairs is given. In Figure 2.9(e), z-axis gyroscope signal of the left leg for the motion cycling vertical can be seen. In Figure 2.9(f), z-axis gyroscope signal of the left leg for the motion cycling horizontal is shown.



(a) (b)

(c) (d)

(e) (f)

Figure 2.9: Example signals for human activities.

2.2 Feature Extraction and Reduction

After acquiring the signals as described above, a discrete-time sequence of N_s elements that can be represented as an $N_s \times 1$ vector $\mathbf{s} = [s_1, s_2, \ldots, s_{N_s}]^T$ is obtained. We have considered using features such as the mean value, variance, minimum and maximum values, kurtosis, skewness, autocorrelation sequence, cross-correlation sequence, total energy, peaks of the discrete Fourier transform (DFT) and the corresponding frequencies, and the discrete cosine transform (DCT) coefficients of $\mathbf{s}$. These features are calculated as follows:

$$
\begin{aligned}
\text{mean}(\mathbf{s}) &= \mu = E\{\mathbf{s}\} = \frac{1}{N_s}\sum_{i=1}^{N_s} s_i \\
\text{variance}(\mathbf{s}) &= \sigma^2 = E\{(\mathbf{s}-\mu)^2\} = \frac{1}{N_s}\sum_{i=1}^{N_s}(s_i-\mu)^2 \\
\text{skewness}(\mathbf{s}) &= \frac{E\{(\mathbf{s}-\mu)^3\}}{\sigma^3} = \frac{1}{N_s\sigma^3}\sum_{i=1}^{N_s}(s_i-\mu)^3 \\
\text{kurtosis}(\mathbf{s}) &= \frac{E\{(\mathbf{s}-\mu)^4\}}{\sigma^4} = \frac{1}{N_s\sigma^4}\sum_{i=1}^{N_s}(s_i-\mu)^4 \\
\text{autocorrelation}: \quad R_{\mathbf{ss}}(k) &= \frac{1}{N_s-k}\sum_{i=0}^{N_s-k-1}(s_i-\mu)\,(s_{i-k}-\mu), \\
&\qquad k = 0, 1, \ldots, N_s-1 \\
\text{cross}-\text{correlation}: \quad R_{\mathbf{su}}(k) &= \frac{1}{N_s-k}\sum_{i=0}^{N_s-k-1}(s_i-\mu)\,(u_{i-k}-\mu_u), \\
&\qquad k = -N_s+1, \ldots, 0, \ldots, N_s-1 \\
\text{DFT}: \quad S_{\text{DFT}}(k) &= \sum_{i=0}^{N_s-1} s_i\, e^{-\frac{j2\pi ki}{N_s}}, \\
&\qquad k = 0, 1, \ldots, N_s-1 \\
\text{DCT}: \quad S_{\text{DCT}}(k) &= \alpha(k)\sum_{i=0}^{N_s-1} s_i \cos\left[\frac{\pi(2i+1)k}{2N_s}\right], \\
&\qquad k = 0, 1, \ldots, N_s-1 \\
\text{where} \quad \alpha(k) &= \begin{cases} \sqrt{\frac{1}{N_s}} & \text{for} \quad k = 0 \\ \sqrt{\frac{2}{N_s}} & \text{for} \quad k \neq 0 \end{cases}
\end{aligned}
\tag{2.1}
$$

where, s_i is the i'th element of the discrete-time sequence $\mathbf{s}$, $E\{.\}$ denotes the expectation operator, μ and σ are the mean and the standard deviation of $\mathbf{s}$,

$R_{\mathbf{ss}}(k)$ is the k'th element of the unbiased autocorrelation sequence of $\mathbf{s}$, $R_{\mathbf{su}}(k)$ is the k'th element of the unbiased cross-correlation sequence between $\mathbf{s}$ and $\mathbf{u}$ where μ_u is the mean of $\mathbf{u}$, $S_{\mathrm{DFT}}(k)$ and $S_{\mathrm{DCT}}(k)$ are the k'th elements of the 1-D N_s-point DFT and N_s-point DCT, respectively. DCT is a transformation technique widely used in image processing that transforms the data into the form of sum of cosine functions [71, 72].

2.2.1 Leg Motions

In constructing the feature vectors based on the acquired signals, features of the two gyroscope signals that correspond to the same time interval (signal segment) are included in each feature vector. A total of 101 features are extracted from the signals of the two gyroscopes so that the size of each feature vector is 101×1. For each leg motion, 56 such feature vectors are obtained. The initial set of features is as follows:

1: mean value of gyro 1 signal

2: mean value of gyro 2 signal

3: kurtosis of gyro 1 signal

4: kurtosis of gyro 2 signal

5: skewness of gyro 1 signal

6: skewness of gyro 2 signal

7: minimum value of gyro 1 signal

8: minimum value of gyro 2 signal

9: maximum value of gyro 1 signal

10: maximum value of gyro 2 signal

11: minimum value of cross-correlation between gyro 1 and gyro 2 signals

12: maximum value of cross-correlation between gyro 1 and gyro 2 signals

13-17: maximum 5 peaks of DFT of gyro 1 signal

18-22: maximum 5 peaks of DFT of gyro 2 signal

23-27: the 5 frequencies corresponding to the maximum 5 peaks of DFT of gyro 1 signal

28-32: the 5 frequencies corresponding to the maximum 5 peaks of DFT of gyro 2 signal

33-38: 6 samples of the autocorrelation function of gyro 1 signal (sample at the midpoint and every 25th sample up to the 125th)

39-44: 6 samples of the autocorrelation function of gyro 2 signal (sample at the midpoint and every 25th sample up to the 125th)

45: minimum value of the autocorrelation function of gyro 1 signal

46: minimum value of the autocorrelation function of gyro 2 signal

47-61: 15 samples of the cross-correlation between gyro 1 and gyro 2 signals (every 20th sample)

63-81: first 20 DCT coefficients of gyro 1

82-101: first 20 DCT coefficients of gyro 2

For the 10 sec time windows and the 116 Hz sampling rate, the number of samples of the sequence is $N_s = 1160$. While extracting the features, autocorrelation function has a length of 1160 samples. The minimum value of the autocorrelation function is calculated only by considering the samples between 0–40. The maximum value of the cross-correlation function is calculated by considering the samples between 0–140.

Since the number of initial set of features was quite large (101) and all of the features were not equally useful in discriminating the motions, we reduced the

number of features in several different ways: First, we reduced the number of features from 101 to 14 by inspection, trying to identify the features that result in the highest differentiation rates by trial and error. Then, by additionally applying PCA (see the appendix) to these 14 selected features, we further reduced their number to 6. Thirdly, we chose the 14 features with the largest variances using the covariance matrix of the feature vectors. We also reduced the 101 features to 6 through PCA. Finally, we employed the sequential forward feature selection (SFFS) method. This method adds features one at a time to the classification algorithm such that the classification performance is maximized. A more detailed description of the method can be found in [73]. We used the arithmetic average of the classification rates obtained by the different classification techniques as an objective in order to ultimately determine the reduced feature set.

As a result of this procedure, the following features are selected:

1. minimum value of gyro 2 signal,
2. maximum value of gyro 1 signal,
3. maximum value of the cross-correlation between gyro 1 and 2 signals,
4. 3rd maximum peak of DFT of gyro 2 signal,
5. minimum value of the cross-correlation between gyro 1 and 2 signals, and
6. 3rd maximum peak of DFT of gyro 1 signal.

All of these features are normalized to the interval $[0, 1]$ to be used for classification.

2.2.2 Human Body Activities

In the second part of the study, human activities are classified. There are 5 sensor units (MTx), each with three tri-axial devices so that a total of 9 measurement signals are acquired for every sensor unit. Features are placed in the feature vector in a certain order: When a feature such as the mean value of a signal is calculated, 45 ($= 9 \times 5$) different values are recorded for each feature. These values from the five sensor units are placed in the feature vectors in the following order: right arm, left arm, right leg, torso, and left leg. For each one of these sensor locations, 9 values for each feature are calculated and recorded in the following order: x, y, z axes acceleration, x, y, z axes rate of turn, and x, y, z axes earth magnetic field. In constructing the feature vectors, the above procedure is applied for the mean, skewness, kurtosis, minimum and maximum value features. Thus, 225 ($= 45$ axes $\times$ 5 features) elements of the feature vectors are obtained by using the above procedure.

After applying DFT to the 5 sec windows, the maximum 5 Fourier peaks are selected for each signal. Therefore, for each sensor unit 45 ($= 9 \times 5$) Fourier peaks, and a total of 225 ($= 45$ axes $\times 5$ peaks) Fourier peaks are obtained. Each group of 45 peaks is placed in the order of right arm, left arm, right leg, torso, left leg, as above. The 225 frequency values that correspond to these Fourier peaks are placed after the Fourier peaks in the given order.

11 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 for each 5 sec window, 495 ($= 45 \times 11$) autocorrelation samples are placed in each feature vector. The sample at the center of the autocorrelation function (variance) and every 5th sample up to the 50th sample 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 sec signals and the dimensions of the resulting feature vectors are 1170×1. All of these features are normalized to the interval $[0, 1]$ to be used for classification.

Again, since the number of initial set of features was very large and all of the features were not equally useful and meaningful, we reduced the number of features from 1170 to 8 through PCA. This reduced dimension of the feature vectors is determined by observing the eigenvalues of the covariance matrix of the 1170×1 training vectors. The sorted eigenvalues are shown in Figure 2.10. A zoomed version of this figure can be seen in Figure 2.11 from which it can be observed that the first eight eigenvalues have considerably larger values when compared to the remaining ones and there is a break point around this value. Therefore, only the eight eigenvectors that correspond to these eight eigenvalues are used to form the transformation matrix and 8×1 feature vectors are obtained. However, because of the transformation involved, these feature vectors usually do not have any physical meaning.

In both parts of the study we assume that after feature reduction or selection, the resulting feature vector is an $N \times 1$ vector $\mathbf{x} = [x_1, \ldots, x_N]^T$.

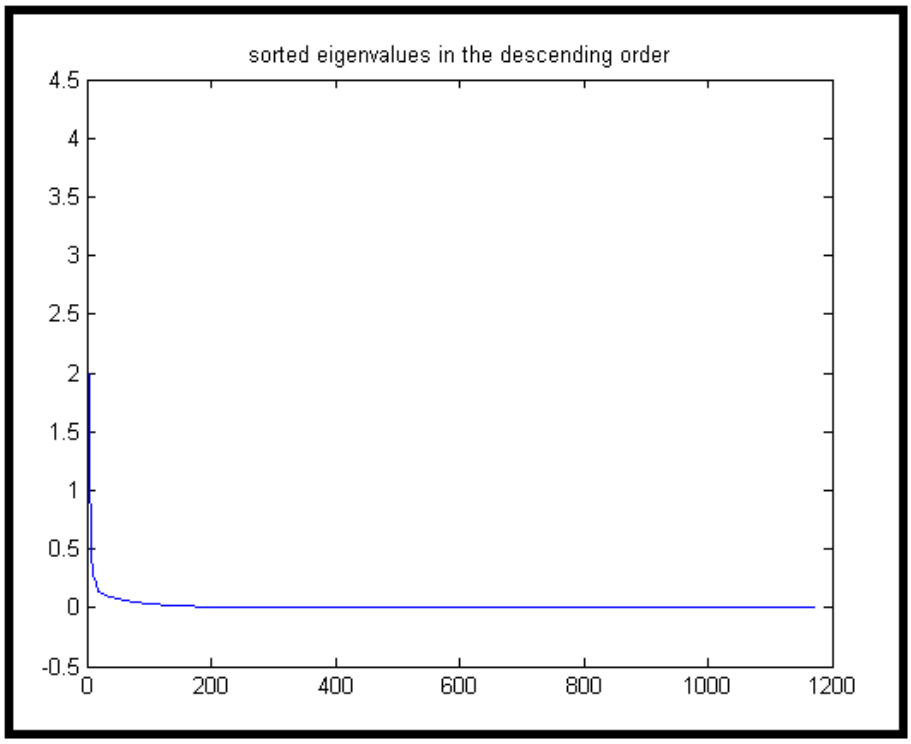


Figure 2.10: 1170 eigenvalues of the covariance matrix in the descending order.

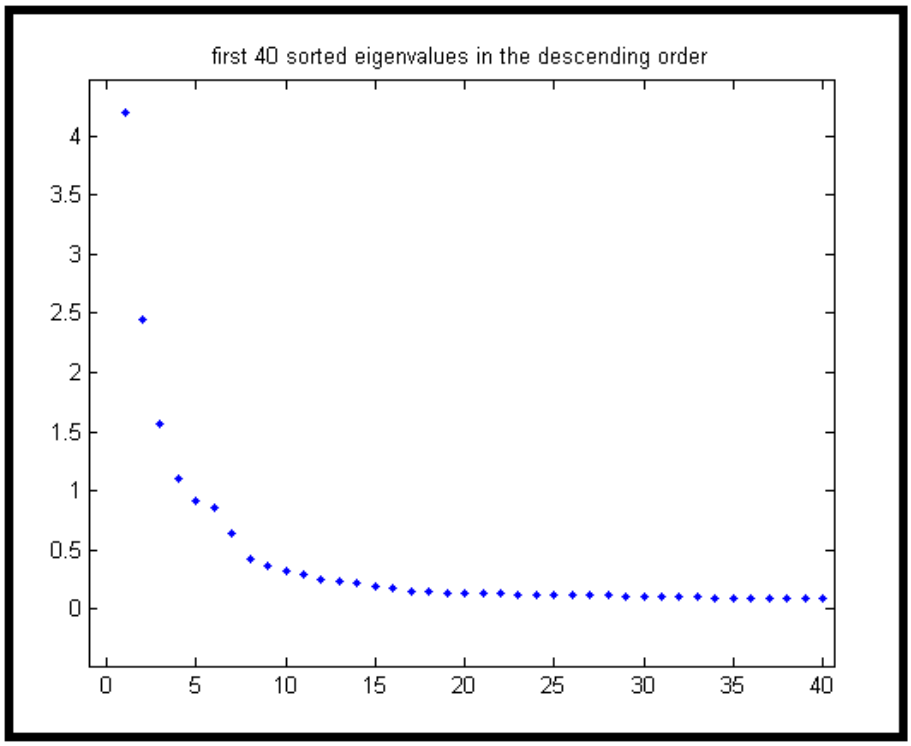


Figure 2.11: First 40 eigenvalues of the covariance matrix in the descending order.

