ACTIVITY RECOGNITION INVARIANT TO POSITION AND ORIENTATION OF WEARABLE MOTION SENSOR UNITS

A DISSERTATION SUBMITTED TO

THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR

THE DEGREE OF

DOCTOR OF PHILOSOPHY

IN

ELECTRICAL AND ELECTRONICS ENGINEERING

By

Aras Yurtman

April 2019

ACTIVITY RECOGNITION INVARIANT TO POSITION AND ORIENTATION OF WEARABLE MOTION SENSOR UNITS

By Aras Yurtman

April 2019

We certify that we have read this dissertation and that in our opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Billur Barshan Özaktaş (Advisor)

Emine Ülkü Sarıtaş Çukur

Çiğdem Gündüz Demir

Nevzat Güneri Gençer

Aykut Yıldız

Approved for the Graduate School of Engineering and Science:

Ezhan Karaşan
Director of the Graduate School

ABSTRACT

ACTIVITY RECOGNITION INVARIANT TO POSITION AND ORIENTATION OF WEARABLE MOTION SENSOR UNITS

Aras Yurtman
Ph.D. in Electrical and Electronics Engineering
Advisor: Billur Barshan Özaktaş
April 2019

We propose techniques that achieve invariance to the placement of wearable motion sensor units in the context of human activity recognition. First, we focus on invariance to sensor unit orientation and develop three alternative transformations to remove from the raw sensor data the effect of the orientation at which the sensor unit is placed. The first two orientation-invariant transformations rely on the geometry of the measurements, whereas the third is based on estimating the orientations of the sensor units with respect to the Earth frame by exploiting the physical properties of the sensory data. We test them with multiple state-of-the-art machine-learning classifiers using five publicly available datasets (when applicable) containing various types of activities acquired by different sensor configurations. We show that the proposed methods achieve a similar accuracy with the reference system where the units are correctly oriented, whereas the standard system cannot handle incorrectly oriented sensors. We also propose a novel non-iterative technique for estimating the orientations of the sensor units based on the physical and geometrical properties of the sensor data to improve the accuracy of the third orientation-invariant transformation. All of the three transformations can be integrated into the pre-processing stage of existing wearable systems without much effort since we do not make any assumptions about the sensor configuration, the body movements, and the classification methodology.

Secondly, we develop techniques that achieve invariance to the positioning of the sensor units in three ways: (1) We propose transformations that are applied on the sensory data to allow each unit to be placed at any position within a pre-determined body part. (2) We propose a transformation technique to allow the units to be interchanged so that the user does not need to distinguish between them before positioning. (3) We employ three different techniques to classify the activities based on a single sensor unit, whereas the training set may contain data acquired by multiple units placed at different positions. We combine (1) with (2) and also

with (3) to achieve further robustness to sensor unit positioning. We evaluate our techniques on a publicly available dataset using seven state-of-the-art classifiers and show that the reduction in the accuracy is acceptable, considering the flexibility, convenience, and unobtrusiveness in the positioning of the units.

Finally, we combine the position- and orientation-invariant techniques to simultaneously achieve both. The accuracy values are much higher than those of random decision making although some of them are significantly lower than the reference system with correctly placed units. The trade-off between the flexibility in sensor unit placement and the classification accuracy indicates that different approaches may be suitable for different applications.

Keywords: Wearable sensing, human activity recognition, sensor placement, sensor position, sensor orientation, position-invariant sensing, orientation-invariant sensing, orientation estimation, motion sensors, inertial sensors, accelerometer, gyroscope, magnetometer.

ÖZET

GİYİLEBİLİR HAREKET ALGILAYICI ÜNİTELERİNİN KONUM VE YÖNLERİNDEN BAĞIMSIZ OLARAK AKTİVİTE TANIMA

Aras Yurtman
Elektrik ve Elektronik Mühendisliği, Doktora
Tez Danışmanı: Billur Barshan Özaktaş
Nisan 2019

İnsan aktivitelerinin tanınması bağlamında giyilebilir hareket algılayıcı ünitelerinin yerleşimine karşı değişmezlik elde eden yöntemler öne sürülmektedir. İlk olarak, algılayıcı ünitelerinin yönlerine yoğunlaşılarak ünitelerin takılış yönünün etkisini ham algılayıcı verilerinden çıkaran üç alternatif dönüşüm geliştirilmektedir. Yönden-bağımsız dönüşümlerin ilk ikisi, ölçümlerin geometrisine dayanırken, üçüncüsü, algılayıcı verilerinin fiziksel özellikleri kullanılarak algılayıcı ünitelerinin dünyanın koordinat sistemine göre yönlerinin kestirimini esas almaktadır. Bu yöntemler, birden fazla güncel makine öğrenme sınıflandırıcısı ile (mümkün olduğunda) herkese açık, çeşitli aktivite tiplerini içeren ve farklı algılayıcı düzenleşimleriyle elde edilmiş olan beş veri kümesi kullanılarak değerlendirilmiştir. Alışılagelmiş sistem, yanlış yönlü algılayıcılarla baş edemezken, bu yöntemlerin, algılayıcı yönlerinin doğru olduğu referans sistemle benzer başarım elde ettiği gösterilmiştir. Üçüncü dönüşümün başarımını arttırmak için, algılayıcı verilerinin fiziksel ve geometrik özelliklerine dayanan, algılayıcı üniteleri için yenilikçi ve yinelemesiz bir yön kestirim yöntemi de öne sürülmektedir. Algılayıcı düzenleşimleri, beden hareketleri ve sınıflandırma yöntemi ile ilgili herhangi bir varsayımda bulunulmadığı için, yönden-bağımsız yöntemlerin üçü de, var olan giyilebilir sistemlerin ön-işleme aşamalarına kolayca dahil edilebilir.

İkinci olarak, algılayıcı ünitelerinin konumlandırılmasına karşı üç farklı şekilde değişmezlik elde eden yöntemler geliştirilmektedir: (1) Her bir giyilebilir ünitenin önceden belirlenmiş bir beden parçası üzerinde herhangi bir konuma yerleştirilmesine izin vermek için algılayıcı verilerine uygulanan iki farklı dönüşüm öne sürülmektedir. (2) Kullanıcının, üniteleri yerleştirmeden önce birbirinden ayırt etmesine gerek kalmaması için, ünitelerin değiş tokuş edilebilmelerine izin veren bir dönüşüm öne sürülmektedir. (3) Öğrenme verileri birden fazla konuma yerleştirilmiş birden fazla üniteden elde edilen veriler içermesine karşın, aktiviteleri tek bir algılayıcı ünitesine dayanarak sınıflandırabilen üç farklı yöntem kullanılmaktadır. Daha fazla gürbüzlük elde etmek için (1)'deki yöntem, (2) ile ve ayrıca (3) ile birleştirilmektedir. Önerilen yöntemler, yedi güncel sınıflandırıcı kullanılarak herkese açık bir veri kümesi üzerinde gerçeklenmiş ve sağlanan esneklik düşünüldüğünde başarımdaki düşüşün kabul edilebilir olduğu değerlendirilmiştir.

Son olarak, konum ve yönden bağımsız yöntemler, bu iki önemli özelliğin aynı anda sağlanabilmesi için tümleştirilmiştir. Başarım değerleri, doğru biçimde takılmış olan algılayıcı ünitelerinin başarımından daha düşük olsa da, rastgele karar verme stratejisine göre çok daha yüksektir. Algılayıcı ünitelerinin yerleşimi ve sınıflandırma başarımı arasındaki ödünleşime göre, farklı uygulamalar için farklı yöntem seçimleri yapılabilmektedir.

Anahtar sözcükler: Giyilebilir algılama, insan aktivitesi tanıma, algılayıcı yerleşimi, algılayıcı konumu, algılayıcı yönü, konumdan bağımsız algılama, yönden bağımsız algılama, yön kestirimi, hareket algılayıcıları, ataletsel sensörler (eylemsizlik duyucuları), ivmeölçer, dönüölçer (jiroskop), manyetometre.

Acknowledgement

I would like to express my gratitude to my supervisor Prof. Billur Barshan for her support in the development of this thesis.

I would like to thank the members of my Thesis Tracking Committee, Asst. Prof. Emine Ülkü Sarıtaş Çukur and Assoc. Prof. Çiğdem Gündüz Demir, for their supervision, support, encouragement, and suggestions for the completion of my thesis. I would also like to thank Prof. Nevzat Güneri Gençer and Asst. Prof. Aykut Yıldız for accepting to review this thesis and to serve as jury members.

I greatly appreciate the support I have received from the Rector Prof. Abdullah Atalar; the director of Graduate School of Engineering and Science, Prof. Ezhan Karaşan; and the Associate Provost Prof. Hitay Özbay towards obtaining my Ph.D. degree. I would like to thank Damla Ercan Atay, the administrative assistant at the graduate school.

I am grateful for the support of the Department of Electrical and Electronics Engineering throughout my Ph.D. study. I would like to thank the department chair Prof. Orhan Arıkan and the administrative assistants Mürüvet Parlakay, Aslı Tosuner, Tuğba Özdemir, and Görkem Uğuroğlu for their motivation and help in academic and administrative issues.

This long journey would not have been possible without the help of my friends Merve Begüm Terzi, Muhammad Anjum Qureshi, Muhammad Zakwan, Saeed Ahmed, Ahmet Safa Öztürk, Dilan Öztürk, Hasan Eftun Orhon, Okan Demir, Cemre Arıyürek, Yalım İşleyici, Reyhan Ergün, Bahram Khalichi, Seçil Eda Doğan, Polat Göktaş, Serkan Sarıtaş, Ahmet Dündar Sezer, Ali Nail İnal, İsmail Uyanık, Meysam Ghomi, Soheil Taraghinia, Sinan Alemdar, Salman Ul Hassan Dar, Mert Özateş, Ali Alp Akyol, Ertan Kazıklı, Oğuzcan Dobrucalı, Mustafa Şahin Turan, Mehdi Dabirnia, Sina Rezaei Aghdam, Ersin Yar, Bahadır Çatalbaş, Caner Odabaş, Adamu Abdullahi, Mahdi Shakiba Herfeh, Alireza Nooraiepour, Muhammad Umar B. Niazi, Mustafa Gül, Cem Emre Akbaş, Rasim Akın Sevimli, Gökçe Kuralay, Parisa Sharif, and Mohammad Tofighi.

I appreciate the support of the technical staff Yusuf Çalışkan, Ergün Hırlakoğlu, Onur Bostancı, Ufuk Tufan, and Mehmet Bal who helped me in serving for the department. In particular, for their support in the organization of the Graduate Research Conference 2016–2019 in which I took part, I would like to thank the faculty members Asst. Prof. Cem Tekin, Prof. Süleyman Serdar Kozat, Prof. Orhan Arıkan, and Prof. Sinan Gezici as well as the members of the IEEE Bilkent University Student Branch Ayça Takmaz, Ömer Karakaş, Erdem Bıyık, Mert Canatan, Hasan Emre Erdemoğlu, Elif Aygün, Çağan Selim Çoban, Hikmet Demir, Doğa Gürgünoğlu, Mehmet Ali Karakamış, Abdulkadir Erol, Yiğit Efe Erginbaş, Deniz Ulusel, and many others whom I could not name here.

I would like to express my deepest gratitude to the cats İrma (*aka* Cankız), Mine, Tesla (*aka* Sultan), and Sütlaç (*aka* Pamuk, Boncuk, Paspas, and Snowball) for boosting my endorphins by visiting my house and my office.

Last but not the least, I would like to thank my parents Nilgül Yurtman and Dr. Talat Yurtman for supporting me spiritually throughout my Ph.D. study and my life in general.

Contents

List of Figures

List of Tables

List of Abbreviations

OIT	Orientation-Invariant Transformation
OEM	Orientation Estimation Method
PCA	Principal Component Analysis
SVD	Singular Value Decomposition
U-SVD	Unit-Based Singular Value Decomposition
1-NN	1-Nearest Neighbor
k-NN	k-Nearest Neighbor
ANN	Artificial Neural Networks
BDM	Bayesian Decision Making
LDC	Linear Discriminant Classifier
OMP	Orthogonal Matching Pursuit
RF	Random Forest
SVM	Support Vector Machines
MAP	Maximum a Posteriori
RBF	Radial Basis Function
L1O	Leave-one-Subject-Out
KF	Kalman Filter
GD	Gradient-Descent
GN	Gauss-Newton
LM	Levenberg-Marquardt
ENU	East-North-Up
NED	North-East-Down
RIU	Randomly Interchanged Units
RD	Random Displacement
RR	Random Rotation
SUC	Single-Unit Classification
nD	n-Dimensional
DFT	Discrete Fourier Transform
GPS	Global Positioning System

Chapter 1

Introduction

Human activity recognition has been an active field of research since the late 1990s, with applications including but not limited to healthcare, surveillance, entertainment, and military systems [1–3]. The recognized activities can be daily activities such as walking and sitting as well as sports activities such as jumping and running on a treadmill. Recent work on automatically recognizing daily activities focuses on machine learning algorithms that rely on simultaneous input from several different sensor modalities such as visual, inertial, acoustic, force, pressure, strain, physiological, and kinetic sensors, among others [4–7]. Collecting information about a user's activities for ambient-assisted living in smart homes and detecting abnormal behavior to assist the elderly or people with special needs are challenging research issues [8, 9]. These systems aim to maintain the user's independence, enhancing their personal safety and comfort and delaying the process of moving to a care home. However, automatic monitoring of people performing daily activities should be done without restricting their independence, intruding on their privacy, or degrading their quality of life.

A commonly used approach in designing smart environments involves the use of one or more types of external sensors in a complementary fashion (e.g., cameras and tactile sensors), usually with relatively high installation cost and heavy demands on computing power [10, 11]. If a single camera is used, the 3D scene is projected onto a 2D one, with significant information loss. Other people or pets moving around may easily confuse such systems. Occlusion or shadowing of points of interest (by human body parts or objects in the surroundings) is resolved by using 2D projections from multiple cameras in the environment to reconstruct the 3D scene. Each camera needs to be individually calibrated and suffers from the correspondence problem. To resolve the latter, points of interest on the human body are pre-identified by placing special, visible markers at those points and the positions of the markers are recorded by cameras. Processing and storing camera recordings is costly and camera systems obviously interfere with privacy. Recorded data are highly sensitive to privacy breaches when transmitted or stored [12]. Continuous monitoring may cause stress and discomfort on the subject and may subsequently cause changes in his natural movements.

The main advantage of embedding external sensors in the environment is that the person does not have to wear or carry any sensors or devices [13, 14]. This approach may also eliminate problems related to placing the sensors incorrectly on the body, although some camera systems do require wearing/pasting on special tags or markers as mentioned above. Designing smart environments may be acceptable when the activities of the person are confined to certain parts of a building. However, when the activities are performed both indoors and outdoors and involve going from one place to another (e.g., riding a vehicle, going shopping, commuting, etc.), this approach becomes unsuitable. It imposes restrictions on the mobility of the person since the system operates only in the limited environment being monitored.

The use of wearable motion sensors in activity recognition has pervaded since this approach is superior to using external sensors in many respects [15]. The required infrastructure and associated costs of wearable sensors are much lower than designing smart environments. Unlike visual motion-capture systems that require a free line of sight, wearable sensors can be flexibly used inside or behind objects without occlusion. They can acquire the required 3D motion data directly on the spot without the need for multiple camera projections. The 1D signals acquired from the multiple axes of wearable motion sensors are much simpler and faster to process. Because they are light, comfortable, and easy to carry, wearable sensors do not restrict people to a studio-like environment and can operate both indoors and outdoors, allowing free pursuit of activities without intruding on privacy.

Wearable systems are criticized mainly because people may forget, neglect, or not want to wear them. If they are battery operated, batteries need to be recharged or replaced from time to time. However, with the advances of the MEMS (Micro-Electro-Mechanical Systems) technology, these devices have been miniaturized. Their lightness, low power consumption, and wireless use have eliminated the concerns related to portability and discomfort. Furthermore, the algorithms developed can be easily embedded to a device or accessory that the person normally carries, such as a mobile phone, watch, bracelet, or a hearing aid. Wearable sensors are thus a very suitable domain for automatic monitoring and classification of daily activities, and we have chosen to follow this approach in our works [16–25].

With the advancements mentioned above, proper placement of wearable devices on the body has become a challenging task for the user, making wearables prone to be fixed to the body at incorrect positions and orientations. In most applications of wearable sensing, it is assumed that sensor units are placed at pre-determined positions and orientations that remain constant over time [26]. This assumption may be obtrusive because the user needs to be attentive to placing the sensor unit correctly and to keeping it at the same position and orientation. In practice, users may place the sensor units incorrectly on the body and even if this is not the case, their positions and orientations may gradually change because of loose attachments and body movement. If the sensor units are worn on specially designed clothing or accessories, these may vibrate or move relative to the body. Often, elderly, disabled, injured people or children also need to wear these sensors for health, state, or activity monitoring [16, 27], and may have difficulty placing them correctly. Hence, transformations that achieve position- and orientation-invariance to the placement of the sensor units would be advantageous for the users.

Earlier works on activity recognition that employ wearable sensors are reviewed in [28–30]. Incorrect placement of a wearable sensor unit may involve placing it at a different position as well as at a different orientation. The majority of existing wearable activity recognition studies neglect this issue and assume that the sensor units are properly placed on the body or, alternatively, use simple features (such as the vector norms) that are invariant to sensor unit placement. It would be a valuable contribution to develop wearable systems that are invariant to sensor unit position and orientation without any significant degradation in performance. In the former, sensor units can be placed anywhere on the same body part (e.g., lower arm) or on different body parts; in the latter, the units can be fixed to pre-determined positions at any orientation. Studies that consider both position and orientation invariance at the same time are reported but none of these works can handle incorrect placement of sensor units without a considerable loss in performance (between 20–50%) [31]. Existing studies on position- and orientation-invariant sensing have strong limitations and have been tested in very restricted scenarios. Thus, these two problems have not been completely solved to date. In this thesis, we focus on these problems and develop transformations for the generic activity recognition scheme that can be easily adapted to existing systems. Our aim is to develop techniques that can be applied at the pre-processing stage of the activity recognition framework to make this process robust to variable sensor unit placement. The proposed techniques can also be integrated into other applications of wearable sensing such as fall detection and classification [32], gesture recognition [33], leg motion classification [34, 35], authentication of users in mobile sensing systems [36], and automated evaluation of physical therapy exercises [16, 20].

We utilize widely available sensor types and do not make any assumptions about the sensor configuration, data acquisition, activities, and activity recognition procedure. Our proposed method can be integrated into existing activity recognition systems by applying transformations to the time-domain data in the pre-processing stage without modifying the rest of the system or the methodology. We outperform the existing methods for position and orientation invariance and achieve accuracies close to those of the standard activity recognition system in most cases.

We employ tri-axial wearable motion sensors (accelerometer, gyroscope, and magnetometers when applicable) to capture the body motions. Data acquired by these sensors not only contain information about the body movements but also about the placement of the sensor unit. However, these two types of information are coupled in the sensory data and it is not straightforward to decouple them. More specifically, a tri-axial accelerometer captures the vector sum of the gravity vector and the acceleration resulting from the motion. A tri-axial gyroscope detects the angular rate about each axis of sensitivity and can provide the angular velocity vector. A tri-axial magnetometer captures the vector sum of the magnetic field of the Earth and external magnetic sources, if any. We propose various techniques that preserve the information related to the body motions and satisfy invariance to the placement of the sensor unit at the same time. Our first aim is to minimize the reduction in the accuracy caused by the removal of the placement information. Our second aim is to achieve robustness to sensor unit placement so that the accuracy does not degrade.

1.1 Literature Review

The methods that have been proposed to achieve robustness to the placement of wearable motion sensor units are grouped as position- and orientation-invariant techniques as well as those that are invariant to both.

1.1.1 Invariance to Sensor Unit Position

A number of methods have been proposed to achieve robustness to the positioning of wearable motion sensor units [3, 26]. These methods can be grouped into four categories as described below, with their main features summarized in Table 1.1.

1.1.1.1 Extracting Position-Invariant Information from Sensor Data

Some studies propose to heuristically transform the sensor data or extract heuristic features to achieve robustness to the positioning of the sensor units. Reference [37] ignores acceleration data when there is too much rotational movement. It considers that the acceleration caused by rotational movements depends on the sensor

Table 1.1: Properties of the existing studies on position invariance.

reference	sensors used*	movement types (datasets are separated by "|")	stationary activities	orientation invariance
[37] Kunze 2008	accelerometer, gyroscope	forearm gestures | daily and sports activities | gym exercises	no	yes (based on initial calibration, only for gesture recognition)
[26] Kunze 2014	accelerometer, gyroscope	various datasets and methods analyzed	various datasets analyzed	yes (different methods analyzed separately)
[38] Jiang 2011	accelerometer	daily and sports activities	yes	yes (based on initial calibration)
[39] Hur 2017	accelerometer, gyroscope (for orientation invariance only), global positioning system (GPS)	daily activities (including transportation)	yes	yes
[40] Ravi 2016	accelerometer, gyroscope	unknown | daily activities | gait for Parkinson's disease | high-level activities	yes	yes
[41] Dernbach 2012	accelerometer	daily activities	yes	partial
[42] Wang 2009	accelerometer	daily	yes (a single stationary class)	yes
[43] Khan 2013	accelerometer	daily	yes (a single stationary activity)	no
[44] Khan 2014	accelerometer, air pressure sensor, microphone	daily and sports	yes (a single stationary class)	no
[45] Khan 2010	accelerometer	daily activities	yes (combined into a single stationary class)	no
[46] Lester 2006	accelerometer, microphone, barometer	daily activities (including high-level activities)	yes	no
[47] Sun 2010	accelerometer	daily activities	yes (a single stationary class)	partial
[33] Förster 2009	accelerometer	hand gestures (guided by geometric structures) | aerobic movements	no	no
[48] Reddy 2010	accelerometer, GPS	daily (including transportation)	yes (combined into a single stationary class)	yes
[49] Anjum 2013	accelerometer, gyroscope, GPS	daily	yes (a single stationary class)	yes
[50] Siirtola 2013	accelerometer	daily	yes (a single stationary class)	partial
[51] Doppler 2009	unknown (Euler angles used)	daily activities (only three)	no	yes (based on initial calibration)
[52] Henpraserttae 2011	accelerometer	daily activities	yes	partial
[53] Thiemjarus 2013	accelerometer	daily activities	yes	yes
[54] Chavarriaga 2011	accelerometer	hand gestures (guided by geometric structures) | aerobic movements	no	partial (separate from position invariance)
[55] Förster 2009	accelerometer	synthetic data | hand gestures (guided by geometric structures) | aerobic movements	no	no
[56] Kunze 2005	accelerometer	daily activities (including high-level activities)	yes	yes
[57] Xu 2012	accelerometer, bi-axial gyroscope	daily activities (including transition activities)	no	no
[58] Lu 2010	accelerometer, microphone, GPS	daily (including transportation)	yes (a single stationary class)	yes
[59] Martin 2013	accelerometer, gyroscope, magnetometer	daily activities	yes	no
[60] Sztyler 2016	accelerometer	daily activities	yes	partial
[31] Banos 2014	accelerometer	sports activities	no	partial
[61] Zhong 2014	accelerometer, gyroscope	gait | gait	no	yes

* accelerometers and gyroscopes are tri-axial unless stated otherwise

position, whereas the acceleration caused by linear movements is fixed over all sensor positions within the same body part under the assumption that the body part is rigid. The acceleration data are omitted only if the magnitude of the measured acceleration vector is not close to the magnitude of the Earth's gravity and the difference between these magnitudes (which roughly represents the magnitude of pure acceleration) is small compared to the magnitudes of the angular velocity and angular acceleration detected by the gyroscope. In [37], an additional low-pass filtered acceleration signal is also used in classification because it mostly contains the gravitational component, whose direction depends on the sensor unit orientation but not its position within the same body part. Low-pass filtering the acceleration data is proposed in [38] as well to achieve robustness to the positioning of the sensor units.