Chapter 3

CLASSIFICATION METHODS

Some of the methods summarized below require a training phase, some do not.

We associate a class w_i with each motion type ($i = 1, \ldots, c$). An unknown motion is assigned to class w_i if its feature vector $\mathbf{x} = [x_1, \ldots, x_N]^T$ falls in the region Ω_i. A rule which partitions the decision space into regions $\Omega_i, i = 1, \ldots, c$ is called a *decision rule*. Each one of these regions corresponds to a different motion type. Boundaries between these regions are called *decision surfaces*. Let $p(w_i)$ be the a priori probability of the motion belonging to class w_i. To classify a motion with feature vector $\mathbf{x}$, a posteriori probabilities $p(w_i|\mathbf{x})$ are compared and the motion is classified into class w_j if $p(w_j|\mathbf{x}) > p(w_i|\mathbf{x}) \quad \forall i \neq j$. This is known as *Bayes minimum error rule*. However, since these a posteriori probabilities are rarely known, they need to be estimated. A more convenient formulation of this rule can be obtained by using Bayes' theorem: $p(w_i|\mathbf{x}) = p(\mathbf{x}|w_i)p(w_i)/p(\mathbf{x})$ which results in $p(\mathbf{x}|w_j)p(w_j) > p(\mathbf{x}|w_i)p(w_i) \quad \forall i \neq j \implies \mathbf{x} \in \Omega_j$ where $p(\mathbf{x}|w_i)$ are the class-conditional probability density functions (CCPDFs) which are also unknown and need to be estimated in their turn based on the training set. The training set contains a total of $I = I_1 + I_2 + \ldots + I_c$ sample feature vectors where I_i sample feature vectors belong to class w_i, and $i = 1, \ldots, c$. The *test set* is then

used to evaluate the performance of the decision rule used. This decision rule can be generalized as $q_j(\mathbf{x}) > q_i(\mathbf{x}) \;\; \forall i \neq j \implies \mathbf{x} \in \Omega_j$ where the function q_i is called a *discriminant function*.

The various statistical techniques for estimating the CCPDFs based on the training set are often categorized as non-parametric and parametric. In non-parametric methods, no assumptions on the parametric form of the CCPDFs are made; however, this requires large training sets. This is because any non-parametric PDF estimate based on a finite number of samples is biased [74]. In parametric methods, specific models for the CCPDFs are assumed and then the parameters of these models are estimated. Parametric methods can be further categorized as normal and non-normal models.

3.1 Rule-Based Algorithm (RBA)

A rule-based algorithm (RBA) or a decision tree can be considered as a sequential procedure that classifies given inputs [75]. A rule-based algorithm follows predefined rules at each node of the tree and makes binary decisions based on these rules. An example of a rule-based algorithm is given in Figure 3.1. At each node, a condition such as "is feature $x_i \leq \tau_i$?" is checked. Here, $i = 1, 2, \ldots, S$ where T is the total number of features that is used in the tree and τ is the threshold value for this feature at the given node [76]. These threshold values are determined by examining the training vectors of all classes. Decision tree algorithms start from the top of the tree and go down to branches by splitting each node to two descendant nodes based on checking conditions similar to above [76]. This process continues until one of the leaves is reached or until a branch is terminated.

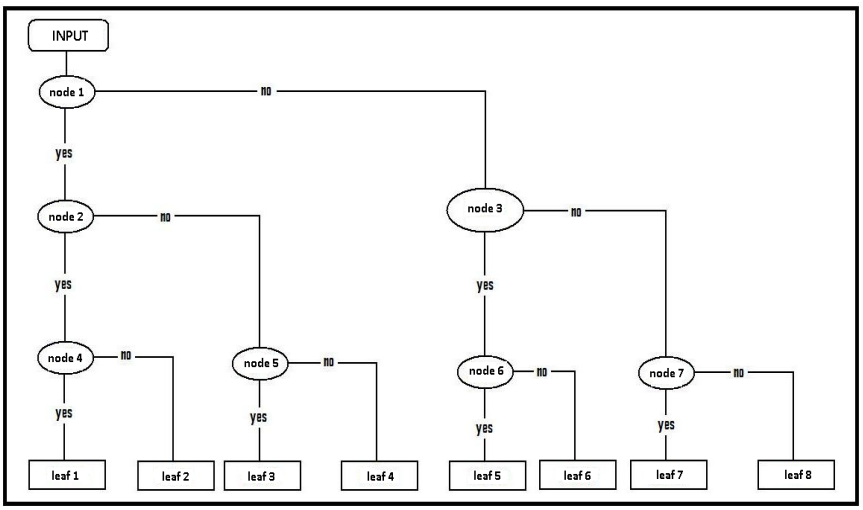


Figure 3.1: Tree structure of the RBA.

More discriminative features are used at the nodes higher up in the tree hierarchy to decrease the misclassification rate. Selection and calculation of features before using them in the rule-based algorithm is an important issue to make the algorithm independent of calculation cost of different features.

The rule-based method has the advantage that it does not require storage of any reference feature vectors since the information necessary to differentiate the motions is completely embodied in the decision rules.

To classify gyroscope signals, some simple rules are generated by using the extracted features. The generated decision tree has 8 leaves (for 8 motions) as expected and 7 decision nodes. These decision nodes are numerated by beginning from the top towards the bottom and from the left towards the right, respectively. These rules are determined by using the normalized values of the features between 0 and 1. Some of these rules are inequalities that compare the value of certain features with a constant value and some of the rules are inequalities that compare the ratio of some features with some threshold. These rules are:

1. `is the variance of gyro 2 signal` < 0.1?

2. `is the variance of gyro 1 signal` < 0.1?

3. is the min value of gyro 1 signal > 0.6?

4. is $\frac{\text{max value of gyro 1 signal}}{\text{min value of gyro 1 signal}} < 0.1$?

5. is $\frac{\text{variance of gyro 2 signal}}{\text{min value of autocorrelation function of gyro 2}} > 1.04$?

6. is max value of cross-correlation function < 0.4?

7. is $\frac{\text{max value of gyro 2 signal}}{\text{min value of gyro 2 signal}} < 1.4$?

The diagram of this algorithm is shown in Figure 3.2.

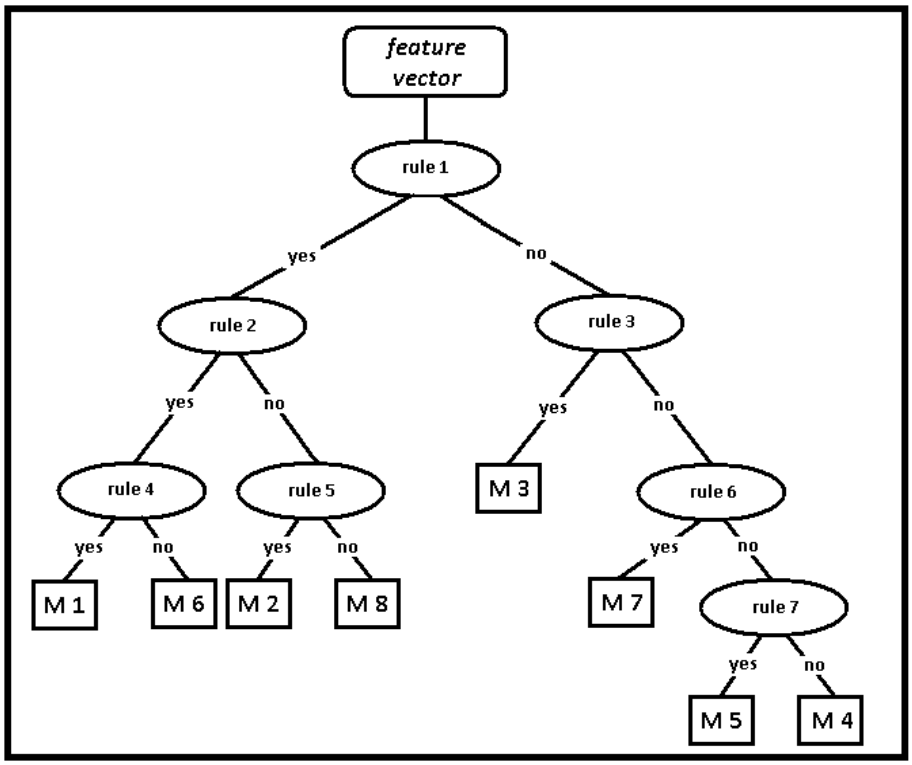


Figure 3.2: RBA for gyroscope data.

To classify human activities, the same approach is used. 18 threshold values are determined to compare sensor signal features. The rules used at these 18 decision nodes are given as follows:

1. is the max value of right leg z-axis gyroscope signal < 0.1?
2. is the mean value of chest z-axis accelerometer signal > 0.95?
3. is the mean value of right leg x-axis accelerometer signal > 0.3?
4. is the mean value of chest y-axis accelerometer signal > 0.95?
5. is the max value of left leg z-axis gyroscope signal > 0.3?
6. is the mean value of right arm x-axis accelerometer signal > 0.4?
7. is the mean value of right leg y-axis accelerometer signal > 0.9?
8. is the mean value of left leg x-axis accelerometer signal > 0.6?
9. is the mean value of right arm y-axis accelerometer signal > 0.1?
10. is the variance of chest x-axis accelerometer signal > 0.5?
11. is the max value of chest z-axis magnetometer signal > 0.2?
12. is the variance of right arm z-axis accelerometer signal > 0.1?
13. is the variance of left leg z-axis accelerometer signal > 0.0001?
14. is the min value of right leg x-axis magnetometer signal > 0.7?
15. is the max value of right arm x-axis accelerometer signal > 0.2?
16. is the min value of chest x-axis magnetometer signal > 0.3?
17. is the max value of right arm x-axis magnetometer signal > 0.1?
18. is the variance of left leg x-axis magnetometer signal > 0.7?

The diagram of this algorithm is shown in Figure 3.3.

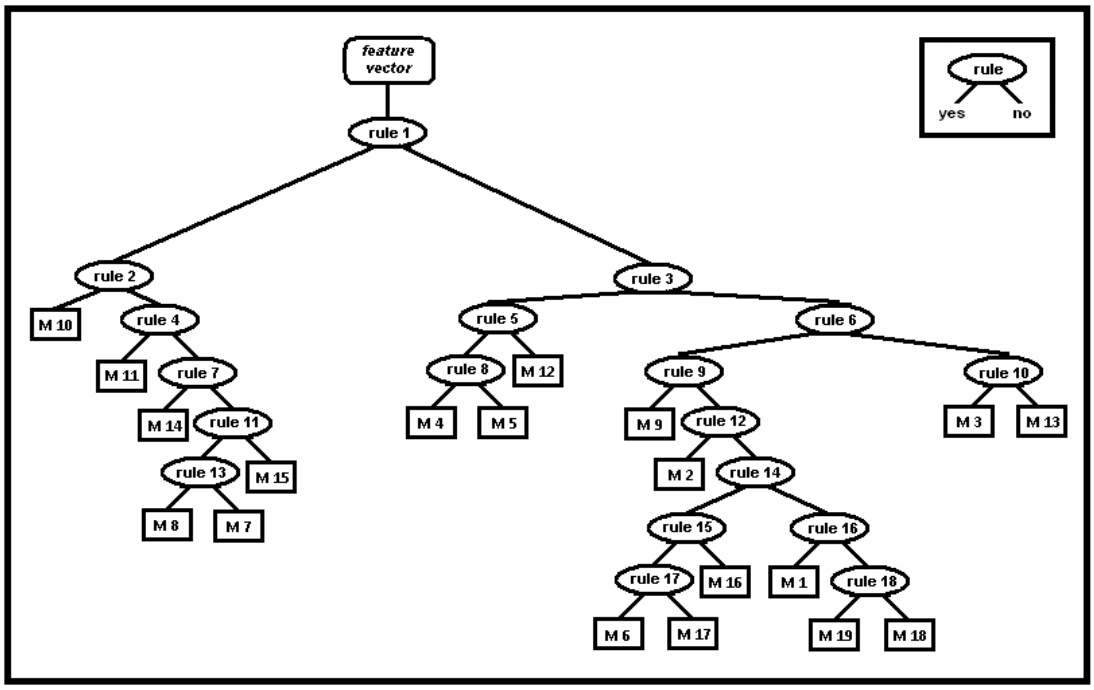


Figure 3.3: RBA for classifying human activities.

3.2 Least-Squares Method (LSM)

LSM is one of the simplest algorithms that can be used for classification. We have implemented LSM in two different ways: In the first approach, each test feature vector is compared with each reference vector stored in the database and the test vector is assigned to the same class as the nearest reference vector. This approach, in fact, corresponds to the k-NN method described below, when k is selected as 1.

In the second approach, 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 by using the following equation:

$$\mathcal{D}_i^2 = \sum_{n=1}^{N}(x_n - r_{in})^2 = (x_1 - r_{i1})^2 + \ldots + (x_N - r_{iN})^2 \quad i = 1, \ldots, c \qquad (3.1)$$

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.