Reference [39] recognizes the uncommon activities "riding in a bus" and "riding in a subway" in addition to simple daily activities. The vibrations caused by the transportation types are experienced by the whole body; hence, the smart phone (whose motion sensors are used) is allowed to be placed at any position and orientation on the body. Classification is performed based on heuristic features extracted from the acceleration magnitude, discrete Fourier transform (DFT) of the vertical acceleration, and the speed measured by the global positioning system (GPS), which are obtained using built-in features of the Android mobile operating system.

1.1.1.2 Training Classifiers with Different Sensor Unit Positions

Another method to handle the varying positioning of the sensor units is to train an activity classifier in a generalized way to capture all (possible or considered) sensor unit positions. Some studies rely on such generalized classifiers only because data are acquired from different sensor configurations. This type of variation in the datasets makes the activity recognition inherently invariant to the positioning of the sensor units due to the variation in the training data, even though no specific techniques are used for this purpose. In particular, the studies [40–44] allow smart phones that contain motion sensors to be placed at any position on the body as a real-world scenario. However, it is not clear how differently the subjects positioned them in the experiments. Commonly used classifiers in these studies are Support Vector Machines (SVM), Artificial Neural Networks (ANN), decision trees, and naïve Bayes classifiers as well as deep learning approaches.

The datasets in [33, 45–50] contain data from multiple sensor units and the segments obtained from each unit are considered as separate training and test instances for generalized classification. In this scheme, the classifiers are trained with multiple unit positions and tested by using each position separately so that a single unit is sufficient for activity recognition. In [33, 46–48], generalized classifiers trained with multiple sensor unit positions achieve an accuracy slightly lower than position-specific classifiers. In [33], the accuracy further decreases when the leave-one-position-out method is used, where, for each position, a classifier trained with the data of the remaining positions is used. The studies [46–50] consider no more than several possible sensor unit positions and several activities, and the accuracy can drop abruptly if the numbers are increased. Reference [33], on the other hand, classifies aerobic movements with all the sensor units placed on the left leg and basic hand gestures with all the units on the right arm.

References [33, 51–53] analyze the case where training and test data originate from different sensor unit positions and provide the accuracy separately across the positions. In all of them, the accuracy significantly decreases if the classifier is trained with the data of a different sensor unit position because a single unit position is not sufficient to train a generalized classifier.

According to the results of the previous work, if training and test data originate from different sensor unit positions, an acceptable accuracy can be obtained if the training data include multiple positions, especially those that are on the same body part with the position at which the test data are acquired. On the other hand, training data acquired only from a single position cannot provide a classifier generalizable to the other positions.

1.1.1.3 Adapting Classifiers for New Sensor Unit Positions

Positioning the sensor units differently on the body causes variations in the features extracted from the acquired data. References [54, 55] assume that these variations only cause some shifts in the class means in the feature space and calculate the amount of shifts in an unsupervised way (i.e., without using the class labels) given new data obtained from a different sensor unit position. This assumption seems to hold for the position changes that occur only within the same body part (such as the left lower leg or the torso), as both studies obtain unsatisfactory classification accuracies across the different body parts, even across the lower and the upper arm/leg, which shows that different body parts have different motion characteristics even though they are close to each other, as stated in [26]. Another drawback of these adaptation-based methods is the difficulty of deciding when to start the adaptation process, which is suggested to be manually initiated by the user in [55], whereas this issue is not mentioned at all in [54].

1.1.1.4 Classifying Sensor Unit Positions

Some studies classify the sensor unit's position on the body during a pre-determined set of activities assuming that there is a finite set of positions, which is not valid in some scenarios. This position information can be used for context awareness or to select an activity classifier that is trained specifically for that position. Reference [56] distinguishes the walking activity from other activity types by training a generalized classifier for four pre-determined sensor positions. Recordings of the walking activity of at least one minute duration are used to classify the sensor unit's position. In this scheme, it is assumed that the sensor unit remains at the same position for at least a few minutes. Both classification techniques are invariant to the sensor unit orientations as the magnitude of the acceleration vectors are used.

In [57], a sparse representation classifier is trained for all activity-sensor unit position pairs. Then, Bayesian fusion is used to recognize the activity independently of the sensor unit position and to classify the position of the unit independently of the performed activity. Reference [58] considers each activity-position pair as a different class so that the activity and sensor unit position can be simultaneously classified. Another study [59] follows a two-stage approach by first classifying the sensor unit's position on the body and then recognizing the activity type using a classifier specifically trained for that position. By evaluating the accuracy through leave-one-subject-out (L1O) method (where the training and test sets originate from different subjects) on the same dataset, it shows that the two-stage approach performs considerably better than a single-stage generalized activity classifier trained using all the sensor unit positions. Reference [60] also classifies the activity and the sensor unit position simultaneously, following a more complicated approach: For each time segment, it first determines the activity category as static or dynamic, without the position information. Then, it classifies the sensor position by using the classifier specifically trained for the determined category. Finally, it recognizes the activity type by relying on the classifier trained for that particular sensor unit position. The subjects are isolated in all three steps where all the classifiers are trained and tested separately for each subject. Hence, the method may not be generalizable to a new subject, considering that activity recognition rate highly depends on the subject(s) from whom the training data are acquired [17, 62].

1.1.1.5 Other Approaches

Reference [31] relies on a machine-learning approach that is robust to incorrect positioning of some of multiple sensor units. It fuses the decisions of the classifiers, each of which is trained specifically for a sensor unit, instead of the usual approach where a single classifier is trained by aggregating the features of all the units. This method can tolerate incorrect positioning of some of the sensor units by relying on the correctly placed ones in the classification process.

1.1.2 Invariance to Sensor Unit Orientation

A variety of methods have been proposed to achieve orientation invariance with wearable motion sensors. These methods can be grouped as transformation-based geometric methods, learning-based methods, and other approaches.

1.1.2.1 Transformation-Based Geometric Methods

A straightforward method for achieving orientation invariance is to calculate the magnitudes (the Euclidean norms) of the 3D vectors acquired by tri-axial sensors and to use these magnitudes as features in the classification process instead of individual vector components. When the sensor unit is placed at a different orientation, the magnitude of the sensor readings remains the same, making this method invariant to sensor unit orientation [26, 48, 63]. Reference [26] states that a significant amount of information is lost with this approach and the accuracy drops off even for classifying simple daily activities. Instead of using only the magnitude, references [47, 64, 65] append the magnitude of the tri-axial acceleration vector as a fourth axis to the tri-axial data. Reference [47] shows that this modification slightly increases the accuracy compared to using only the tri-axial acceleration components. Even if the magnitude of the acceleration is not appended to the data, the limited number of sensor unit orientations considered (only four) allows accurate classification to be achieved with SVM classifiers [47]. Reference [66] uses the magnitude, the y-axis data, and the squared sum of x and y axes of the tri-axial acceleration sequences acquired by a mobile phone, assuming that the orientation of the phone carried in a pocket has natural limitations: the screen of the phone either faces inward or outward.

In a number of studies [58, 67, 68], the direction of the gravity vector is estimated by averaging the acceleration vectors in the long term. This is based on the assumption that the acceleration component associated with daily activities averages out to zero, causing the gravity component to remain dominant. Then, the amplitude of the acceleration along the gravity vector direction and the magnitude of the acceleration perpendicular to that direction are used for activity recognition [58, 67, 68], which is equivalent to transforming tri-axial sensor sequences into bi-axial ones. In terms of activity recognition accuracy, in reference [67], this method is shown to perform slightly better and in reference [68], significantly worse than using only the magnitude of the acceleration vector.

In addition to the direction of the gravity vector, reference [52] also estimates the direction of the forward-backward (saggital) axis of the human body based on the assumption that most of the body movements as well as the variance of the acceleration sequences are in this direction. The sensor data are transformed into the body frame whose axes point in the direction of the gravity vector, the forward-backward direction of the body that is perpendicular to that, and a third direction perpendicular to both, forming a right-handed coordinate frame. The method in [52] does not distinguish between the forward and backward directions of the body, whereas reference [26] determines the forward direction from the sign of the integral of the acceleration as the subject walks.

Reference [69] proposes a coordinate transformation from the sensor frame to the Earth frame to achieve orientation invariance. To transform the data, the orientation of a mobile phone is estimated based on the data acquired from the accelerometer, gyroscope, and magnetometer of the sensor unit embedded in the device. An accuracy level close to the fixed orientation case is obtained by representing the sensor data with respect to the Earth frame. However, only two different orientations of the phone are considered, which is a major limitation of the study in [69]. Reference [70] calculates three principal axes based on acceleration and angular rate sequences by using Principal Component Analysis (PCA) and represents the sensor data with respect to these axes. Among the references [71–73] that employ deep learning for activity recognition, reference [73] increases robustness to variable sensor unit orientations by summing the features extracted from the x, y, z axes.

1.1.2.2 Learning-Based Methods

Reference [31] proposes a high-level machine-learning approach for activity recognition that can tolerate incorrect placement (both position and orientation) of *some of* multiple wearable sensor units. In the standard approach, features extracted from all the sensor units are aggregated and the activity is classified at once. In reference [31], the performed activity is classified by processing the data acquired from each sensor unit separately and the decisions are fused by using the confidence values. The proposed method is compared with the standard approach for different sets of activities, features, and different numbers of incorrectly placed sensor units by using three types of classifiers. When the subjects are requested to place the sensor units at any position and orientation on the appropriate body parts, incorrect placement of some of the units can be tolerated when all nine units are employed, but not with only a single unit. Adapting the class means in the feature space is proposed to achieve position invariance in [54] in addition to orientation invariance (see Section 1.1.1.3).

1.1.2.3 Other Approaches

Reference [74] proposes to classify the sensor unit orientation to compensate for variations in orientation. Dynamic portions of the sensor sequences are extracted by thresholding the standard deviation of the acceleration sequence and four pre-determined sensor unit orientations are perfectly recognized by a one-nearest-neighbor (1-NN) classifier. Then, the sensory data are rotated accordingly prior to activity recognition. However, the number of sensor unit orientations considered is again very limited and the direction of one of the sensor axes is common to all four orientations.

1.1.3 Simultaneous Invariance to Sensor Unit Position and Orientation

Among the studies on position invariance, references [39, 40, 53, 56, 58, 61] employ transformations that completely remove the orientation information. References [37, 38, 51] rely on initial calibration poses or movements to achieve orientation invariance throughout the recording session. Reference [54] claims to handle variations in both the position and the orientation by adapting the class means in the feature space. Reference [75] integrates the magnitude of the angular rate for position and orientation invariance within the same body part; however, it also uses the magnitude of the acceleration which is invariant only to sensor unit orientation. The classification schemes in [41, 47] are not fully orientation invariant but they include additional features to increase robustness to the sensor unit orientations. One of the three sensor axes is assumed to point either away from or towards the body in [50]. Datasets used in studies [52–54, 60] contain a set of pre-determined orientations by discretization. On the other hand, references [43, 44] do not specify how the mobile phones (whose motion sensors are employed) are oriented, and may include multiple orientations.

1.1.4 Discussion

Most of the existing methods are not comparable with each other because of the difference in the sensor types, sensor placement, activity and movement types, classification schemes, and the techniques used for evaluating the accuracy. Moreover, the impact of the proposed position and orientation invariance methods on the accuracy is not always presented because it is not possible to directly compare them with the fixed-position or fixed-orientation approaches in some scenarios; e.g., when no data are acquired with fixed sensor unit positions and/or orientations. The studies [33, 45–50, 56, 57, 59, 60] consider only a finite number of possible positions for the sensor units on the body, which is an unrealistic assumption. Some of the existing methods such as [26, 47, 52, 66, 67] either impose a major restriction on the possible sensor unit orientations or the types of body movements, which prevents them from being used in a wide range of applications such as health, state, and activity monitoring of elderly or disabled people.

Different activity or movement classes are considered in the previous studies, which highly affect the classification accuracy, as shown in [31]. For instance, some studies consider only one stationary activity (during which the subject is not moving) [43], combine several activity types into a single class [42, 44, 45, 47–50, 58], or do not include any [31, 33, 37, 51, 54, 55, 57, 61], as shown in Table 1.1. Some datasets consider the activities that are often poorly classified or confused with each other as a single class. For example, ascending and descending stairs are combined in [37, 41, 56], which expectedly has a positive effect on the accuracy, given that these activities are classified with lower accuracy than the others in [3, 26, 43, 44, 46, 47, 49]. Most of the existing studies do not utilize a magnetometer, which measures the Earth's magnetic field superposed with external magnetic sources (if any) and provides the orientation information.

1.2 Main Contributions of the Thesis

We develop transformation and classification techniques that are applicable to wearable motion sensor data to achieve robustness to the placement of the sensor units in terms of their position and orientation:

- In Chapter 2, we propose two different techniques for orientation invariance. They are based on geometrical transformations that remove the orientation information from the data while preserving the remaining information about the movements of the sensor unit. We mathematically prove the orientation-invariance property of the transformations without making any assumptions. They are computationally efficient and easy to implement, can be applied to different sensor types, and integrated into the pre-processing stage of many wearable sensing schemes.

- In Chapter 3, we develop a transformation technique as an alternative to those proposed in Chapter 2 and improve the classification accuracy while still preserving the orientation-invariance property. The transformation requires each sensor unit to contain an accelerometer, a gyroscope, and a magnetometer, each being tri-axial because it exploits the information acquired by these three sensor types to estimate the orientations of the units with respect to the Earth frame at each time sample. The transformation is sufficiently efficient to be implemented in near real time although its run times are longer than those in Chapter 2. It can be applied in the pre-processing stage of existing wearable systems, as those proposed in Chapter 2.

- In Chapter 4, we develop a novel non-iterative orientation estimation method (OEM) for motion sensor units. When it is integrated into the orientation-invariant transformation (OIT) that is proposed in Chapter 3, it improves the activity recognition accuracy compared to the existing methods, as well as being computationally efficient.

- In Chapter 5, we provide flexibility in the positioning of the sensor units in multiple ways: First, we propose transformation techniques to allow the units to be positioned anywhere within the same body part to improve the robustness to their attachment and also shifts in position and orientation that may occur in the long term. Secondly, we develop a transformation that makes the activity recognition system invariant to the interchanging of the sensor units so that the users do not need to identify them before putting them on their body. Finally, we perform activity recognition based on a single sensor unit where the dataset may contain multiple units that are placed at different positions on the body. We also achieve the position-invariance property simultaneously with the interchangeable units and also with the single-unit classification scheme.

- In Chapter 6, we simultaneously implement the position- and orientation-invariant techniques that are proposed in the previous chapters. We achieve activity recognition accuracies well above random decision making while allowing the sensor units to be placed arbitrarily on the body.

1.3 Organization of the Thesis

The rest of this thesis is organized as follows: In Chapters 2 and 3, we provide transformations to achieve orientation invariance of wearable motion sensor units. In Chapter 4, we propose a novel method to estimate the orientation of sensor units and integrate it into the transformation proposed in Chapter 3. Chapter 5 presents the techniques proposed for invariance to the positioning of the units, their interchangeability, and classification based on a single unit. Chapter 6 combines the position- and orientation-invariant techniques to simultaneously achieve position and orientation invariance. Finally, in Chapter 7, we provide concluding remarks and indicate directions for future research.

Chapter 2

Invariance to Sensor Unit Orientation Based on Geometrical Transformations

In this chapter, we focus on invariance to sensor unit orientation and propose to transform the 3D time-domain sensor data in a way that the resulting sequences do not depend on the absolute sensor orientation (but they *should* depend on the *changes* in the orientation over time to preserve activity-related rotational information). In other words, each 3D time-domain sensor sequence is transformed to another multi-dimensional time-domain sequence in an orientation-invariant manner, as depicted in Figure 2.1.

We propose two different OIT techniques, namely the *heuristic* OIT [18, 21] and the *singular value decomposition (SVD)-based* OIT [18, 22], described below. The content of this chapter has appeared in [18].

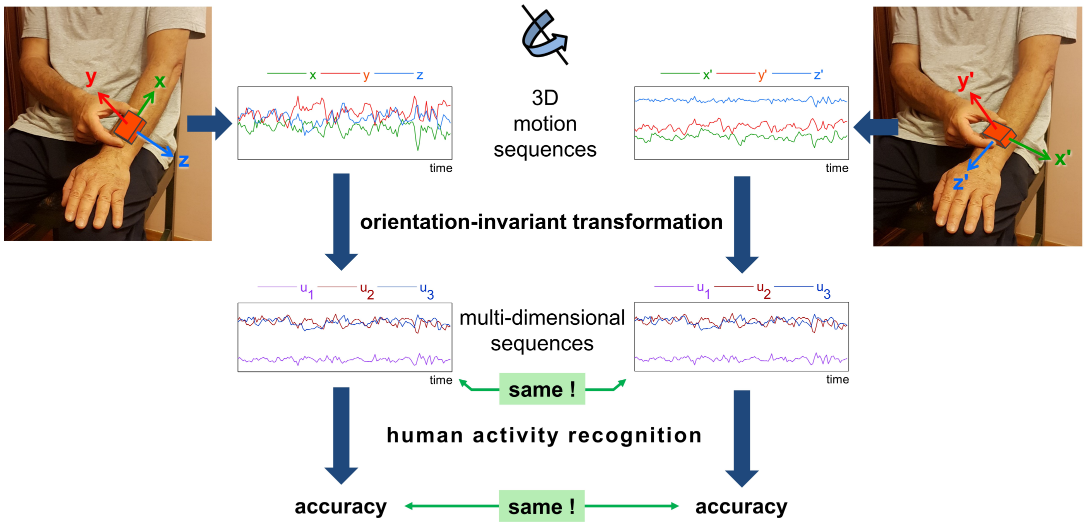


Figure 2.1: An overview of the proposed methodology for sensor unit orientation invariance.

2.1 Heuristic Orientation-Invariant Transformation

In the heuristic OIT, 3D sensor data are transformed into 9D data, invariant to sensor unit orientation. Let $\vec{v}_n = (v_x[n], v_y[n], v_z[n])^T$, $1 \leq n \leq N$ be the data vector in 3D space $\mathbb{R}^3$ acquired from the x, y, z axes of a tri-axial sensor, such as an accelerometer, at time sample n. The first- and second-order time-differences of $\vec{v}_n$ are defined as $\Delta\vec{v}_n = \vec{v}_{n+1} - \vec{v}_n$ and $\Delta\!\!\!\Delta\vec{v}_n = \Delta\vec{v}_{n+1} - \Delta\vec{v}_n$, respectively. The heuristic OIT, represented by a transformation $\mathcal{T}_{\text{heuristic}} : \vec{v}_n \rightarrow \vec{w}_n \;\; \forall n$, transforms the measurement vectors $\vec{v}_n \in \mathbb{R}^3$ to orientation-invariant vectors $\vec{w}_n \in \mathbb{R}^9$, whose elements are selected as follows:

$$w_1[n] = \|\vec{v}_n\| \qquad \text{(the norm)} \tag{2.1a}$$

$$w_2[n] = \|\Delta\vec{v}_n\| \qquad \text{(the norm of the first-order difference } \Delta\vec{v}_n\text{)} \tag{2.1b}$$

$$w_3[n] = \|\Delta\!\!\!\Delta\vec{v}_n\| \qquad \text{(the norm of the second-order difference } \Delta\!\!\!\Delta\vec{v}_n\text{)} \tag{2.1c}$$

$$w_4[n] = \alpha_n = \angle\left(\vec{v}_n, \vec{v}_{n+1}\right) \qquad \text{(the angle between } \vec{v}_n \text{ and } \vec{v}_{n+1}\text{)} \tag{2.1d}$$

$$w_5[n] = \beta_n = \angle\left(\Delta\vec{v}_n, \Delta\vec{v}_{n+1}\right) \qquad \text{(the angle between } \Delta\vec{v}_n \text{ and } \Delta\vec{v}_{n+1}\text{)} \tag{2.1e}$$

$$w_6[n] = \gamma_n = \angle\left(\boldsymbol{\Delta}\vec{v}_n, \boldsymbol{\Delta}\vec{v}_{n+1}\right) \quad \text{(the angle between } \boldsymbol{\Delta}\vec{v}_n \text{ and } \boldsymbol{\Delta}\vec{v}_{n+1}\text{)} \tag{2.1f}$$

$$w_7[n] = \theta_n = \angle\left(\vec{p}_n, \vec{p}_{n+1}\right) \text{ where } \vec{p}_n = \vec{v}_n \times \vec{v}_{n+1} \tag{2.1g}$$
$$\text{(the angle between rotation axes } \vec{p}_n \text{ and } \vec{p}_{n+1}\text{)}$$

$$w_8[n] = \phi_n = \angle\left(\vec{q}_n, \vec{q}_{n+1}\right) \text{ where } \vec{q}_n = \Delta\vec{v}_n \times \Delta\vec{v}_{n+1} \tag{2.1h}$$
$$\text{(the angle between rotation axes } \vec{q}_n \text{ and } \vec{q}_{n+1}\text{)}$$

$$w_9[n] = \psi_n = \angle\left(\vec{r}_n, \vec{r}_{n+1}\right) \text{ where } \vec{r}_n = \boldsymbol{\Delta}\vec{v}_n \times \boldsymbol{\Delta}\vec{v}_{n+1} \tag{2.1i}$$
$$\text{(the angle between rotation axes } \vec{r}_n \text{ and } \vec{r}_{n+1}\text{)}$$

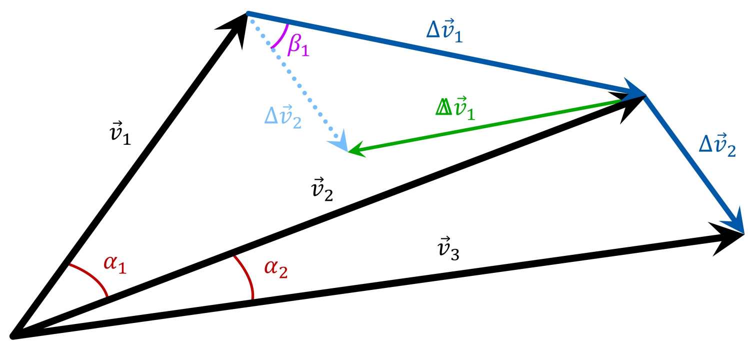
The rationale for selecting these nine elements among many is that apart from the norms covered by the first three elements, the angles between the successive time samples of the sensor sequence and its first- and second-order differences (fourth to sixth elements) contain more granularity and fine detail regarding the activities performed. The last three elements consider rotation axes between successive time samples and contain information about the rotational movements of the data vectors in 3D space.