3.3 k-Nearest Neighbor (k-NN) Algorithm

Consider the k nearest neighbors of a feature vector $\mathbf{x}$ in a given set of many feature vectors. The neighbors are taken from a set of feature vectors (the training set) for which the correct classification is known. The occurrence number of each class is counted among these neighbor vectors and suppose that k_i of these k vectors come from class ω_i. Then, a k-NN estimator for class ω_i can be defined as $\hat{p}(\omega_i|\mathbf{x}) = \frac{k_i}{k}$, and $\hat{p}(\mathbf{x}|\omega_i)$ can be obtained from $\hat{p}(\mathbf{x}|\omega_i)\hat{p}(\omega_i) = \hat{p}(\omega_i|\mathbf{x})\hat{p}(bfx)$. This results in a classification rule such that $\mathbf{x}$ is classified into class ω_j if $k_j = \max_i(k_i)$, where $i = 1, \ldots, c$. In other words, 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 [77]. It is common to use the Euclidean distance measure, although other distance measures such as the Manhattan distance could in principle be used instead. The k-NN algorithm is sensitive to the local structure of the data.

For example in Figure 3.4, assume that the square is the test vector, diamonds and stars are the vectors which correspond to two different classes that will be named as class 1 and class 2, respectively. If $k = 4$, the vectors in the inner circle will be considered as the neighbors of the test vector (square), since these four are the nearest neighbors of the test vector. Three of these vectors belong to class 2 and the remaining one belongs to class 1, so the test vector will be classified as a class 2 vector. If $k = 12$, then the class of the nearest 12 vectors are important for the classification of the test vector (square). These 12 vectors can be seen inside the larger circle in Figure 3.4. Seven of these vectors are represented with diamonds (class 1) and the remaining 5 are stars (class 2), so the test vector

(square) will be classified as a class 1 vector. As can be seen from the above example, selection of the parameter k, the number of neighbors considered, is a very important issue for the k-NN classifier. Unfortunately, a pre-defined rule for the selection of the value of k does not exist [78].

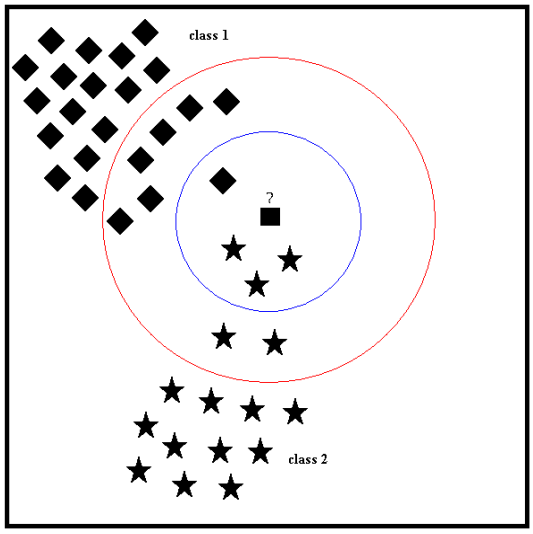


Figure 3.4: An example on the selection of the parameter k in the k-NN algorithm. The inner circle corresponds to $k = 4$ and the outer circle corresponds to $k = 12$, producing different classification results for the test vector.

Assigning the training feature vectors to a predefined class and storing them for distance comparison can be thought of as the training phase of this technique, although no explicit training step is required. Calculating the distances of test vectors to each of the training vectors and selecting those with the the k smallest distances comprises the test phase.

3.4 Dynamic Time Warping (DTW)

DTW is an algorithm for measuring the similarity between two sequences which may vary in time or speed. An optimal match between two given sequences (e.g. time series) is found under certain restrictions. The sequences are "warped" non-linearly in the time dimension to determine a measure of their similarity independent of certain non-linear variations in the time dimension. DTW is

used mostly in finite vocabulary speech recognition to handle different speaking speeds [79, 80]. Besides speech recognition, DTW has been used in word spotting in handwritten historical documents on electronic media [81] and machine printed documents [82], signature [83, 84] and gait recognition [85], ECG signal classification [86, 87, 88], fingerprint verification [89], and face localization in color images [90]. In this study, DTW is used for the classification of feature vectors extracted from inertial sensor and magnetometer signals.

In DTW, the aim is to find the least-cost warping path for the tested feature vector among the stored reference feature vectors [79]. The cost measure is typically taken as the Euclidean distance between the elements of the feature vectors. Given two feature vectors $\mathbf{x}$ and $\mathbf{y}$ with lengths N and M:

$$\begin{aligned} \mathbf{x} &= [x_1, x_2, \ldots, x_n, \ldots, x_N]^T \\ \mathbf{y} &= [y_1, y_2, \ldots, y_m, \ldots, y_M]^T \end{aligned} \tag{3.2}$$

An $N \times M$ distance matrix $\mathbf{d}$ is constructed by using all the elements of the feature vectors $\mathbf{x}$ and $\mathbf{y}$. The (n, m)'th element of this matrix, $d(n, m)$, is the distance between the n'th element of $\mathbf{x}$ and the m'th element of $\mathbf{y}$ and is given by $d(n, m) = \sqrt{(x_n - y_m)^2} = |x_n - y_m|$ [80].

A warping path $\mathbf{W}$ is a contiguous set of matrix elements that defines a mapping between $\mathbf{x}$ and $\mathbf{y}$. Assuming that the l'th element of the warping path is $w_l = (n_l, m_l)$, the warping path $\mathbf{W}$ with length L is given as:

$$\mathbf{W} = w_1, w_2, \ldots, w_l, \ldots, w_L \qquad \max(N, M) \leq L < N + M - 1 \tag{3.3}$$

The minimum length of the warping path corresponds to $\max(N, M)$, corresponding to the diagonal of $\mathbf{d}$ when $N = M$. The maximum length is $L = N + M - 1$ when the warping path follows the two edges of the distance matrix. The time and space complexity of DTW is $\mathcal{O}(nm)$.

The warping path $\mathbf{W}$ must minimize the overall cost function

$$\text{COST}(\mathbf{W}) = \min\left(\sum_{l=1}^{L} \text{cost}[w_l]\right) \tag{3.4}$$

with the following four conditions [79, 80, 91]:

1. (monotonicity) Warping function should be monotonic, meaning that the warping function cannot go to "south" or "west":
   $n_l \geq n_{l-1}$ and $m_l \geq m_{l-1}$

2. (boundary condition) End points of the two vectors/sequences that are compared should be matched at the warping path:
   $w_1 = (1, 1)$ and $w_L = (N, M)$

3. (continuity condition) Warping function should not bypass any points:
   $n_l - n_{l-1} \leq 1$ and $m_l - m_{l-1} \leq 1$

4. Maximum amount of warp is controlled by a global limit:
   $|n_l - m_l| < G$
   This global constraint G is named as "window width" and it is used to speed up DTW and prevent pathological warpings [80]. A good path is unlikely to wander very far from the diagonal.

For a given pair of sequences, many different warping paths between $(1, 1)$ and (N, M) exist but the aim is to find the least-cost one. Therefore, a cumulative distance or cost matrix $\mathbf{D}$ is constructed starting at $(n, m) = (1, 1)$. $D(n, m)$ represents the cost of the least-cost path that can be obtained until reaching point (n, m). As stated above, the warp path must either be incremented by one or stay the same along the n and m axes. Therefore, the distances of the optimal warp paths one data point smaller than lengths n and m are contained in the matrix elements $D(n-1, m-1)$, $D(n-1, m)$, and $D(n, m-1)$. Therefore, $D(n, m)$ is calculated by:

$$D(n, m) = d(n, m) + \min\left[D(n-1, m-1), D(n-1, m), D(n, m-1)\right] \tag{3.5}$$

This equation defines the cumulative distance $D(n, m)$ as the distance $d(n, m)$ found in the current cell and the minimum of the cumulative distances of the three adjacent cells. Since this recurrence equation determines the value of a cell by using the values in three adjacent cells, the order that the cell values are evaluated is important: The cost matrix is filled one column at a time from the bottom up, and from the left to the right. The final value $D(N, M)$ is used as a measure of distance when comparing two given feature vectors.

After the entire matrix is filled, the least-cost warping path between $D(1, 1)$ and $D(N, M)$ can be found. This can be calculated very efficiently by using dynamic programming starting in reverse order with the (N, M) element and going backwards until reaching $(1, 1)$. At each step, adjacent cells at the left, at the bottom, and at the lower-left diagonal of the present cell are checked. In Figure 3.5, the three possible directions for constructing each step of the path are illustrated. Whichever of these three cells has the smallest value is added to the warp path found so far, and the search continues from that cell. In finding the smallest value among $D(n-1, m-1), D(n-1, m)$, and $D(n, m-1)$, if any two or three of these elements including $D(n-1, m-1)$ are equal, $D(n-1, m-1)$ is selected as the minimum. In other words, the diagonal path segment is preferred whenever possible. If $D(n-1, m)$, and $D(n, m-1)$ are equal and smaller than $D(n-1, m-1)$, then either $D(n-1, m)$ or $D(n, m-1)$ is chosen randomly. The search stops when $D(1, 1)$ is reached. The rationale for using a dynamic programming approach in this problem is that instead of attempting to solve the problem all at once, solutions to sub-problems (portions of the two sequences) are found, and used to iteratively find solutions until the solution is found for the entire sequences.

An example warping path $\mathbf{W}$ is shown in Figure 3.6. Part of the DTW path in this figure is given by:

$$\mathbf{W} = (1, 1), (2, 2), (3, 2), (4, 2), (5, 3), (6, 4), (6, 5), (7, 5), \ldots, (N, M) \qquad (3.6)$$

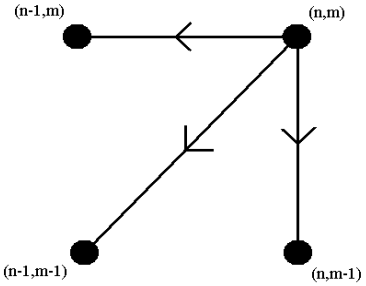


Figure 3.5: Three possible directions for constructing each step of the path.

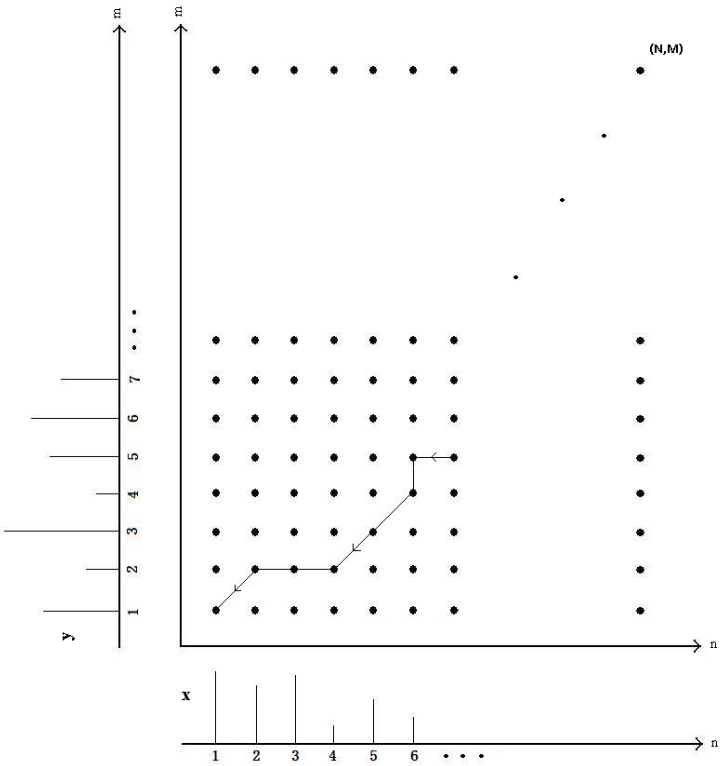
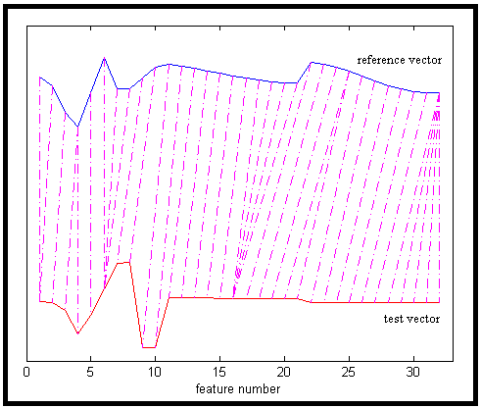
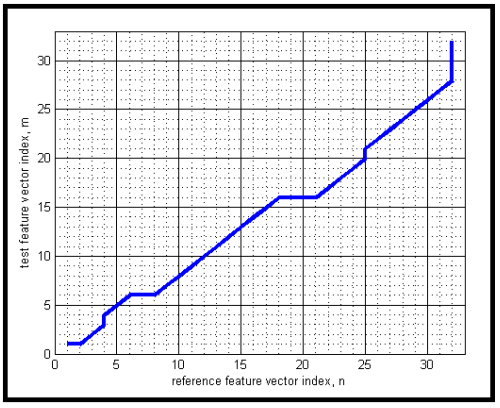


Figure 3.6: DTW mapping function.

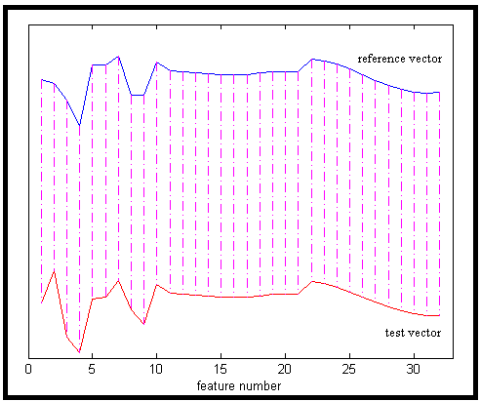
As an example, in Figure 3.7(a), the upper and lower curves represent a 32×1 reference vector and a a 32×1 test vector from two different classes. The alignment between the samples of these two vectors is illustrated with dot-dash lines. Since these two feature vectors are very different, there is a lot of warping when they are tried to be aligned, as illustrated in Figure 3.7(b). The reference and test vectors in part (c) of the figure both belong to the same class. Since these two vectors are very similar, warping is not observed between these two vectors and the corresponding minimum-distance warp path shown in Figure 3.7(d) is a straight line. In Figure 3.7(e) and (f), although both the reference and the test vector belong to the same class, there appears to be some warping. Sometimes, warping between reference and test vectors from the same class can be larger, so that errors at the classification stage become inevitable.