The first five elements are shown geometrically in Figure 2.2(a). In Equation (2.1) and throughout this thesis, $\|\cdot\|$ denotes the Euclidean norm. In Equation (2.1d), the angle α_n between $\vec{v}_n$ and $\vec{v}_{n+1}$ is calculated based on the two vectors' normalized inner product:

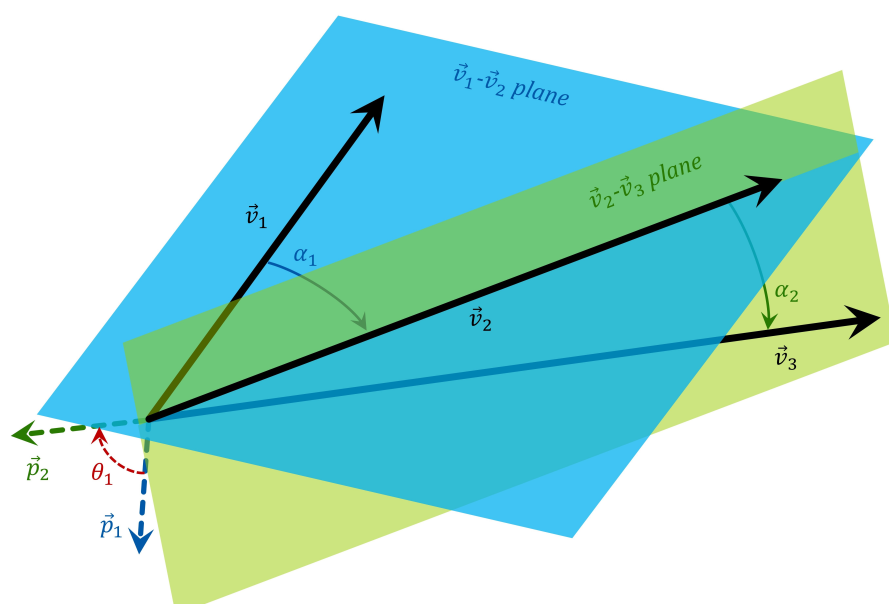
$$\alpha_n = \angle\left(\vec{v}_n, \vec{v}_{n+1}\right) = \cos^{-1}\left(\frac{\vec{v}_n \cdot \vec{v}_{n+1}}{\|\vec{v}_n\|\,\|\vec{v}_{n+1}\|}\right) \tag{2.2}$$

The angle α_n is set to zero when $\vec{v}_n = \vec{0}$ and/or $\vec{v}_{n+1} = \vec{0}$, in which case it is not defined. The angles in Equation (2.1e–i) are calculated in the same way.

In Equation (2.1g), $\vec{p}_n$ is the vector representing the axis of rotation from $\vec{v}_n$ to $\vec{v}_{n+1}$; that is, $\vec{v}_{n+1}$ is obtained when $\vec{v}_n$ is rotated about $\vec{p}_n$ by an angle of α_n (see Equation (2.1d) and Figure 2.2(b)). Similarly, $\vec{v}_{n+2}$ is obtained when $\vec{v}_{n+1}$ is rotated about $\vec{p}_{n+1}$ by α_{n+1}. Then, the angle between the consecutive rotation axes, $\vec{p}_n$ and $\vec{p}_{n+1}$, is calculated, which is denoted by θ_n, as shown in Figure 2.2(b). In Equation (2.1h,i), the rotation axes are calculated based on the first- and



(a)



(b)

Figure 2.2: Graphical illustration of the selected axes of the heuristic OIT. The geometric features of three sequential measurements $\vec{v}_1, \vec{v}_2, \vec{v}_3$ in 3D space are shown. The first- and second-order difference sequences, the angles between successive measurement vectors, and the angles between successive difference vectors are shown in (a); The rotation axes and the angles between them are illustrated in (b).

second-order difference sequences $\Delta\vec{v}_n$ and $\Delta\!\!\!\Delta\vec{v}_n$, respectively, and the angle between the consecutive rotation axes is calculated.[1]

The transformed vector $\vec{w}_n$ has nine elements, corresponding to the new axes that are completely invariant to sensor orientation. Mathematically, when $\vec{v}_n$ is pre- or post-multiplied by any rotation matrix for all n, the transformed vector $\vec{w}_n$ remains unchanged. Note that for this transformation to be orientation invariant, the measured sequence $\vec{v}_n$ needs to be multiplied by the *same* rotation matrix for all n; that is, the sensor can be placed at any orientation at some given position on the body, but its orientation with respect to the body must remain the same during the short time period over which data are processed. This is a necessary restriction because we preserve the *change* in the orientation of measurement vectors $\vec{v}_n$ in the transformation over time, which provides information about the orientation change of the body if the sensor rotates with the body rather than rotating freely.

To prove the orientation invariance of the transformation $\mathcal{T}_{\text{heuristic}}$ mathematically, assume that the sensor is placed at a different orientation and the acquired data are $\vec{v}_n{}' = \mathbf{R}\vec{v}_n \;\; \forall n$, where $\mathbf{R}$ is a rotation matrix that is constant over n. Then, we need to prove that its transformation $\vec{w}_n{}'$ is the same as $\vec{w}_n$:

$$\vec{w}_n = \vec{w}_n{}' \;\; \forall n \quad \text{where} \quad \vec{v}_n \xrightarrow{\mathcal{T}_{\text{heuristic}}} \vec{w}_n \quad \text{and} \quad \vec{v}_n{}' \xrightarrow{\mathcal{T}_{\text{heuristic}}} \vec{w}_n{}' \tag{2.3}$$

For the proof, note the following facts: (1) multiplying a vector by a rotation matrix does not change its norm; (2) multiplying two vectors by the same rotation matrix affects neither the angle between them nor their inner product;[2] and (3) if a time-varying vector is multiplied by a constant rotation matrix over time, its first-

[1] $\vec{p}_n$, $\vec{q}_n$, and $\vec{r}_n$ need not have unit norms because only their directions are used in Equation (2.1g–i).

[2] For the proof, let $\alpha_n = \angle(\vec{v}_n, \vec{v}_{n+1})$. Then,

$$\angle(\mathbf{R}\vec{v}_n, \mathbf{R}\vec{v}_{n+1}) = \cos^{-1}\left(\frac{(\mathbf{R}\vec{v}_n)\cdot(\mathbf{R}\vec{v}_{n+1})}{\|\mathbf{R}\vec{v}_n\|\,\|\mathbf{R}\vec{v}_{n+1}\|}\right) = \cos^{-1}\left(\frac{\vec{v}_n\cdot\vec{v}_{n+1}}{\|\vec{v}_n\|\,\|\vec{v}_{n+1}\|}\right) = \alpha_n$$

for any rotation matrix $\mathbf{R}$.

and second-order differences are also multiplied by the same rotation matrix.[3] Using these facts, we prove Equation (2.3) for the first six dimensions of the heuristic OIT:

$$\begin{aligned}
w'_1[n] &= \|\mathbf{R}\vec{v}_n\| = \|\vec{v}_n\| = w_1[n] \\
w'_2[n] &= \|\Delta(\mathbf{R}\vec{v}_n)\| = \|\mathbf{R}\,\Delta\vec{v}_n\| = \|\Delta\vec{v}_n\| = w_2[n] \\
w'_3[n] &= \|\mathbb{\Delta}(\mathbf{R}\vec{v}_n)\| = \|\mathbf{R}\,\mathbb{\Delta}\vec{v}_n\| = \|\mathbb{\Delta}\vec{v}_n\| = w_3[n] \\
w'_4[n] &= \angle\left(\mathbf{R}\vec{v}_n, \mathbf{R}\vec{v}_{n+1}\right) = \angle\left(\vec{v}_n, \vec{v}_{n+1}\right) = w_4[n] \\
w'_5[n] &= \angle\left(\Delta(\mathbf{R}\vec{v}_n)\,, \Delta(\mathbf{R}\vec{v}_{n+1})\right) = \angle\left(\mathbf{R}\,\Delta\vec{v}_n, \mathbf{R}\,\Delta\vec{v}_{n+1}\right) = \angle\left(\Delta\vec{v}_n, \Delta\vec{v}_{n+1}\right) \\
&= w_5[n] \\
w'_6[n] &= \angle\left(\mathbb{\Delta}(\mathbf{R}\vec{v}_n)\,, \mathbb{\Delta}(\mathbf{R}\vec{v}_{n+1})\right) = \angle\left(\mathbf{R}\,\mathbb{\Delta}\vec{v}_n, \mathbf{R}\,\mathbb{\Delta}\vec{v}_{n+1}\right) = \angle\left(\mathbb{\Delta}\vec{v}_n, \mathbb{\Delta}\vec{v}_{n+1}\right) \\
&= w_6[n]
\end{aligned} \tag{2.5}$$

For the remaining axes, note that if any two vectors are multiplied by the same rotation matrix, the rotation axis between them also rotates in the same way. To prove this, let $\vec{p}_n{}' = \vec{v}_n{}' \times \vec{v}_{n+1}{}'$ be the rotation axis between $\vec{v}_n{}'$ and $\vec{v}_{n+1}{}'$. Then,

$$\vec{p}_n{}' = \vec{v}_n{}' \times \vec{v}_{n+1}{}' = (\mathbf{R}\vec{v}_n) \times (\mathbf{R}\vec{v}_{n+1}) = \mathbf{R}\left(\vec{v}_n \times \vec{v}_{n+1}\right) = \mathbf{R}\vec{p}_n \tag{2.6}$$

The rotation axes $\vec{q}_n$ and $\vec{r}_n$ also rotate in the same way as $\vec{v}_n$ rotates. Based on these observations, we prove Equation (2.3) for the remaining dimensions:

$$\begin{aligned}
w'_7[n] &= \angle\left(\vec{p}_n{}', \vec{p}_{n+1}{}'\right) = \angle\left(\mathbf{R}\vec{p}_n, \mathbf{R}\vec{p}_{n+1}\right) = \angle\left(\vec{p}_n, \vec{p}_{n+1}\right) = w_7[n] \\
w'_8[n] &= \angle\left(\vec{q}_n{}', \vec{q}_{n+1}{}'\right) = \angle\left(\mathbf{R}\vec{q}_n, \mathbf{R}\vec{q}_{n+1}\right) = \angle\left(\vec{q}_n, \vec{q}_{n+1}\right) = w_8[n] \\
w'_9[n] &= \angle\left(\vec{r}_n{}', \vec{r}_{n+1}{}'\right) = \angle\left(\mathbf{R}\vec{r}_n, \mathbf{R}\vec{r}_{n+1}\right) = \angle\left(\vec{r}_n, \vec{r}_{n+1}\right) = w_9[n]
\end{aligned} \tag{2.7}$$

Therefore, the orientation invariance of the heuristic OIT is proven.

[3] For the proof, let $\Delta\vec{v}_n = \vec{v}_{n+1} - \vec{v}_n$ and $\mathbb{\Delta}\vec{v}_n = \Delta\vec{v}_{n+1} - \Delta\vec{v}_n$. Then,

$$\Delta(\mathbf{R}\vec{v}_n) = \mathbf{R}\vec{v}_{n+1} - \mathbf{R}\vec{v}_n = \mathbf{R}\,\Delta\vec{v}_n$$

and

$$\mathbb{\Delta}(\mathbf{R}\vec{v}_n) = \Delta(\mathbf{R}\vec{v}_{n+1}) - \Delta(\mathbf{R}\vec{v}_n) = \mathbf{R}\Delta\vec{v}_{n+1} - \mathbf{R}\Delta\vec{v}_n = \mathbf{R}\,\mathbb{\Delta}\vec{v}_n$$

for any rotation matrix $\mathbf{R}$.

2.2 Orientation-Invariant Transformation Based on Singular Value Decomposition

As an alternative to the heuristic approach, orientation invariance can be achieved by singular value decomposition [76]. In the SVD approach, the x, y, z axes of the original tri-axial sensor are transformed to three principal axes that are orthogonal to each other and along which the variance of the data is the largest. The directions of the principal axes, hence the transformation, depends on the data to be transformed. The motivation for using SVD to achieve orientation invariance is that when the data constellation is rotated as a whole, the principal axes also rotate in the same way, and the representation of the data in terms of the principal axes remains the same.