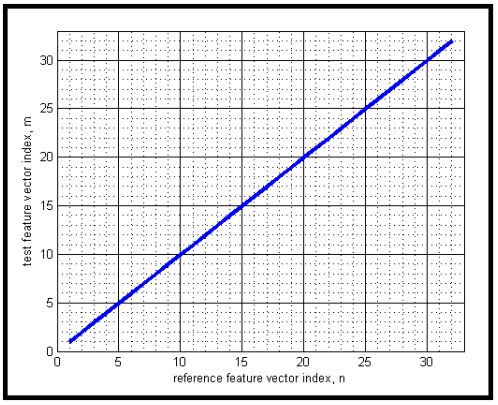
(a)



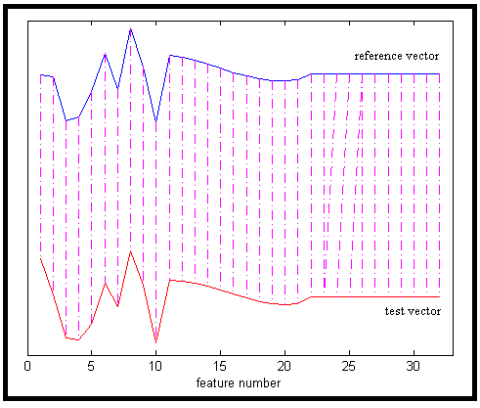
(b)



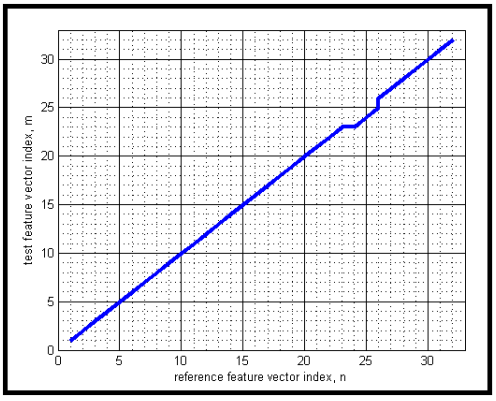
(c)



(d)



(e)



(f)

Figure 3.7: In (a), (c) and (e), upper curves show reference vectors and lower curves represent test vectors of size 32×1. Parts (b), (d) and (f) show least-cost warp paths between these two feature vectors, respectively. In (a), reference and test vectors are from different classes. In (c) and (e), both the reference and the test vectors are from the same class.

3.5 Support Vector Machines (SVMs)

SVM classifier is a machine learning technique proposed early in the eighties [92, 93]. It has been mostly used in applications such as object, voice, and handwritten character recognition, and text classification.

If the feature vectors in the original feature space are not linearly separable, SVMs preprocess and represent them in a space of higher dimension where they become linearly separable. The dimension of the transformed space may sometimes be much higher than the original feature space. With a suitable nonlinear mapping $\phi(.)$ 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 depends on the prior information available to the designer. If such 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.

Consider SVMs in a binary classification setting. We are given the training feature vectors $\mathbf{x}_i$ that are vectors in some space $\mathcal{X} \subseteq \Re^N$ and their labels $\ell_i \in \{-1, 1\}$ where $i = 1, \ldots, I$. Here, ℓ_i parameter is used to label the class of the feature vectors. If the feature vector is a class 1 vector, then $\ell_i = +1$, if it is a class 2 vector $\ell_i = -1$. The goal in training a SVM is to find the separating hyperplane with the largest margin so that the generalization of the classifier is better. All vectors lying on one side of the hyperplane are labeled as +1, and all vectors lying on the other side are labeled as –1. The support vectors are the (transformed) training patterns that lie closest to the hyperplane and are at equal distance from it. They correspond to the training samples that define the

optimal separating hyperplane and are the most difficult patterns to classify, yet the most informative for the classification task.

More generally, SVMs allow one to project the original training data in space $\mathcal{X}$ to a higher-dimensional feature space $\mathcal{F}$ via a Mercer kernel operator K [94]. We consider a set of classifiers of the form $f(\mathbf{x}) = \sum_{i=1}^{I} \beta_i \; K(\mathbf{x}, \mathbf{x}_i)$. When $f(\mathbf{x}) \geq 0$, we label $\mathbf{x}$ as +1, otherwise as –1. When K satisfies Mercer's condition, $K(\mathbf{u}, \mathbf{v}) = \phi(\mathbf{u}) \cdot \phi(\mathbf{v})$ where $\phi(.) : \mathcal{X} \rightarrow \mathcal{F}$ is a nonlinear mapping and "$\cdot$" denotes the inner or dot product. We can then rewrite $f(\mathbf{x})$ in the transformed space as $f(\mathbf{x}) = \mathbf{a} \cdot \phi(\mathbf{x})$. The linear discriminant function $f(\mathbf{x})$ is based on the hyperplane $\mathbf{a} \cdot \phi(\mathbf{x}) = 0$ where $\mathbf{a} = \sum_{i=1}^{I} \beta_i \; \phi(\mathbf{x}_i)$ is a weight vector. Thus, by using K, the training data is projected into a new feature space $\mathcal{F}$ which is often higher dimensional. The SVM then computes the β_i's that correspond to the maximal margin hyperplane in $\mathcal{F}$. By choosing different kernel functions, we can project the training data from $\mathcal{X}$ into spaces $\mathcal{F}$ for which hyperplanes in $\mathcal{F}$ correspond to more complex decision boundaries in the original space $\mathcal{X}$. Hence, by nonlinear mapping of the original training patterns into other spaces, decision functions can be found using a linear algorithm in the transformed space by only computing the kernel $K(\mathbf{x}, \mathbf{x}_i)$.

To illustrate the problem in 2-D, consider the training set feature vectors in Figure 3.8. In this example, there are two classes, squares ($\ell_i = +1$) symbolize the first class (class 1) and circles ($\ell_i = -1$) symbolize the second class (class 2). These two type of training vectors can be separated with infinitely many different hyperplanes, three of which are shown in Figure 3.8(a). For each of these hyperplanes, success rates may be different when test vectors are presented to the system. To have the smallest classification error at the test stage, hyperplane should be placed between support vectors of two classes with the maximum and equal margin for both of the classes [95]. For a SVM, the optimal hyperplane classifier is unique [76]. The equation of a hyperplane that may be used to classify

these two classes is given by:

$$\mathbf{a} \cdot \phi(\mathbf{x}) = 0 \tag{3.7}$$

and is represented by the solid line in Figure 3.8(b). Here, both the weight vector $\mathbf{a}$ and the transformed feature vector $\phi(\mathbf{x}_i)$ have been augmented by one dimension to include a bias weight so that the hyperplanes need not pass through the origin.

For this hyperplane to have maximum margins, dotted and dashed margin lines in Figure 3.8(b) are given by the following two equations, respectively:

$$\begin{aligned} \mathbf{a} \cdot \phi(\mathbf{x}) &= 1 \\ \mathbf{a} \cdot \phi(\mathbf{x}) &= -1 \end{aligned} \tag{3.8}$$

In the same figure, vectors that are marked with extra circles correspond to the support vectors.

Since there should not be training set vectors dropping between these margin lines, the following equations should be satisfied:

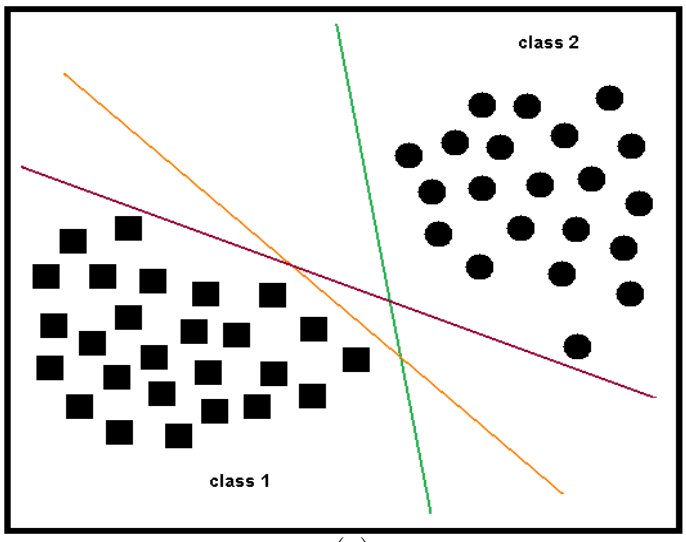
$$\begin{aligned} \mathbf{a} \cdot \phi(\mathbf{x}_i) &\geq 1, \quad \forall \mathbf{x}_i \in \text{ class 1} \\ \mathbf{a} \cdot \phi(\mathbf{x}_i) &\leq -1, \quad \forall \mathbf{x}_i \in \text{ class 2} \end{aligned} \tag{3.9}$$

More compactly, a separating hyperplane ensures

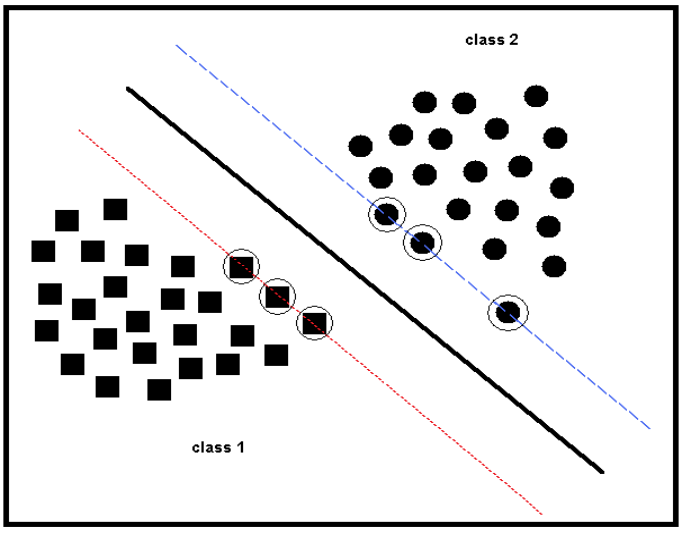
$$\ell_i \; f(\mathbf{x}_i) = \ell_i \; \mathbf{a} \cdot \phi(\mathbf{x}_i) \geq 1 \quad \text{for} \quad i = 1, \ldots, I \tag{3.10}$$

Assuming $\mathbf{a} = [\mathbf{n}, a_0]$ where $\mathbf{n}$ is the normal vector of the hyperplane, it can be shown that the distance between the two margin lines is $2/\left\|\mathbf{n}\right\|$. Therefore, to maximize the separation between these margin lines, $\|\mathbf{n}\|$ should be minimized. Since a_0 is a constant, this is equivalent to minimizing $\|\mathbf{a}\|$.

To have optimal margin hyperplanes for classification of feature vectors, the optimal hyperplane can be found by minimizing the magnitude of the weight



(a)



(b)

Figure 3.8: (a) Three different hyperplanes separating two classes. (b) SVM hyperplane, its margins, and the support vectors.

vector $\| \mathbf{a} \|^2$ subject to the constraint given by Equation (3.10) [96]. Using the method of Lagrange multipliers, we construct the functional

$$\mathcal{L}(\mathbf{a}, \lambda) = \frac{1}{2} \| \mathbf{a} \|^2 - \sum_{i=1}^{I} \lambda_i \left[\ell_i \, \mathbf{a} \cdot \phi(\mathbf{x}_i) - 1 \right] \tag{3.11}$$

where the second term in the above equation expresses the goal of classifying the points correctly. To find the optimal hyperplane, we minimize $\mathcal{L}(.)$ with respect to the weight vector $\mathbf{a}$, while maximizing with respect to the undetermined Lagrange multipliers $\lambda_i \geq 0$. This can be done by solving the constrained optimization problem by quadratic programming [97] or by other alternative techniques. The solution of the weight vector is $\mathbf{a}^* = \sum_{i=1}^{I} \ell_i \, \lambda_i \, \phi(\mathbf{x}_i)$ corresponding to $\beta_i = \ell_i \lambda_i$. Then, the decision function is given by:

$$f^*(\mathbf{x}) = \sum_{i=1}^{I} \lambda_i \, \ell_i \, \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}) \tag{3.12}$$

In this study, the method summarized above is applied to differentiate feature vectors that belong to more than two classes. Following the one-versus-the-rest method, c different binary classifiers are trained, where each classifier recognizes one of c motion types.

In this study, performance of linear classifiers was not satisfactory for classifying human motions. Therefore, a nonlinear classifier is used with radial basis function (RBF) kernel according to the following model with $\gamma = 4$ for the first part and $\gamma = 0.2$ for the second part of this study:

$$K(\mathbf{x}, \mathbf{x}_i) = e^{-\gamma |\mathbf{x} - \mathbf{x}_i|^2} \tag{3.13}$$

A library for SVMs (LIBSVM toolbox) is used in the MATLAB environment [98].

Chapter 4

EXPERIMENTAL RESULTS

4.1 Leg-Motion Classification Results

In this study, the classification methods described in the previous section are used to classify eight different leg motions.

A total of 448 ($= 7 \times 8 \times 8$) feature vectors are available for this part of the study. In the training and testing phases of the classification methods, we used different approaches which are: repeated random sub-sampling (RRSS), P-fold, and leave-one-out (LOO) cross-validation techniques. In RRSS, we divided the 56 feature vectors from each motion type randomly into two sets so that each of the two sets contains 28 feature vectors. In total, 224 ($= 28 \times 8$) vectors are used for training and the same number of vectors is used for testing. This is repeated 100 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 total number of 448 feature vectors are divided into $P = 8$ partitions where each partition contains 7 randomly selected feature vectors from each class, therefore a total of 56 vectors. Of the P partitions, a single partition 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. This process is repeated 100 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.

Finally, we also used LOO cross validation, where a single feature vector out of 448 is used in turn for validation, and the remaining 447 feature vectors are used for training. This is repeated such that each feature vector is used once as the validation data. This is the same as a P-fold cross-validation with P being equal to the number of feature vectors in the original sample ($P = 448$). Since the training process is repeated a large number of times, LOO cross-validation technique is often computationally expensive.

Correct differentiation rates obtained with different classification techniques are given in Tables 4.1–4.3 for the five different feature sets we have considered and the three different validation techniques. For the RBA, the features used do not correspond to one of the sets presented in Tables 4.1–4.3. Therefore, RBA results are not listed in these tables. Correct differentiation rates of 95.2%, 95.1%, and 95.1% are achieved with RBA for RRSS, P-fold, and LOO cross-validation techniques, respectively.

Among the five different feature sets that we have considered, the first two and the last one result in higher classification rates. Since the last feature set

method:	correct differentiation rate (%)				
	by inspection (14 features)	PCA to 14 features (6 features)	covariance matrix (14 features)	PCA to 101 features (8 features)	SFFS (6 features)
LSM	97.0	96.9	91.8	88.5	94.6
k-NN ($k = 1$)	96.9	96.9	95.3	94.9	96.4
DTW-1	92.1	92.2	87.9	82.6	95.4
DTW-2	96.9	96.3	95.1	93.6	95.7
SVM	99.2	99.1	94.6	94.6	97.2

Table 4.1: Correct differentiation rates for all classification methods for different feature reduction methods and RRSS cross validation.

method:	correct differentiation rate (%)				
	by inspection (14 features)	PCA to 14 features (6 features)	covariance matrix (14 features)	PCA to 101 features (8 features)	SFFS (6 features)
LSM	97.3	97.5	92.1	89.5	94.6
k-NN ($k = 1$)	97.1	98.1	94.8	95.4	97.4
DTW-1	91.8	92.8	87.7	83.8	95.7
DTW-2	98.0	96.9	96.1	95.2	97.0
SVM	99.7	99.4	95.3	96.7	97.9

Table 4.2: Correct differentiation rates for all classification methods for different feature reduction methods and P-fold cross validation.

method:	correct differentiation rate (%)				
	by inspection (14 features)	PCA to 14 features (6 features)	covariance matrix (14 features)	PCA to 101 features (8 features)	SFFS (6 features)
LSM	97.1	97.3	92.0	90.4	94.2
k-NN ($k = 1$)	97.1	98.2	94.6	95.1	97.6
DTW-1	91.7	93.8	88.0	83.7	96.0
DTW-2	98.2	97.8	95.2	95.1	97.3
SVM	98.9	98.4	96.4	98.4	98.2

Table 4.3: Correct differentiation rates for all classification methods for different feature reduction methods and LOO cross validation.

(obtained by SFFS) can be obtained more systematically, we used this feature set in reporting the confusion matrices of the different techniques.

From the tables, it can be observed that there is not a significant difference between the results of different cross-validation techniques. Among the classification techniques we have considered and implemented, SVM in general gives the highest classification rate, followed by k-NN (for $k = 1$) except for a few cases. Since the LOO cross-validation gives slightly larger correct differentiation rates, this cross-validation technique is used in obtaining the confusion matrices of the classification techniques presented in Tables 4.4–4.8.

		c l a s s i f i e d							
		M1	M2	M3	M4	M5	M6	M7	M8
a	M1	56	0	0	0	0	0	0	0
c	M2	0	56	0	0	0	0	0	0
t	M3	0	0	49	0	0	0	7	0
u	M4	0	0	0	46	10	0	0	0
a	M5	0	0	0	4	52	0	0	0
l	M6	0	0	0	0	0	56	0	0
	M7	0	0	1	0	0	0	55	0
	M8	0	0	0	0	0	0	0	56

Table 4.4: Confusion matrix for RBA (LOO cross-validation, 95.1%).

		c l a s s i f i e d							
		M1	M2	M3	M4	M5	M6	M7	M8
a	M1	56	0	0	0	0	0	0	0
c	M2	0	46	0	0	0	0	0	10
t	M3	0	0	54	2	0	0	0	0
u	M4	0	0	0	50	6	0	0	0
a	M5	0	0	0	3	53	0	0	0
l	M6	0	0	0	0	0	56	0	0
	M7	0	0	0	0	0	0	56	0
	M8	0	5	0	0	0	0	0	51

Table 4.5: Confusion matrix for LSM (LOO cross-validation, 94.2%).

		c l a s s i f i e d							
		M1	M2	M3	M4	M5	M6	M7	M8
a	M1	56	0	0	0	0	0	0	0
c	M2	0	52	0	0	0	0	0	4
t	M3	0	0	56	0	0	0	0	0
u	M4	0	0	0	52	4	0	0	0
a	M5	0	0	0	2	54	0	0	0
l	M6	0	0	0	0	0	56	0	0
	M7	0	0	0	0	0	0	56	0
	M8	0	1	0	0	0	0	0	55

Table 4.6: Confusion matrix for the k-NN algorithm for $k = 1$ (LOO cross-validation, 97.6%).

		c l a s s i f i e d							
		M1	M2	M3	M4	M5	M6	M7	M8
a	M1	56	0	0	0	0	0	0	0
c	M2	0	49	0	0	0	2	0	5
t	M3	0	0	56	0	0	0	0	0
u	M4	0	0	0	52	4	0	0	0
a	M5	0	0	1	3	52	0	0	0
l	M6	0	0	0	0	0	56	0	0
	M7	0	0	2	0	0	0	54	0
	M8	0	1	0	0	0	0	0	55

Table 4.7: Confusion matrix for DTW-1 (LOO cross-validation, 96.0%).

		c l a s s i f i e d							
		M1	M2	M3	M4	M5	M6	M7	M8
a	M1	56	0	0	0	0	0	0	0
c	M2	0	54	0	0	0	0	0	2
t	M3	0	0	56	0	0	0	0	0
u	M4	0	0	0	53	3	0	0	0
a	M5	0	0	0	5	51	0	0	0
l	M6	0	0	0	0	0	56	0	0
	M7	0	0	1	0	0	0	55	0
	M8	0	1	0	0	0	0	0	55

Table 4.8: Confusion matrix for DTW-2 (LOO cross-validation, 97.3%).

In the LSM approach, test vectors are compared with the average of the reference vectors that are calculated for each of the eight classes. Confusion matrix for this method is provided in Table 4.5. Overall successful differentiation rate of LSM is 94.2%.

Performance of the k-NN algorithm changes for different values of k. Correct differentiation rates for different k values varying between 1 to 28 and 1 to 55 have been considered in Figures 4.1–4.2 for RRSS and LOO cases respectively. As the value of k increases, successful classification rate decreases. Values of k between 1 and 6 seem to be more suitable since they provide large classification rates. Confusion matrix of the k-NN algorithm for $k = 1$ is provided in Table 4.6, where a successful differentiation rate of 97.6% is achieved.

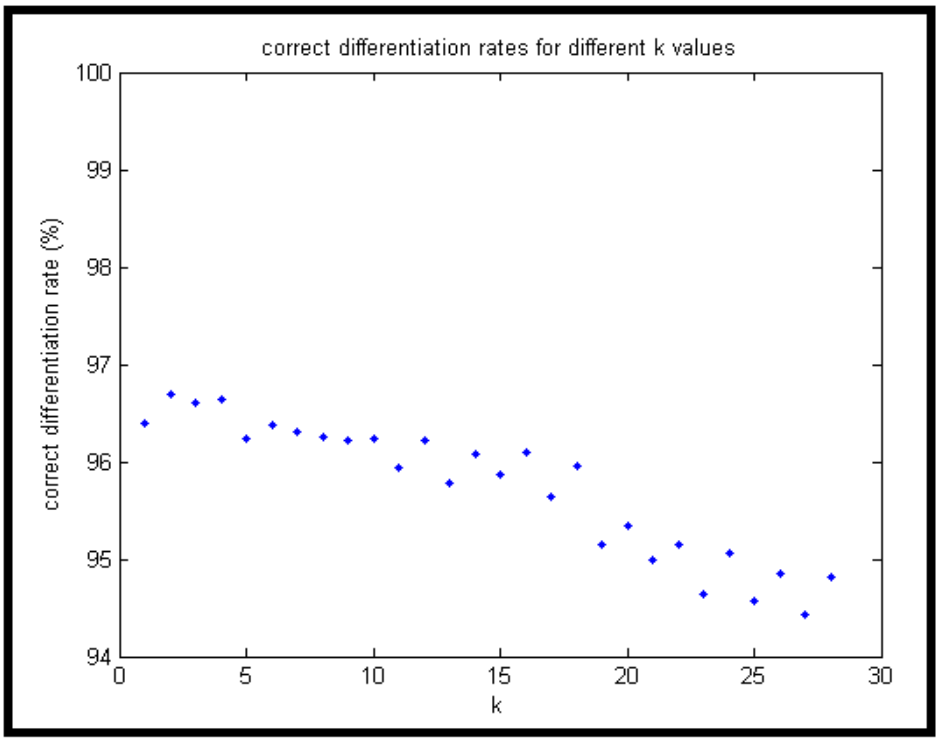


Figure 4.1: Correct differentiation rates of k-NN algorithm for $k = 1, \ldots, 28$ (RRSS).

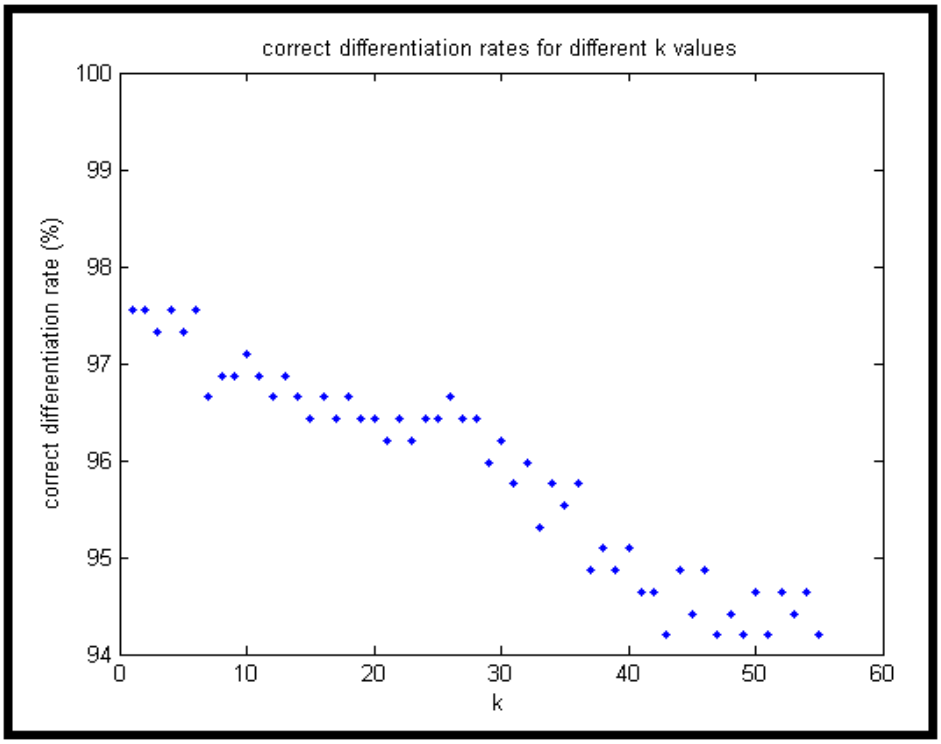


Figure 4.2: Correct differentiation rates of k-NN algorithm for $k = 1, \ldots, 55$ (LOO).

We implemented the DTW algorithm in two different ways: In the first approach, the average of the reference feature vectors for each motion is used for comparison. Confusion matrix for the DTW method by using this first approach (DTW-1) is presented in Table 4.7 where a correct differentiation rate of 96.0% is achieved.

As a second approach (DTW-2), DTW distances are calculated between the test vector and each of the $(56 \times 8) - 1 = 447$ reference vectors from different classes. The class of the nearest reference vector is assigned as the class of the test vector. Success rate of this second approach is 97.3%. Corresponding confusion matrix can be seen in Table 4.8.

In SVM, following the one-versus-the-rest method, each leg motion is assumed as the first class and the remaining seven leg motions are assumed as the second class. A different SVM model is created for the classification of each test vector. This process is repeated 448 times and the average success rate and the number of correctly classified motions for each SVM model is calculated. Number of correctly and incorrectly classified feature vectors is tabulated in Table 4.9. Overall success rate of the SVM method is calculated as 98.2%.

		classified	
		correct	incorrect
actual	M1	56	0
	M2	54	2
	M3	56	0
	M4	53	3
	M5	53	3
	M6	56	0
	M7	56	0
	M8	56	0

Table 4.9: Number of correctly and incorrectly classified feature vectors out of 56 for SVMs (LOO cross-validation, 98.2%).

4.2 Human Activity Classification Results

In this part of the study, the same classification and cross-validation techniques used above are employed to classify 19 different human activities. The eight features selected by PCA and listed in Section 2.2.2 are used as the only feature set. A total of 1140 ($= 60 \times 19$) feature vectors are available for this part of the study.

In RRSS, we divided the 60 feature vectors from each motion type randomly into two sets so that each of the two sets contains 30 feature vectors. In total, 30×19 vectors are used for training and the same number of vectors is used for testing. This is repeated 100 times and the resulting correct differentiation percentages are averaged. Given correct classification rates and confusion matrices are the average values that are obtained at the end of these 100 runs.

In P-fold cross-validation, the total number of 1140 feature vectors are divided into $P = 10$ partitions where each partition contains 6 randomly selected feature vectors from each class, therefore a total of 114 vectors. Of the P partitions, a single partition 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. This process is repeated 100 times and the average correct differentiation percentage is reported.

In LOO cross validation, a single feature vector out of 1140 is used in turn for validation, and the remaining 1139 feature vectors are used for training. This is repeated such that each feature vector is used once as the validation data. This is the same as a P-fold cross-validation with P being equal to 1140.