To apply SVD, data acquired from each tri-axial sensor are represented as a matrix $\mathbf{V}$ of size $3 \times N$, with the rows corresponding to the x, y, z axes and the columns representing the time samples:

$$\mathbf{V} = \begin{bmatrix} \vec{v}_1 & \vec{v}_2 & \cdots & \vec{v}_N \end{bmatrix} \tag{2.8}$$

Then, $\mathbf{V}$ is decomposed into three matrices by SVD as

$$\mathbf{V} = \mathbf{U}\mathbf{\Sigma}\mathbf{W}^{\mathrm{T}} \tag{2.9}$$

In general, for complex $\mathbf{V}$, $\mathbf{U}$ is a 3×3 unitary matrix, $\mathbf{\Sigma}$ is a $3 \times N$ rectangular diagonal matrix containing the singular values along the diagonal, and $\mathbf{W}$ is an $N \times N$ unitary matrix. In our application, $\mathbf{V}$ is real, so $\mathbf{U}$ and $\mathbf{W}$ are real unitary, hence, orthonormal matrices that satisfy $\mathbf{U}^{\mathrm{T}}\mathbf{U} = \mathbf{U}\mathbf{U}^{\mathrm{T}} = \mathbf{I}_{3\times3}$ and $\mathbf{W}^{\mathrm{T}}\mathbf{W} = \mathbf{W}\mathbf{W}^{\mathrm{T}} = \mathbf{I}_{N\times N}$, where $\mathbf{I}$ is the identity matrix. The matrix $\mathbf{U}$ can also be viewed as a 3×3 rotation matrix.

Since the matrix $\mathbf{V}$ only has three rows, its rank is at most three, and only the first three singular values can be non-zero. Hence, SVD can be represented more compactly by considering only the first three columns of $\mathbf{\Sigma}$ and $\mathbf{W}$, in which case their sizes become 3×3 and $N \times 3$, respectively. This compact representation will be used in the rest of the thesis, where $\mathbf{W}$ is no longer unitary because it is not square, but has orthonormal columns that satisfy $\mathbf{W}^{\mathrm{T}}\mathbf{W} = \mathbf{I}_{3\times 3}$.

Changing the orientation of a sensor unit is equivalent to rotating the measurement vectors for each time sample in the same way; that is, pre-multiplying $\mathbf{V}$ by a rotation matrix $\mathbf{R}$:

$$\tilde{\mathbf{V}} = \mathbf{R}\mathbf{V} \tag{2.10}$$

$\mathbf{V}$ is constant over time because it is assumed that the sensor orientation with respect to the body part onto which the sensor is placed remains the same while acquiring the data stored in $\mathbf{V}$, as done in the heuristic OIT. The SVD of the rotated data matrix $\tilde{\mathbf{V}}$ becomes

$$\tilde{\mathbf{V}} = \mathbf{R}\left(\mathbf{U}\boldsymbol{\Sigma}\mathbf{W}^{\mathrm{T}}\right) = \left(\mathbf{R}\mathbf{U}\right)\boldsymbol{\Sigma}\mathbf{W}^{\mathrm{T}} = \tilde{\mathbf{U}}\boldsymbol{\Sigma}\mathbf{W}^{\mathrm{T}} \tag{2.11}$$

where $\tilde{\mathbf{U}} = \mathbf{R}\mathbf{U}$ because the product of two rotation matrices is another rotation matrix, and the SVD representation is *almost unique* [77] up to the signs of the columns of $\mathbf{U}$ and $\mathbf{W}$. In other words, if a principal vector $\vec{u}_i$ (the ith column of $\mathbf{U}$, where $i = 1, 2, 3$) is selected in the opposite direction, the variance along that axis is still maximized and the decomposition can be preserved by negating the corresponding column of $\mathbf{W}$. (Another ambiguity in SVD is that the principal vectors can be selected in any direction in case of degenerateness, that is, when $\mathbf{V}$ is not full-rank. This situation is not observed in experimental data because of the presence of noise.)

Because of the almost-uniqueness property of SVD, the matrices $\boldsymbol{\Sigma}$ and $\mathbf{W}$ are not affected by the sensor orientation (up to the signs of the columns of $\mathbf{W}$). Therefore, the proposed SVD-based OIT omits the leftmost matrix and takes $\boldsymbol{\Sigma}\mathbf{W}^{\mathrm{T}}$ as the part of the data that is invariant to sensor orientation (up to the signs of the resulting axes). Then, the SVD-based OIT can be represented as

$$\mathcal{T}_{\mathrm{SVD}} : \mathbf{V} \rightarrow \boldsymbol{\Sigma}\mathbf{W}^{\mathrm{T}} \tag{2.12}$$

This transformation is equivalent to a rotational transformation because

$$\mathbf{\Sigma W}^{\mathrm{T}} = \left(\mathbf{U}^{\mathrm{T}}\mathbf{U}\right)\mathbf{\Sigma W}^{\mathrm{T}} = \mathbf{U}^{\mathrm{T}}\left(\mathbf{U\Sigma W}^{\mathrm{T}}\right) = \mathbf{U}^{\mathrm{T}}\mathbf{V} \tag{2.13}$$

and $\mathbf{U}^{\mathrm{T}}$ is the corresponding rotation matrix. Note that the rotation may be right- or left-handed, that is, proper or improper because $\det \mathbf{U} = \pm 1$.

The SVD-based OIT rotates the measurement vectors in 3D space such that the variance of the data along the first principal axis $\vec{u}_1$ is the largest, followed by the second principal axis $\vec{u}_2$, which is orthogonal to $\vec{u}_1$, and followed by the third axis $\vec{u}_3$, which is orthogonal to both $\vec{u}_1$ and $\vec{u}_2$. Thus, if all the vectors are rotated in the same way because of a different sensor orientation, the rotation $\mathbf{U}^{\mathrm{T}}$ will change accordingly to yield the same transformed sequence (up to the signs of the axes). Mathematically, if the data matrix is rotated as in Equation (2.10), the same transformed data, $\mathbf{\Sigma W}^{\mathrm{T}}$, must be obtained (except for the signs of the rows). Hence, using the fact that $\mathbf{RU}$ is also a rotation matrix composed of two rotations, one can write

$$\mathbf{\Sigma W}^{\mathrm{T}} = \left[(\mathbf{RU})^{\mathrm{T}}(\mathbf{RU})\right]\mathbf{\Sigma W}^{\mathrm{T}} = (\mathbf{RU})^{\mathrm{T}}\left[(\mathbf{RU})\mathbf{\Sigma W}^{\mathrm{T}}\right] = (\mathbf{RU})^{\mathrm{T}}\tilde{\mathbf{V}} \tag{2.14}$$

which reveals that the new rotation matrix of the transformation is $(\mathbf{RU})^{\mathrm{T}}$.

If the unit contains more than one type of sensor (e.g., an accelerometer and a gyroscope), all the sensors have the same orientation with respect to the body part the sensor unit is placed on, ignoring the misalignment errors between the sensors in the same unit [78, 79]. In this case, the same rotational transformation should be applied to the data acquired by all the sensor types in the same unit. Let $\mathbf{V}_1, \mathbf{V}_2, \ldots, \mathbf{V}_S$ be the data matrices of sensors 1–S, defined as in Equation (2.8). These are concatenated as $\begin{bmatrix}\mathbf{V}_1 & \mathbf{V}_2 & \cdots & \mathbf{V}_S\end{bmatrix}$ to obtain a joint transformation, as illustrated in Figure 2.3(a) for the first dataset (dataset A in Section 2.3.1). In the figure, sequences of the three sensor types, namely the accelerometer, gyroscope, and magnetometer, are concatenated along the time-sample dimension. Gyroscope sequences have the smallest variance and accelerometer sequences have the largest. However, the more the data of a sensor type vary, the more the SVD transformation is affected, and that sensor type will have a greater contribution. Hence, we normalize the data of the different sensor types to equalize their effect on the transformation: In each dataset, we scale the sequences of each sensor type to have unit variance over the whole dataset. Then, we concatenate the normalized sequences (indicated by an overbar) as $\overline{\mathbf{V}} = \begin{bmatrix} \overline{\mathbf{V}}_1 & \overline{\mathbf{V}}_2 & \cdots & \overline{\mathbf{V}}_S \end{bmatrix}$ and use it in place of $\mathbf{V}$ in Equations (2.9)–(2.13). The normalized sequences are shown in Figure 2.3(b). Finally, we apply the SVD-based OIT, where a single 3×3 rotational transformation is employed for the same segment of all the sensor sequences acquired from the same sensor unit.

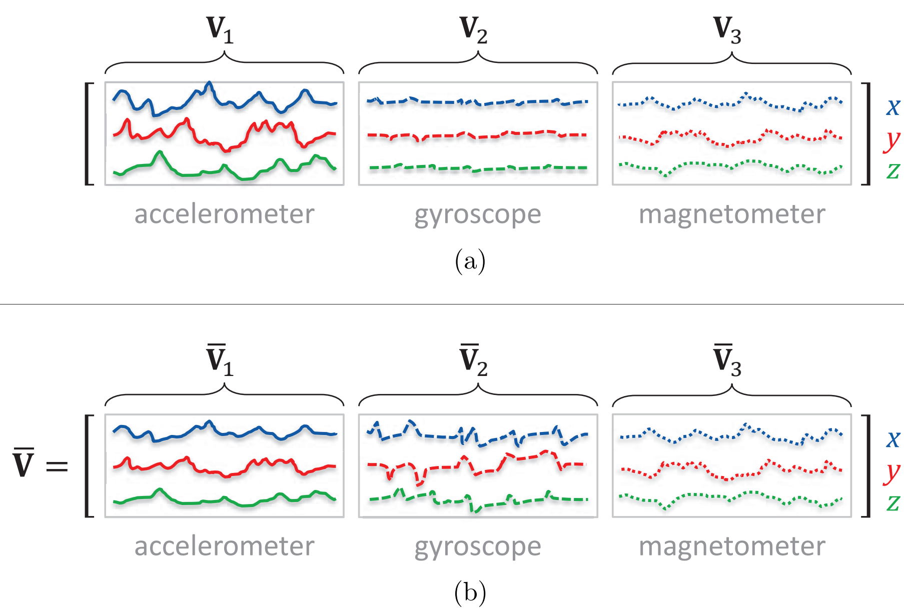


Figure 2.3: Concatenation of the sequences of the different sensor types. (a) Accelerometer, gyroscope, and magnetometer sequences are concatenated along the time-sample dimension to obtain a joint 3×3 transformation; and (b) the three sequences are normalized to have unit variance (over the whole dataset) before applying SVD-based OIT.

As an example, the 3D sequence of the accelerometer on the left leg of the first subject as he performs the tenth activity (A_{10}) in our activity dataset (dataset A in Section 2.3.1) is plotted in Figure 2.4(a). The sequence is rotated arbitrarily in 3D space and plotted in Figure 2.4(b).[4] To obtain orientation-invariant sequences, the

[4] For this specific example, the rotation matrix is calculated using Equation 3.6 with the angles $\theta = 12.9°$, $\phi = -54.3°$, and $\psi = -23.8°$.