For the RBA, the features used in the rules do not correspond to the 8 features selected based on PCA. Therefore, RBA results are not listed in Table 4.10. Using RBA, correct differentiation rates of 97.1%, 96.9%, and 97.0% are achieved for RRSS, P-fold, and LOO cross-validation techniques, respectively.

method:	correct differentiation rate (%)		
	RRSS	P-fold	LOO
LSM	97.6	97.8	97.8
k-NN ($k = 1$)	98.9	98.9	99.0
DTW-1	97.4	97.5	97.5
DTW-2	98.7	98.7	98.7
SVM	98.6	98.9	98.9

Table 4.10: Correct differentiation rates for all classification methods and three cross-validation techniques.

Successful differentiation rates of the remaining classification methods are given in Table 4.10. All of the correct differentiation rates are above 97% and there is not a significant difference between the results of different cross-validation techniques. Among the classification techniques we have considered and implemented, the k-NN algorithm (for $k = 1$) consistently gives the highest classification rate. Correct differentiation rates for RRSS and LOO with the k values $k = 1, \ldots, 30$ and $k = 1, \ldots, 59$ are shown in Figure 4.3 and Figure 4.4 respectively. It can be observed that correct classification rate starts to drop after $k = 5$ for RRSS and $k = 19$ for LOO. Since the LOO cross-validation gives slightly larger correct differentiation rates, the confusion matrices of the classification techniques are tabulated for LOO cross validation in Tables 4.11–4.15. Number of correctly and incorrectly classified feature vectors is tabulated in Table 4.16.

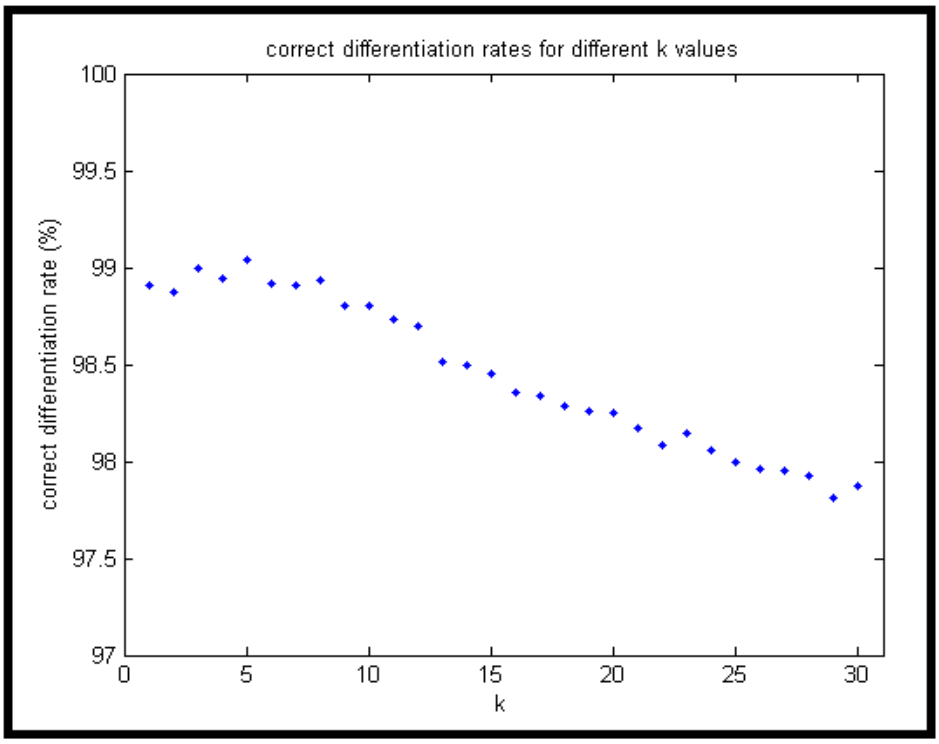


Figure 4.3: Correct differentiation rates of k-NN algorithm for $k = 1, \ldots, 30$ (RRSS).

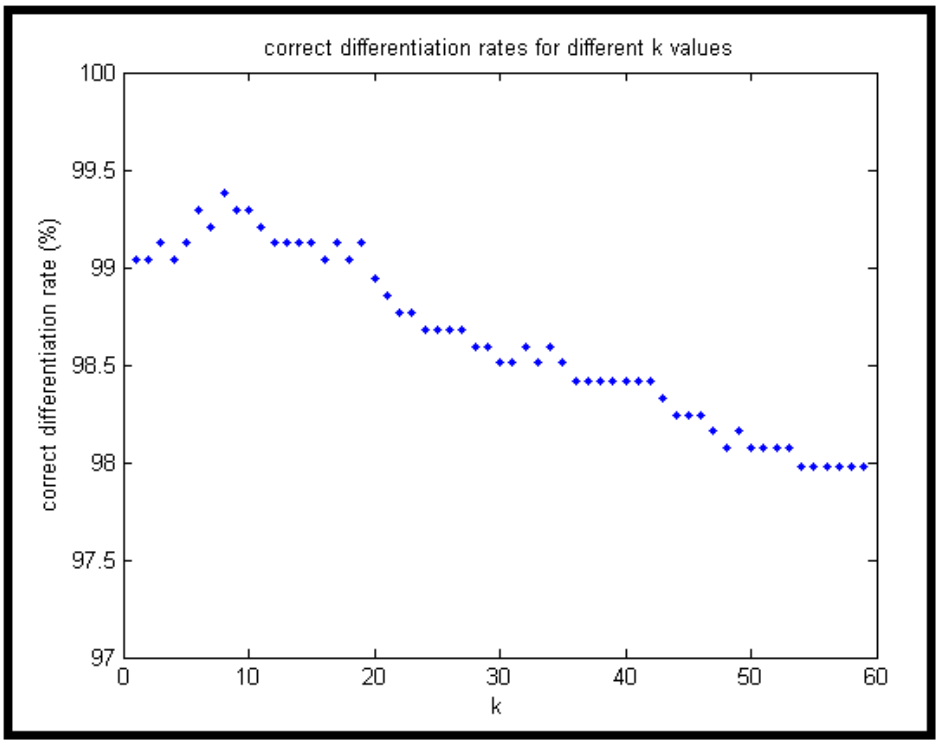


Figure 4.4: Correct differentiation rates of k-NN algorithm for $k = 1, \ldots, 59$ (LOO).

		classified																		
		M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	M15	M16	M17	M18	M19
actual	M1	59	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	M2	0	57	2	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	M3	0	0	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	M4	0	0	0	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	M5	0	0	0	0	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	M6	0	0	0	0	0	60	0	0	0	0	0	0	0	0	0	0	0	0	0
	M7	0	0	0	0	0	0	53	7	0	0	0	0	0	0	0	0	0	0	0
	M8	7	0	0	0	0	4	7	38	0	0	0	0	0	0	0	4	0	0	0
	M9	0	0	0	0	0	0	0	0	60	0	0	0	0	0	0	0	0	0	0
	M10	0	0	0	0	0	0	0	0	0	60	0	0	0	0	0	0	0	0	0
	M11	0	0	0	0	0	0	0	0	0	0	60	0	0	0	0	0	0	0	0
	M12	0	0	0	0	0	0	0	0	0	0	0	60	0	0	0	0	0	0	0
	M13	0	0	1	0	0	0	0	0	0	0	0	0	59	0	0	0	0	0	0
	M14	0	0	0	0	0	0	0	0	0	0	0	0	0	60	0	0	0	0	0
	M15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60	0	0	0	0
	M16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60	0	0	0
	M17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60	0	0
	M18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60	0
	M19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60

Table 4.11: Confusion matrix for RBA (LOO cross-validation, 97.0%).

		classified																		
		M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	M15	M16	M17	M18	M19
actual	M1	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	M2	0	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	M3	0	0	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	M4	0	0	0	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	M5	0	0	0	0	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	M6	0	0	0	0	0	60	0	0	0	0	0	0	0	0	0	0	0	0	0
	M7	0	0	0	0	0	0	56	1	0	0	0	0	0	0	3	0	0	0	0
	M8	5	0	0	0	0	0	9	46	0	0	0	0	0	0	0	0	0	0	0
	M9	0	0	0	0	0	0	0	0	60	0	0	0	0	0	0	0	0	0	0
	M10	0	0	0	0	0	0	0	0	0	60	0	0	0	0	0	0	0	0	0
	M11	0	0	0	0	0	0	0	0	0	0	60	0	0	0	0	0	0	0	0
	M12	0	0	0	0	0	0	0	0	0	0	0	60	0	0	0	0	0	0	0
	M13	0	0	0	0	0	0	0	0	0	0	0	0	60	0	0	0	0	0	0
	M14	0	0	0	0	0	0	0	0	0	0	0	0	0	60	0	0	0	0	0
	M15	0	0	0	0	0	0	6	0	0	0	0	0	0	0	54	0	0	0	0
	M16	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	59	0	0	0
	M17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60	0	0
	M18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60	0
	M19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60

Table 4.12: Confusion matrix for LSM (LOO cross-validation, 97.8%).

		classified																		
		M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	M15	M16	M17	M18	M19
actual	M1	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	M2	0	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	M3	0	0	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	M4	0	0	0	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	M5	0	0	0	0	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	M6	0	0	0	0	0	60	0	0	0	0	0	0	0	0	0	0	0	0	0
	M7	0	0	0	0	0	0	56	4	0	0	0	0	0	0	0	0	0	0	0
	M8	0	0	0	0	0	0	7	53	0	0	0	0	0	0	0	0	0	0	0
	M9	0	0	0	0	0	0	0	0	60	0	0	0	0	0	0	0	0	0	0
	M10	0	0	0	0	0	0	0	0	0	60	0	0	0	0	0	0	0	0	0
	M11	0	0	0	0	0	0	0	0	0	0	60	0	0	0	0	0	0	0	0
	M12	0	0	0	0	0	0	0	0	0	0	0	60	0	0	0	0	0	0	0
	M13	0	0	0	0	0	0	0	0	0	0	0	0	60	0	0	0	0	0	0
	M14	0	0	0	0	0	0	0	0	0	0	0	0	0	60	0	0	0	0	0
	M15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60	0	0	0	0
	M16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60	0	0	0
	M17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60	0	0
	M18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60	0
	M19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60

Table 4.13: Confusion matrix for the k-NN algorithm for $k = 1$ (LOO cross-validation, 99.0%).

		classified																		
		M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	M15	M16	M17	M18	M19
actual	M1	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	M2	0	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	M3	0	0	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	M4	0	0	0	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	M5	0	0	0	0	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	M6	0	0	1	0	0	59	0	0	0	0	0	0	0	0	0	0	0	0	0
	M7	0	0	0	0	0	0	55	1	0	0	0	0	0	0	4	0	0	0	0
	M8	6	0	0	0	0	0	8	46	0	0	0	0	0	0	0	0	0	0	0
	M9	0	0	1	0	0	0	0	0	58	0	0	0	0	0	0	1	0	0	0
	M10	0	0	0	0	0	0	0	0	0	59	0	0	0	1	0	0	0	0	0
	M11	0	0	0	0	0	0	0	0	0	0	60	0	0	0	0	0	0	0	0
	M12	0	0	0	0	0	0	0	0	0	0	0	60	0	0	0	0	0	0	0
	M13	0	0	0	0	0	0	0	0	0	0	0	0	60	0	0	0	0	0	0
	M14	0	0	0	0	0	0	0	0	0	0	0	0	0	60	0	0	0	0	0
	M15	0	0	0	0	0	0	5	0	0	0	0	0	0	0	55	0	0	0	0
	M16	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	59	0	0	0
	M17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60	0	0
	M18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60	0
	M19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60

Table 4.14: Confusion matrix for DTW-1 (LOO cross-validation, 97.5%).

		classified																		
		M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	M15	M16	M17	M18	M19
actual	M1	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	M2	0	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	M3	0	0	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	M4	0	0	0	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	M5	0	0	0	0	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	M6	0	0	0	0	0	60	0	0	0	0	0	0	0	0	0	0	0	0	0
	M7	0	0	0	0	0	0	55	5	0	0	0	0	0	0	0	0	0	0	0
	M8	0	0	0	0	0	0	10	50	0	0	0	0	0	0	0	0	0	0	0
	M9	0	0	0	0	0	0	0	0	60	0	0	0	0	0	0	0	0	0	0
	M10	0	0	0	0	0	0	0	0	0	60	0	0	0	0	0	0	0	0	0
	M11	0	0	0	0	0	0	0	0	0	0	60	0	0	0	0	0	0	0	0
	M12	0	0	0	0	0	0	0	0	0	0	0	60	0	0	0	0	0	0	0
	M13	0	0	0	0	0	0	0	0	0	0	0	0	60	0	0	0	0	0	0
	M14	0	0	0	0	0	0	0	0	0	0	0	0	0	60	0	0	0	0	0
	M15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60	0	0	0	0
	M16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60	0	0	0
	M17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60	0	0
	M18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60	0
	M19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60

Table 4.15: Confusion matrix for DTW-2 (LOO cross-validation, 98.7%).

		classified	
		correct	incorrect
actual	M1	60	0
	M2	60	0
	M3	60	0
	M4	60	0
	M5	60	0
	M6	60	0
	M7	57	3
	M8	54	6
	M9	60	0
	M10	60	0
	M11	60	0
	M12	60	0
	M13	60	0
	M14	60	0
	M15	57	3
	M16	59	1
	M17	60	0
	M18	60	0
	M19	60	0

Table 4.16: Number of correctly and incorrectly classified feature vectors out of 60 for SVMs (LOO cross-validation, 98.9%).