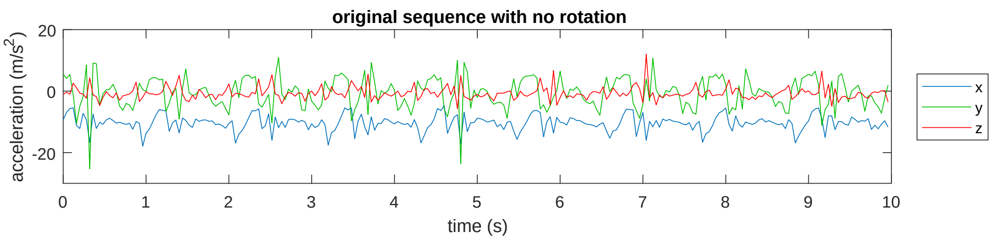
original sequence (or, equivalently, the rotated sequence) is transformed by the heuristic OIT (Figure 2.4(c)) and the SVD-based OIT (Figure 2.4(d)). Note that the sequences in Figure 2.4 (c) and (d) can be obtained by transforming either the original sequence in Figure 2.4(a) or its rotated form in Figure 2.4(b), or by any other arbitrarily rotated form of Figure 2.4(a). It is observed that the quasi-periodic nature of the data is preserved in both transformations. Since we observe in Figure 2.4(c) that the last two elements of the sequence transformed by the heuristic OIT contains much noise, we did not consider including differences of the sensor sequences beyond second order.

2.3 Methodology and Results

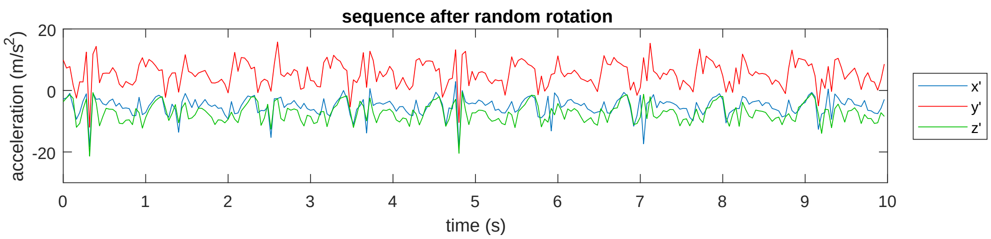
2.3.1 Datasets

We use five publicly available datasets recorded by different research groups to observe the effects of the proposed transformations on the acquired data. The datasets are labeled A–E and their attributes are provided in Table 2.1 [80–85]. The sensor configurations for the datasets are shown in Figure 2.5.

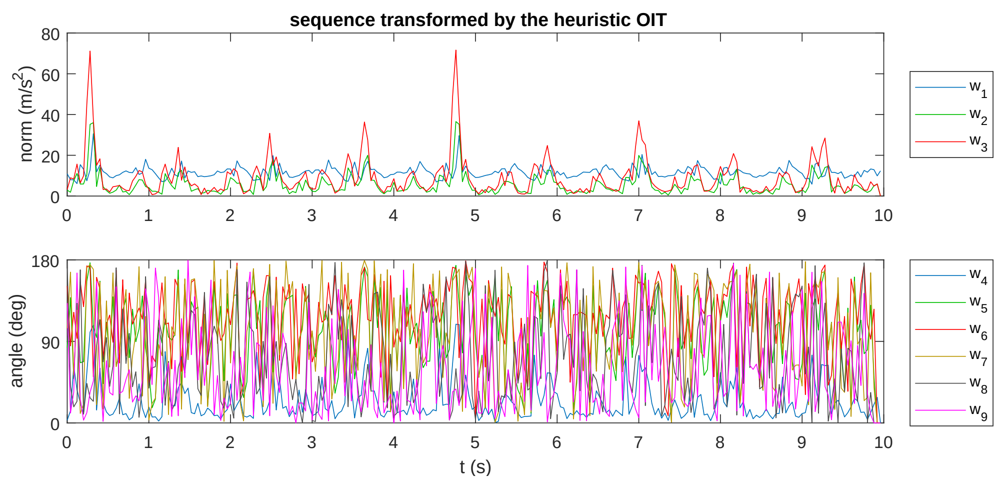
Dataset A was acquired by our research group [80, 81, 86, 87] using five Xsens MTx wearable sensor units containing tri-axial accelerometers, gyroscopes, and magnetometers [88]. Nineteen activities were considered, including random activities such as playing basketball (see Table 2.1 for the list of activities in the datasets). Among the five datasets, A is the largest, including a wide range of activities and employing a small network of five sensor units. Unlike in the other four datasets, in dataset A, each subject performs each activity for an equal amount of time. Dataset A is accessible through University of California Machine Learning Repository [80] and IEEE Data Port [81]. Dataset B utilizes four accelerometers and considers five basic activities, some of which are transitional activities, such as sitting down [82, 89]. However, this property is not used in the classification process. Dataset C considers six basic activities and utilizes a smartphone containing an



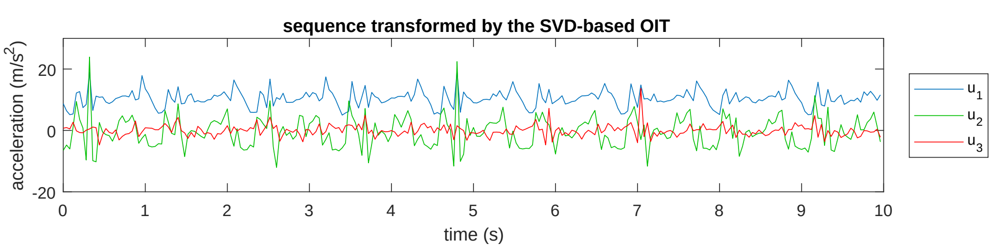
(a)



(b)



(c)



(d)

Figure 2.4: Original and orientation-invariant sensor sequences. (a) Original and (b) randomly rotated accelerometer sequences while performing A_{10} in dataset A. Orientation-invariant sequences transformed by the (c) heuristic and (d) SVD-based OIT.

Table 2.1: Attributes of the five datasets.

dataset	**A** [80,81]	**B** [82]	**C** [83]	**D** [84]	**E** [85]
no. of subjects	8	4	30	14	15
no. of activities	19	5	6	12	7
activities	sitting (A_1), standing (A_2), lying on back and on right side (A_3, A_4), ascending and descending stairs (A_5, A_6), standing still in an elevator (A_7), moving around in an elevator (A_8), walking in a parking lot (A_9), walking on a treadmill in flat and 15° inclined positions at a speed of 4 km/h (A_{10}, A_{11}), running on a treadmill at a speed of 8 km/h (A_{12}), exercising on a stepper (A_{13}), exercising on a cross trainer (A_{14}), cycling on an exercise bike in horizontal and vertical positions (A_{15}, A_{16}), rowing (A_{17}), jumping (A_{18}), and playing basketball (A_{19})	sitting down (B_1), standing up (B_2), standing (B_3), walking (B_4), and sitting (B_5)	walking (C_1), ascending stairs (C_2), descending stairs (C_3), sitting (C_4), standing (C_5), and lying (C_6)	walking (D_1), walking left and right (D_2 and D_3), ascending and descending stairs (D_4, D_5), running forward (D_6), jumping (D_7), sitting (D_8), standing (D_9), sleeping (D_{10}), ascending and descending in an elevator (D_{11}, D_{12})	working at a computer (E_1), standing up–walking–ascending/descending stairs (E_2), standing (E_3), walking (E_4), ascending/descending stairs (E_5), walking and talking with someone (E_6), talking while standing (E_7)
no. of non-stationary activities	15 A_5–A_{19}	3 B_1, B_2, B_4	3 C_1–C_3	9 D_1–D_7, D_{11}, D_{12}	4 E_2, E_4–E_6
no. of units	5	4	1	1	1
no. of axes per unit	9	3	6	6	3
unit positions	torso, right and left arm, right and left leg	waist, left thigh, right ankle, right upper arm	waist	front right hip	chest
sensor types	accelerometer, gyroscope, magnetometer	accelerometer	accelerometer, gyroscope (of smartphone)	acceleromter, gyroscope	accelerometer
dataset duration (h)	13	8	7	7	10
sampling rate (Hz)	25	8	50	100	52
no. of segments	9120	4130	10,299 (50% overlap)	5353	7345
segment length (s)	5	5	2.56	5	5
no. of features (for the reference case, with no transformation)	1170	276	234	156	78

accelerometer and a gyroscope [83, 90]. Using a high-pass filter, the gravitational component of the total acceleration is removed and an additional 3D sequence is obtained. This dataset has the largest number of subjects among the five datasets. Dataset D includes 12 activities and utilizes a single sensor unit containing an accelerometer and a gyroscope [84]. Unlike in the other four datasets, the subjects have a diverse range of age, height, and weight attributes. Dataset E utilizes a single tri-axial accelerometer placed on the chest [85, 91]. Most of the 15 subjects are male. Seven activities are considered, some of which are compound activities that contain more than one activity; for example, one of the activities comprises standing up, walking, and going up/down stairs. Some activity pairs seem to be difficult to distinguish, such as "walking" versus "walking and talking with someone." This dataset contains the smallest number of features per segment because only a single tri-axial sensor is used.

2.3.2 Activity Recognition

In activity recognition, a procedure similar to that in [86, 87, 92] is followed, whose block diagram is provided in Figure 2.6. In the pre-processing stage, the following steps are taken in order: the data sequences are segmented into time windows of fixed duration, one of the two OIT methods is applied if orientation invariance is desired, features are extracted from each segment and normalized. Then, the number of features is reduced through principal component analysis (PCA). Finally, classification is performed with four different classifiers and their accuracy is calculated using two cross-validation techniques.

2.3.2.1 Pre-Proccessing

First, the recorded data sequences are divided into non-overlapping segments of five seconds' duration each for datasets A, B, D, and E. Dataset C is originally divided into 50% overlapping segments of 2.56 s duration each and the original segments are used for this dataset. For all datasets, each segment belongs to a continuous

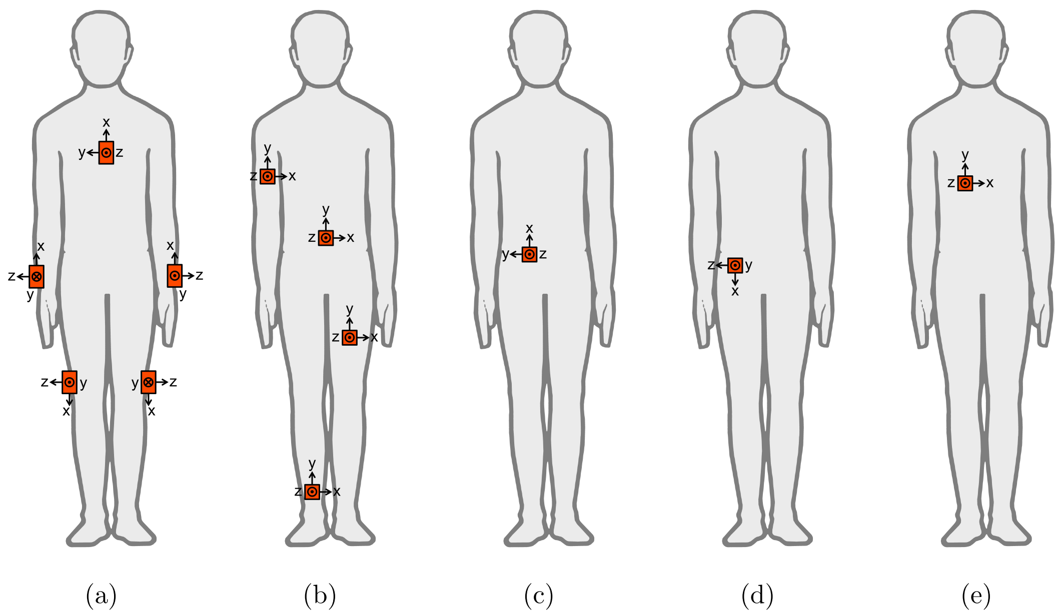


Figure 2.5: (a–e) Configuration of the sensor units in datasets A–E. The body drawing in the figure is from `http://www.clker.com/clipart-male-figure-outline.html` onto which sensor units were added by the authors.