4.3 Processing Times of the Classification Methods

The classification methods given above are also compared based on their pre-processing and classification times. Pre-processing and classification times are calculated on an Intel Centrino Duo CPU T2400 @1.83 GHz, 0.99 GB RAM laptop computer running the Microsoft Windows XP Professional operating system. Pre-processing times of the different techniques are tabulated in Tables 4.17 and 4.18. Processing times required for the classification of one feature vector are given in Tables 4.19 and 4.20. The classification time for RBA is the shortest, followed by SVM or LSM, k-NN ($k = 1$) or DTW-1, and DTW-2 methods. SVM requires the longest training time, whereas DTW-2 takes the longest amount of classification time. Among the different cross-validation techniques, RRSS requires the shortest amount of classification time, whereas LOO takes the longest.

	pre-processing/training time (msec)		
method:	RRSS	P-fold	LOO
RBA	–	–	–
LSM	0.098	0.554	105.141
k-NN ($k = 1$)	–	–	–
DTW-1	0.098	0.554	105.141
DTW-2	–	–	–
SVM	72.933	1880.233	5843.133

Table 4.17: Pre-processing times of the classification methods for leg motion classification part.

method:	pre-processing/training time (msec)		
	RRSS	P-fold	LOO
RBA	–	–	–
LSM	0.829	1.300	835.703
k-NN ($k = 1$)	–	–	–
DTW-1	0.829	1.300	835.703
DTW-2	–	–	–
SVM	500.933	12299.400	42679.890

Table 4.18: Pre-processing times of the classification methods for human activity classification part.

method:	classification time (msec)		
	RRSS	P-fold	LOO
RBA	0.003	0.003	0.003
LSM	0.070	0.074	0.063
k-NN ($k = 1$)	0.095	0.452	24.033
DTW-1	1.775	1.937	2.000
DTW-2	49.640	94.014	107.400
SVM	0.009	0.016	0.132

Table 4.19: Processing times required for the classification of one feature vector for leg motion classification part.

method:	classification time (msec)		
	RRSS	P-fold	LOO
RBA	0.007	0.007	0.007
LSM	0.106	0.156	0.123
k-NN ($k = 1$)	0.253	0.676	58.699
DTW-1	7.675	7.736	8.686
DTW-2	232.057	413.027	502.931
SVM	0.007	0.009	0.150

Table 4.20: Processing times required for the classification of one feature vector for human activity classification part.

Chapter 5

CONCLUSIONS, POTENTIAL APPLICATION AREAS and FUTURE WORK

5.1 CONCLUSIONS

A performance comparison of RBA, LSM, k-NN, DTW-1, DTW-2 and SVM algorithms is provided in terms of their correct differentiation rates, confusion matrices, pre-processing and training times and classification times. Among the classification techniques we have considered and implemented, SVM, in general, gives the highest correct differentiation rate, followed by k-NN. The classification time for RBA is the shortest, followed by SVM or LSM, k-NN or DTW-1, and DTW-2 methods. SVM requires the longest training time, whereas DTW-2 takes the longest amount of classification time. Although there is not a significant difference between the correct differentiation rates obtained by different cross-validation techniques, RRSS uses the shortest amount of classification time, whereas LOO requires the longest.

In the first part of this study, when the confusion matrices are examined it is observed that motion 4 and motion 5 also motion 2 and motion 8 may be confused with each other. In motion 2 and motion 8 only the lower part of the leg is moving backward and forward respectively. In motion 4 and motion 5 both parts of the leg moving without bending the knee forward and backward respectively. It can be said that by using the features we selected and by using only two single-axis gyroscopes, similar motions performed in different directions may be confused with each other.

In the second part of the study motion 7 and motion 8 are confused with each other. Both of these motions are performed in the elevator. In the 5 min data storing period some parts of these motions are very similar so the confusion at the classification steps becomes inevitable.

RBA, LSM and SVM can be used in real-time human activity classification systems since they have high correct differentiation rates and short classification times. However, to use RBA in real-time systems, decision rules should be determined beforehand. If SVM is thought to be used at real-time human activity recognition applications, training of the system should be done before the real-time classification because the training of SVM is a time consuming process. In non-real-time human activity classification applications when pre-processing time, classification time and correct differentiation rates are considered in total k-NN algorithm can be a suitable choice as a classification method.

5.2 POTENTIAL APPLICATION AREAS

Human motion analysis and differentiation has applications in many diverse areas. A significant application area is the remote monitoring of elderly people who live alone and who may need additional support. Emergency situations arising

from accidental falls and changes in vital signs needing attention must be detected in 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 closely related 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 the heart rate, blood pressure, and temperature monitors together with motion and position information (such as whether the subject is exercising, sleeping, has recently fallen) can allow a much better judgment of the situation and help determine whether attention is required.

Another potential area of high impact is ergonomics and the proper use of tools, devices, and instruments, which is important both for efficiency and for human health. Productivity of workers can be improved by monitoring whether they perform their tasks in the most efficient, optimal, safe and non-exhausting manner. This would also help in the prevention of repetitive motion injury (e.g. developing carpal tunnel syndrome) by providing warning signals against improper motions.

Likewise, in the area of physical education, training and sports, ballet and dance, such monitoring can be used to help trainers and individuals to obtain feedback regarding the correctness of their motions in terms of effectiveness and safety, increasing the benefits of physical exercise, improving athletic performance, and most importantly, promoting health and preventing injuries. Detection of sports rule violations can also be handled.

Functional recording of sports performances as well as both traditional and modern ballet and dance, is another application that is significant from the cultural heritage viewpoint. Such recording provides complementary information

to ordinary video recording. Whereas ordinary video recording provides a projection of the 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 may be considered to be more intrinsic than the camera image.

It is not hard to imagine applications in learning to play a musical instrument or even conducting an orchestra! Students and professionals alike can benefit from self-monitoring and use this as an aid to overcome bad habits and to improve and perfect their technique. Motion injuries are also encountered in musicians so it may also have a benefit in their prevention.

Generalizing from these example application areas, 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 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.

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 actions of the subject.

If we extend the application areas beyond motion recognition and classification of human beings, there are also plenty of applications in the monitoring and classification of motions of animals. For example, changes in behavior of

groups of animals due to diseases such as avian flu or the mad cow disease can be detected through the techniques developed here.

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

As the sensors continue to get even 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 their applications, since as the discomfort or burden of wearing these becomes negligible, it will be possible to consider applications in many other areas of daily life that are currently out of question since the present sensors are not lightweight enough.

5.3 FUTURE WORK

Normalization between the way different individuals perform the same activities is an aspect of activity recognition and classification that has not been studied at all. Each person may do a particular activity differently due to their body size, personal differences in style, and timing. For example, in performing a rehabilitation exercise, a patient may need to reach something, pick it up and bring it down. Typically, the timing of each individual will be different for each segment of the motion. Although some approaches may be more prone to personal differences, new techniques need to be developed that involve time-warping and projections of the signals and comparing their differentials. In this study classified motions are performed only by a single male subject. In our future studies we are planning to increase the number of subjects. To the best of our knowledge, optimal positioning, number, and type of sensors issues related to motion comparison have not been studied at all.

An important problem on which there is very little published work is the detection and classification of falls using inertial sensors [99]. Falls are hazardous and it is important to classify the type of the fall so that parts of the body that may have been injured can be identified. One of the reasons that fall detection has not been studied much is the difficulty of designing and performing fair and realistic experiments. The state-of-the-art is such that there still does not exist standard and systematic techniques for activity recognition and classification, in particular, for detecting falls. Fall detection and classification is as an area of current and future research [24] where it is necessary to agree upon a definition of falls and fall detection [41, 100].

Fusion of information from inertial sensors and cameras can be investigated 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.

APPENDIX A

Principal Component Analysis (Karhunen-Loéve Transformation)

Principal Component Analysis (PCA) is a technique used in pattern recognition to reduce the size of the feature vectors by eliminating the redundant features. Components of the feature vector are extracted from the acquired signals or real world data and are transformed to a new space where they become uncorrelated [101]. Features with large variances are more discriminating so they are used to construct the transformation matrix, whereas features with small variances are considered as noise [91]. The steps of PCA are as follows [102]:

- mean of each feature is calculated and subtracted from the corresponding vector element,
- covariance matrix of the training vectors is calculated,
- eigenvalues and eigenvectors of the covariance matrix are calculated,

- transformation matrix is obtained by arranging the eigenvectors in descending order of their eigenvalues, and
- features are transformed and decorrelated.

The diagonal elements of the covariance matrix are the variances of the features and the off-diagonal elements correspond to the correlation between the different features. The feature with the largest eigenvalue is the most discriminative feature, and the corresponding eigenvector is called the principal component of the data set. This eigenvector is placed on the first row of the transformation matrix. The transformed features do not correspond to any physically meaningful quantity [75].

Bibliography

[1] S. Sukkarieh, E. M. Nebot, and H. F. Durrant-Whyte, "A high integrity IMU/GPS navigation loop for autonomous land vehicle applications," *IEEE Transactions on Robotics and Automation*, vol. 15, pp. 572–578, June 1999.

[2] Y. Tao, H. Hu, and H. Zhou, "Integration of vision and inertial sensors for 3D arm motion tracking in home-based rehabilitation," *International Journal of Robotics Research*, vol. 26, pp. 607–624, June 2007.

[3] R. Zhu and Z. Zhou, "A real-time articulated human motion tracking using tri-axis inertial/magnetic sensors package," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 12, pp. 295–302, June 2004.

[4] I. J. Cox and G. T. Wilfong, ed., *Autonomous Robot Vehicles*. New York: Springer-Verlag, 1990. Section on Inertial Navigation edited by M. M. Kuritsky, and M. S. Goldstein.

[5] C. T. Leondes, ed., *Theory and Applications of Kalman Filtering*. London, U.K.: Technical Editing and Reproduction Lmtd., 1970. Sponsored by NATO Advisory Group for Aerospace Research and Development.

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

[7] B. Barshan and H. F. Durrant-Whyte, "Inertial navigation systems for mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 11, pp. 328–342, June 1995.

[8] C.-W. Tan and S. Park, "Design of accelerometer-based inertial navigation systems," *IEEE Transactions on Instrumentation and Measurement*, vol. 54, pp. 2520–2530, December 2005.

[9] J. Vaganay and M. J. Aldon, "Attitude estimation for a vehicle using inertial sensors," in *preprints of the 1st IFAC International Workshop on Intelligent Autonomous Vehicles*, pp. 89–94, Southampton, Hampshire, U.K., 18–21 April 1993. Pergamon Press, edited by D. Charnley.

[10] J. G. Nichol, S. P. N. Singh, K. J. Waldron, L. R. Palmer, III, and D. E. Orin, "System design of a quadrupedal galloping machine," *International Journal of Robotics Research*, vol. 23, pp. 1013–1027, October-November 2004.

[11] P.-C. Lin, H. Komsuoglu, and D. E. Koditschek, "Sensor data fusion for body state estimation in a hexapod robot with dynamical gaits," *IEEE Transactions on Robotics*, vol. 22, pp. 932–943, October 2006.

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

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

[14] K. Maenaka, "MEMS inertial sensors and their applications," in *5th International Conference on Networked Sensing Systems*, pp. 71–73, Kanazawa, Japan, 17–19 June 2008.

[15] M. J. Mathie, B. G. Celler, N. H. Lovell, and A. C. F. Coster, "Classification of basic daily movements using a triaxial accelerometer," *Medical and Biological Engineering and Computing*, vol. 42, pp. 679–687, September 2004.

[16] E. Jovanov, A. Milenkovic, C. Otto, and P. C de Groen, "A wireless body area network of intelligent motion sensors for computer assisted physical rehabilitation," *Journal of NeuroEngineering and Rehabilitation*, vol. 2, no. 6, 2005.

[17] W. H. Wu, A. A. T. Bui, M. A. Batalin, D. Liu, and W. J. Kaiser, "Incremental diagnosis method for intelligent wearable sensor system," *IEEE Transactions on Information Technology in Biomedicine*, vol. 11, pp. 553–562, September 2007.

[18] J. Pärkkä, M. Ermes, P. Korpipää, J. Mäntyjärvi, J. Peltola, and I. Korhonen, "Activity classification using realistic data from wearable sensors," *IEEE Transactions on Information Technology in Biomedicine*, vol. 10, pp. 119–128, January 2006.

[19] M. Ermes, J. Pärkkä, J. Mäntyjärvi, and I. Korhonen, "Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions," *IEEE Transactions on Information Technology in Biomedicine*, vol. 12, pp. 20–26, January 2008.

[20] R. Aylward and 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*, pp. 134–139, Paris, France, 4–8 June 2006.

[21] J. Lee and I. Ha, "Real-time motion capture for a human body using accelerometers," *Robotica*, vol. 19, no. 6, pp. 601–610, 2001.

[22] T. Shiratori and J. K. Hodgins, "Accelerometer-based user interfaces for the control of a physically simulated character," *ACM Transactions on Graphics (SIGGRAPH Asia 2008)*, vol. 27, August 2008.

[23] W. Zijlstra and K. Aminian, "Mobility assessment in older people: new possibilities and challenges," *European Journal of Ageing*, vol. 4, pp. 3–12, March 2007.

[24] M. J. Mathie, A. C. F. Coster, N. H. Lovell, and B. G. Celler, "Accelerometry: providing an integrated, practical method for long-term, ambulatory monitoring of human movement," *Physiological Measurement*, vol. 25, pp. R1–R20, April 2004.

[25] W. Y. Wong, M. S. Wong, and K. H. Lo, "Clinical applications of sensors for human posture and movement analysis: A review," *Prosthetics and Orthotics International*, vol. 31, pp. 62–75, March 2007.

[26] C. V. C. Bouten, K. T. M. Koekkoek, M. Verduin, R. Kodde, and J. D. Janssen, "A triaxial accelerometer and portable data processing unit for the assessment of daily physical activity," *IEEE Transactions on Biomedical Engineering*, vol. 44, pp. 136–147, March 1997.

[27] N. Kern, B. Schiele, and A. Schmidt, "Multi-sensor activity context detection for wearable computing," in *Proceedings of European Symposium on Ambient Intelligence, Lecture Notes in Computer Science*, vol. 2875, (Eindhoven, The Netherlands), pp. 220–232, November 2003. `citeseer.ist.psu.edu/kern03multisensor.html`.

[28] L. Bao and S. S. Intille, "Activity recognition from user-annotated acceleration data," in *Proceedings of Pervasive Computing, Lecture Notes in Computer Science*, vol. 3001, pp. 1–17, 2004. `web.media.mit.edu/~intille/papers-files/BaoIntille04.pdf`.

[29] P. H. Veltink, H. B. J. Bussmann, W. de Vries, W. L. J. Martens, and R. C. Van Lummel, "Detection of static and dynamic activities using uniaxial accelerometers," *IEEE Transactions on Rehabilitation Engineering*, vol. 4, pp. 375–385, December 1996.

[30] K. Kiani, C. J. Snijders, and E. S. Gelsema, "Computerized analysis of daily life motor activity for ambulatory monitoring," *Technology and Health Care*, vol. 5, pp. 307–318, October 1997.

[31] M. J. Mathie, A. C. F. Coster, N. H. Lovell, and B. G. Celler, "Detection of daily physical activities using a triaxial accelerometer," *Medical and Biological Engineering and Computing*, vol. 41, pp. 296–301, May 2003.

[32] D. M. Karantonis, M. R. Narayanan, M. Mathie, N. H. Lovell, and B. G. Celler, "Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring," *IEEE Transactions on Information Technology in Biomedicine*, vol. 10, pp. 156–167, January 2006.

[33] J. Lester, T. Choudhury, N. Kern, G. Borriello, and B. Hannaford, "A hybrid discriminative/generative approach for modeling human activities," in *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI-05)*, pp. 766–772, Edinburgh, Scotland, U.K., 30 July–5 August 2005.

[34] J. Lester, T. Choudhury, and G. Borriello, "A practical approach to recognizing physical activities," in *Proceedings of Pervasive Computing, Lecture Notes in Computer Science*, vol. 3968, pp. 1–16, 2006.

[35] S. Song, J. Jang, and S. Park, "A phone for human activity recognition using triaxial acceleration sensor," in *International Conference on Consumer Electronics*, pp. 1–2, Las Vegas, U.S.A., 9–13 January 2008.

[36] T. Liu, Y. Inoue, K. Shibata, and X. Tang, "A wearable inertial sensor system for human motion analysis," in *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pp. 409–413, Espoo, Finland, 27–30 June 2005.

[37] J. Mäntyjärvi, J. Himberg, and T. Seppänen, "Recognizing human motion with multiple acceleration sensors," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, vol. 2, pp. 747–752, Arizona, U.S.A., 7–10 October 2001.

[38] U. Maurer, A. Smailagic, D. P. Siewiorek, and M. Deisher, "Activity recognition and monitoring using multiple sensors on different body positions," in *IEEE International Workshop on Wearable and Implantable Body Sensor Networks*, pp. 113–116, Boston, MA, U.S.A., 3–5 April 2006.

[39] J. Yin, Q. Yang, and J. J. Pan, "Sensor-based abnormal human-activity detection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, pp. 1082–1090, August 2008.

[40] U. Lindemann, A. Hock, M. Stuber, W. Keck, and C. Becker, "Evaluation of a fall detector based on accelerometers: a pilot study," *Medical and Biological Engineering and Computing*, vol. 43, pp. 548–551, October 2005.

[41] K. Hauer, S. E. Lamb, E. C. Jorstad, C. Todd, and C. Becker, "Systematic review of definitions and methods of measuring falls in randomised controlled fall prevention trials," *Age and Ageing*, vol. 35, pp. 5–10, January 2006.

[42] J. B. Bussmann, P. J. Reuvekamp, P. H. Veltink, W. L. Martens, and H. J. Stam, "Validity and reliability of measurements obtained with an

"activity monitor" in people with and without transtibial amputation," *Physical Therapy*, vol. 78, pp. 989–998, September 1998.

[43] K. Aminian, P. Robert, E. E. Buchser, B. Rutschmann, D. Hayoz, and M. Depairon, "Physical activity monitoring based on accelerometry: validation and comparison with video observation," *Medical and Biological Engineering and Computing*, vol. 37, pp. 304–308, December 1999.

[44] B. Najafi, K. Aminian, A. Paraschiv-Ionescu, F. Loew, C. J. Büla, and P. Robert, "Ambulatory system for human motion analysis using a kinematic sensor: monitoring of daily physical activity in the elderly," *IEEE Transactions on Biomedical Engineering*, vol. 50, pp. 711–723, June 2003.

[45] M. Uiterwaal, E. B. C. Glerum, H. J. Busser, and R. C. van Lummel, "Ambulatory monitoring of physical activity in working situations, a validation study," *Journal of Medical Engineering and Technology*, vol. 22, pp. 168–172, July/August 1998.

[46] F. Foerster, M. Smeja, and J. Fahrenberg, "Detection of posture and motion by accelerometry: a validation study in ambulatory monitoring," *Computers in Human Behavior*, vol. 15, pp. 571–583, September 1999.

[47] F. R. Allen, E. Ambikairajah, N. H. Lovell, and B. G. Celler, "Classification of a known sequence of motions and postures from accelerometry data using adapted Gaussian mixture models," *Physiological Measurement*, vol. 27, pp. 935–951, October 2006.

[48] T. B. Moeslund and E. Granum, "A survey of computer vision-based human motion capture," *Computer Vision and Image Understanding*, vol. 81, pp. 231–268, March 2001.

[49] T. B. Moeslund, A. Hilton, and V. Krüger, "A survey of advances in vision-based human motion capture and analysis," *Computer Vision and Image Understanding*, vol. 104, pp. 90–126, November-December 2006.

[50] L. Wang, W. Hu, and T. Tan, "Recent developments in human motion analysis," *Pattern Recognition*, vol. 36, pp. 585–601, March 2003.

[51] J. K. Aggarwal and Q. Cai, "Human motion analysis: a review," *Computer Vision and Image Understanding*, vol. 73, pp. 428–440, March 1999.

[52] A. F. Bobick and J. W. Davis, "The recognition of human movement using temporal templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, pp. 257–267, March 2001.

[53] Y. Wang, H. Jiang, M. S. Drew, Z. Li, and G. Mori, "Unsupervised discovery of action classes," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 1654–1661, New York, 17–22 June 2006.

[54] M. Brand and V. Kettnaker, "Discovery and segmentation of activities in video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 844–851, August 2000.

[55] N. Robertson and I. Reid, "A general method for human activity recognition in video," *Computer Vision and Image Understanding*, vol. 104, pp. 232–248, November-December 2006.

[56] M. Leo, T. D'Orazio, I. Gnoni, P. Spagnolo, and A. Distante, "Complex human activity recognition for monitoring wide outdoor environments," in *Proceedings of the International Conference on Pattern Recognition*, vol. 4, pp. 913–916, Cambridge, U.K., 23–26 August 2004.

[57] Y. Shi, A. Bobick, and I. Essa, "Learning temporal sequence model from partially labeled data," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, New York, 17–22 June 2006.

[58] T. Xiang and S. Gong, "Beyond tracking: modeling activity and understanding behavior," *International Journal of Computer Vision*, vol. 67, no. 1, pp. 21–51, 2006.

[59] P. C. Ribeiro and J. Santos-Victor, "Human activity recognition from video: modeling, feature selection and classification architecture," in *Proceedings of the International Workshop on Human Activity Recognition and Modelling*, pp. 61–78, Oxford, U.K., 9 September 2005.

[60] J. Rittscher, A. Blake, and S. J. Roberts, "Towards the automatic analysis of complex human body motions," *Image and Vision Computing*, vol. 20, pp. 905–916, October 2002.

[61] E. Ramasso, C. Panagiotakis, D. Pellerin, and M. Rombaut, "Human action recognition in videos based on the transferable belief model," *Pattern Analysis and Applications*, vol. 11, no. 1, pp. 1–19, 2008.

[62] D. Roetenberg, P. J. Slycke, and P. H. Veltink, "Ambulatory position and orientation tracking fusing magnetic and inertial sensing," *IEEE Transactions on Biomedical Engineering*, vol. 54, pp. 883–890, May 2007.

[63] B. Najafi, K. Aminian, F. Loew, Y. Blanc, and 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 Transactions on Biomedical Engineering*, vol. 49, pp. 843–851, August 2002.

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

[65] Special Issue on the 2nd Workshop on Integration of Vision and Inertial Sensors (InerVis05) *International Journal of Robotics Research*, vol. 26, June 2007.

[66] T. Viéville and O. D. Faugeras, *Cooperation of the Inertial and Visual Systems*, vol. F63 of *NATO ASI Series*, pp. 339–350. Berlin, Heidelberg, Germany: Springer-Verlag, 59th ed., 1990. in Traditional and Non-Traditional Robotic Sensors.

[67] O. Tunçel, K. Altun, and B. Barshan, "Classification of leg motions by processing gyroscope signals (Jiroskop sinyallerinin işlenmesiyle bacak hareketlerinin sınıflandırılması)," in *Proceedings of the IEEE 17th Conference on Signal Processing, Communications and Applications (CD-ROM)*, Side, Antalya, Turkey, 9–11 April 2009. in Turkish.

[68] T. Shelley and J. Barrett, "Vibrating gyro to keep cars on route," *Eureka on Campus, Engineering Materials and Design*, vol. 4, p. 17, Spring 1992.

[69] Murata Manufacturing Co., Ltd., *Murata Gyrostar ENV-05A Piezoelectric Vibratory Gyroscope Datasheet*, 1994.

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

[71] E. Feig and S. Winograd, "Fast algorithms for the discrete cosine transform," *IEEE Transactions on Signal Processing*, vol. 40, no. 9, pp. 2174–2193, 1992.

[72] A. K. Jain, *Fundamentals of Digital Image Processing*. New Jersey: Prentice Hall, 1989.

[73] A. Webb, *Statistical Pattern Recognition*. New York: John Wiley & Sons, 2002.

[74] M. Rosenblatt, "Remarks on some nonparametric estimates of a density function," *Annals of Mathematical Statistics*, vol. 27, no. 3, pp. 832–837, 1956.

[75] M. Nadler and E. P. Smith, *Pattern Recognition Engineering*. New York: John Wiley & Sons, Inc., 1993.

[76] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*. Orlando, FL, U.S.A.: Academic Press, Inc, 2006.

[77] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. San Diego: Academic Press, 2nd ed., 1990.

[78] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*. New York: Chapman and Hall, 1986.

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

[80] E. Keogh and C. A. Ratanamahatana, "Exact indexing of dynamic time warping," *Knowledge and Information Systems*, vol. 7, pp. 358–386, March 2005.

[81] T. M. Rath and R. Manmatha, "Word image matching using dynamic time warping," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. II–521–II–527, Madison, Wisconsin, 16–22 June 2003.

[82] T. Konidaris, B. Gatos, S. J. Perantonis, and A. Kesidis, "Keyword matching in historical machine-printed documents using synthetic data, word portions and dynamic time warping," in *The Eighth IAPR International Workshop on Document Analysis Systems*, pp. 539–545, Nara, Japan, 16–19 September 2008.

[83] M. F. Zanuy, "On-line signature recognition based on VQ-DTW," *Pattern Recognition*, vol. 40, pp. 981–992, March 2007.

[84] W. D. Chang and J. Shin, "Modified dynamic time warping for stroke-based on-line signature verification," in *Proceedings of the Ninth International*

Conference on Document Analysis and Recognition, vol. 2, pp. 724–728, Curitiba, State of Parana, Brazil, 23–26 September 2007.

[85] N. V. Boulgouris, K. N. Plataniotis, and D. Hatzinakos, "Gait recognition using dynamic time warping," in *IEEE 6th Workshop on Multimedia Signal Processing*, pp. 263–266, Siena, Italy, 29 September–1 October 2004.

[86] B. Huang and W. Kinsner, "ECG frame classification using dynamic time warping," in *IEEE Canadian Conference on Electrical and Computer Engineering*, vol. 2, pp. 1105–1110, Winnipeg, Manitoba, 12–15 May 2002.

[87] H. J. L. M. Vullings, M. H. G. Verhaegen, and H. B. Verbruggen, "Automated ECG segmentation with dynamic time warping," in *Proceedings of the 20th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, vol. 20, pp. 163–166, Hong Kong, 29 October–1 November 1998.

[88] V. Tuzcu and S. Nas, "Dynamic time warping as a novel tool in pattern recognition of ECG changes in heart rhythm disturbances," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, vol. 1, pp. 182–186, Hawaii, U.S.A., 10–12 October 2005.

[89] Z. M. Kovács-Vajna, "A fingerprint verification system based on triangular matching and dynamic time warping," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 1266–1276, November 2000.

[90] L. E. M. López, R. P. Elías, and J. V. Tavira, "Face localization in color images using dynamic time warping and integral projections," in *Proceedings of International Joint Conference on Neural Networks*, pp. 892–896, Orlando, Florida, 12–17 August 2007.

[91] T. Parsons, *Voice and Speech Processing.* New York: McGraw-Hill Book Company, 1986.

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

[93] C. W. Hsu, C. C. Chang, and C. J. Lin, *A Practical Guide to Support Vector Classification*, 2008.

[94] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," *Journal of Machine Learning Research*, vol. 2, pp. 45–66, November 2001.

[95] B. Schölkopf and A. J. Smola, *Learning with Kernels*. Massachusetts: The MIT Press, 2002.

[96] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. New York: John Wiley & Sons, Inc., 2001. 2nd edition.

[97] R. P. W. Duin, P. Juszczak, P. Paclik, E. Pekalska, D. de Ridder, and D. M. J. Tax, *A Matlab Toolbox for Pattern Recognition, PRTools4*. 2004.

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

[99] M. Kangas, A. Konttila, P. Lindgren, I. Winblad, and T. Jämsä, "Comparison of low-complexity fall detection algorithms for body attached accelerometers," *Gait & Posture*, vol. 28, no. 2, pp. 285–291, 2008.

[100] N. Noury, A. Fleury, P. Rumeau, A. K. Bourke, G. O. Laighin, V. Rialle, and 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*, pp. 1663–1666, August 2007.

[101] D. H. Kil and F. B. Shin, *Pattern Recognition and Prediction with Applications to Signal Characterization*. New York: American Institute of Physics, 1996.

[102] L. I. Smith, "A Tutorial on Principal Components Analysis," tech. rep., 2002. `cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf`.