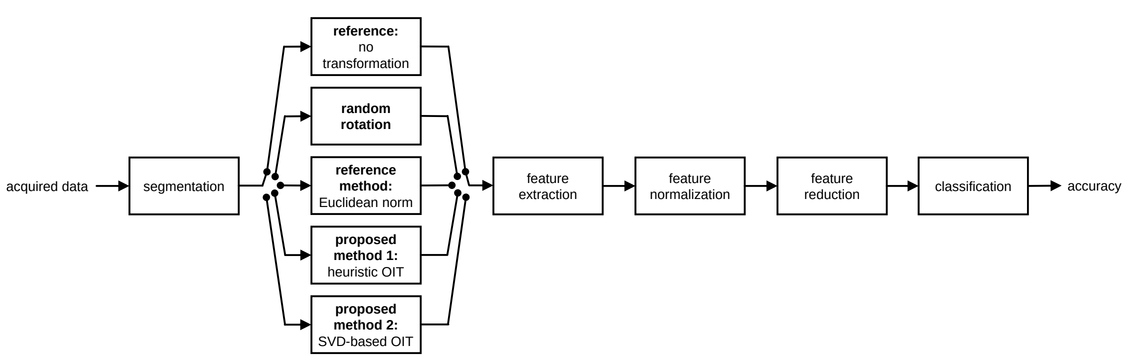


Figure 2.6: Activity recognition paradigm.

recording of a single activity performed by one of the subjects. The number of segments extracted from datasets A–E are 9120, 4130, 10,299, 5353, and 7345, respectively.

Following segmentation, one of the two proposed OIT techniques is applied to each segment of the data if orientation invariance is desired. Five cases are considered to observe the effects of sensor rotation on the classification process and to observe the improvement obtained with the proposed transformations:

- **Reference case** is the standard (ordinary) activity recognition scheme with fixed sensor unit positions and orientations. In this case, originally recorded sequences are used without applying any transformation.

- **Random rotation case** simulates the situation where each sensor unit is placed at a fixed position at any orientation. We use the original dataset by synthetically rotating the data to make a fair comparison between reference and random rotation cases. Tri-axial recordings of each sensor unit in each segment are randomly rotated in 3D space to observe the performance of the system when the units are placed at random orientations. To this end, for each segment of each unit of a given dataset, we generate a random rotation matrix $\mathbf{R}$ and pre-multiply each of the three-element measurement vectors belonging to that segment (for the accelerometer, gyroscope, and magnetometer if available) by this rotation matrix as $\tilde{\mathbf{V}} = \mathbf{R}\mathbf{V}$. The rotation matrix is calculated from yaw, pitch, and roll angles θ, ϕ, ψ that are randomly generated in the interval $[0, 2\pi)$ radians:

$$\mathbf{R} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\phi & 0 & \sin\phi \\ 0 & 1 & 0 \\ -\sin\phi & 0 & \cos\phi \end{bmatrix} \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.15)$$

  Note that while all of the sensor types in the same unit are rotated in the same way for a given segment, each segment recorded from each sensor unit for each dataset is rotated differently (by a different random rotation matrix).

- **Euclidean norm method** takes the Euclidean norm of each 3D sensor sequence at each time sample, and uses only the norms (as functions of the time sample) in classification. This is indeed a basic but proper OIT technique, which corresponds to the first dimension of the transformed signal, $w_1[n]$, in the heuristic OIT. It has been used in some studies to obtain a scalar quantity as a feature [93], to achieve orientation invariance in the simplest possible way [47, 94], or to incorporate additional information such as the energy expenditure estimate of the subject [95]. Taking the Euclidean norm reduces the number of axes by a factor of three.

- **Proposed method 1** corresponds to the heuristic OIT technique. The time-domain sequence contained in each segment of each tri-axial sensor in each unit is transformed to yield a 9D orientation-invariant time-domain sequence. As a consequence, dimensionality of the time-domain data increases by a factor of three (from three to nine). We also consider taking only the first three or the first six elements of the transformation. (Throughout this chapter, all of the nine elements of the heuristic OIT are considered unless stated otherwise.)

- **Proposed method 2** corresponds to the SVD-based OIT. A single transformation is calculated for all the sensor types in each sensor unit, again independently for each time segment, as explained in Section 2.2. The dimensionality is not affected by this transformation, unlike the Euclidean norm method and proposed method 1.

Although the sensor units are placed on the body at the same orientation during data acquisition, the applied transformations in the last three cases remove the orientation information from the data, simulating the case where each sensor unit is placed at any orientation on the body at a fixed position. Thus, a fair comparison can be made among the five cases based on the same experimental data.

For each segment, statistical features are extracted from each axis of the (possibly transformed) data and are concatenated to construct the feature vector associated with that segment. For instance, for dataset A and the reference case, there are 5 units $\times$ 9 sensors $= 45$ axes in total, the features are extracted separately from each of these 45 axes over the given time segment, and concatenated into a single feature vector associated with that particular segment. The following features are considered: mean, variance, skewness, kurtosis, certain coefficients of the autocorrelation sequence,[5] and the five largest DFT peaks with the corresponding frequencies.[6] The number of features are 1170, 276, 234, 156, and 78 for datasets A–E, respectively, for the reference case. Following feature extraction, the features are normalized to the interval $[0, 1]$ for each subject in each dataset.

As the last step of the pre-processing stage, the number of features is reduced through PCA, which linearly and orthogonally transforms the feature space such that the transformed features are sorted in descending order of variance [96]. This approach allows us to consider only the first M dimensions in the classification process, decreasing the computational complexity and possibly improving classification if an appropriate value of M is chosen. Moreover, it enables us to make a comparison between the different datasets by equalizing the dimensionality of the feature space among them. To select an appropriate value for M, the eigenvalues of the covariance matrix of the feature vectors extracted from each of the five cases are sorted in descending order and plotted in Figure 2.7 for each dataset. $M = 30$ appears to be a suitable choice because there is a dramatic decrease from the first eigenvalue to the 30th in all five datasets.

2.3.2.2 Classification

Following feature reduction, classification is performed with four state-of-the-art classifiers. The parameters of the second and the third classifiers are jointly optimized by a grid search for all five cases, the two cross-validation techniques, and the five datasets. The classifiers and the parameter optimization process are explained below.

[5] Every fifth autocorrelation sample up to the 50th is used. The variance is included once as the first autocorrelation sample. Fewer coefficients may be used depending on the length of the segment.

[6] The separation between any two peaks in the DFT sequence is taken to be at least 11 samples. A smaller number of peaks can be used depending on the segment duration.

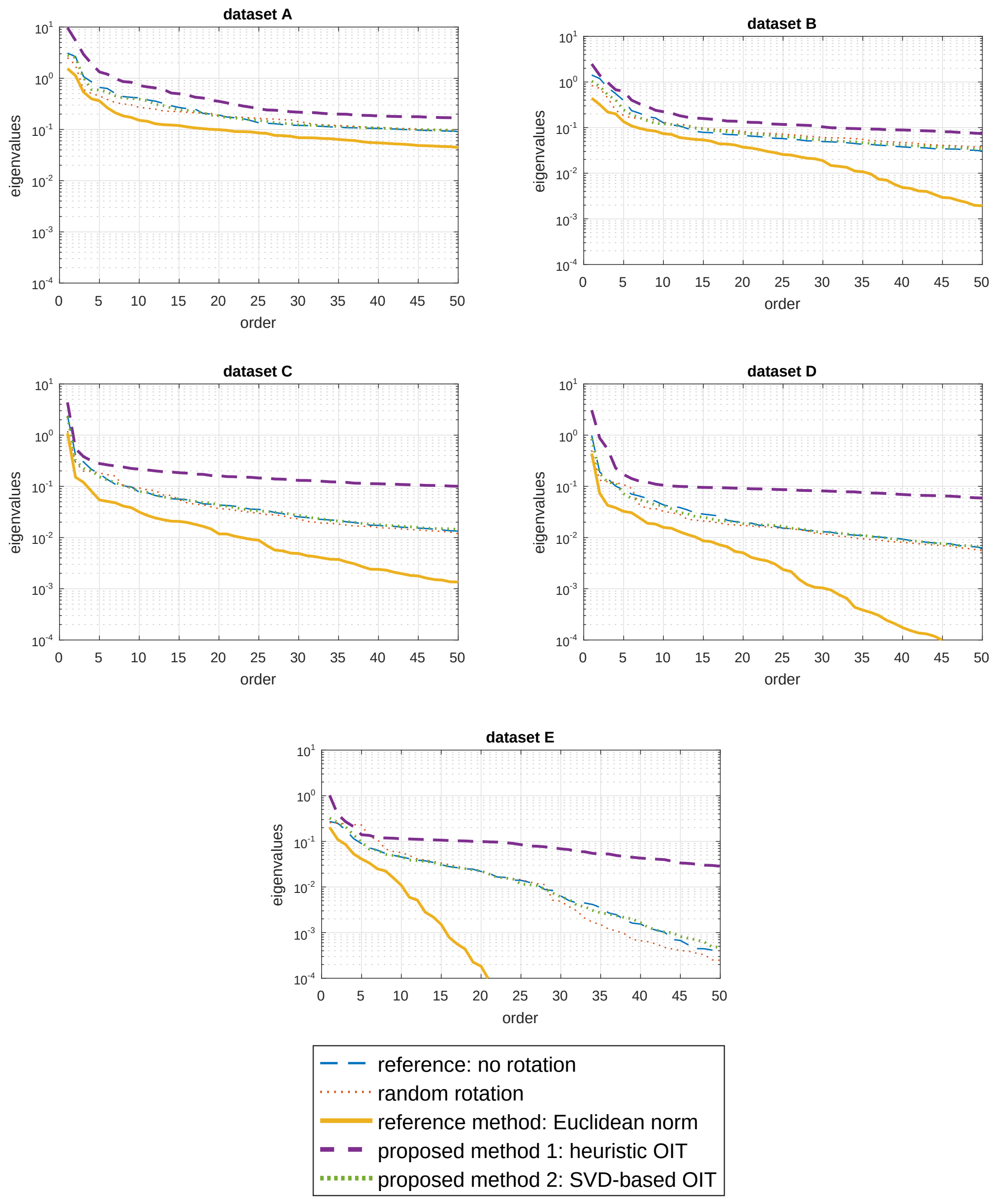


Figure 2.7: The first 50 eigenvalues of the covariance matrix in descending order for the features extracted from the data transformed according to the five cases.

1. **Bayesian Decision Making (BDM):** To train a BDM classifier, for each activity class a multi-variate Gaussian distribution is fitted using the training feature vectors of that class by using maximum likelihood estimation. This process involves estimating the mean vector and the covariance matrix for each class. Then, for a given test vector, its conditional probability, conditioned on the class information (i.e., the probability given that it belongs to a particular class) can be calculated. The class that maximizes this probability is selected according to the maximum a posteriori (MAP) decision rule [96, 97].

2. *k*-**Nearest-Neighbor (*k*-NN):** The k-NN classifier requires storing training vectors. A test vector is classified by using majority voting on the classes of the k nearest training vectors to the test vector in terms of the Euclidean distance, where k is a parameter that takes integer values [96, 97]. In this study, k values ranging from 1 to 30 have been considered for all cases, cross-validation techniques, and datasets. The value $k = 7$ is found to be suitable and is used throughout this work.

3. **Support Vector Machines (SVM):** The SVM is a binary classifier in which the feature space is separated into two classes by an optimal hyperplane that has the maximum margin [97]. In case the original feature space may not be linearly separable, it can be implicitly and nonlinearly mapped to a higher-dimensional space by using a kernel function, which represents a measure of similarity between two data vectors $\mathbf{x}$ and $\mathbf{y}$. There are two commonly used kernels: the Gaussian radial basis function (RBF), $f_{\text{RBF}}(\mathbf{x}, \mathbf{y}) = e^{-\gamma\|\mathbf{x}-\mathbf{y}\|^2}$, and the linear kernel, $f_{\text{linear}}(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T\mathbf{y}$. In this study, we use the former, which is equivalent to mapping the feature space to a Hilbert space of infinite dimensionality. The reason for this choice is that there is no need to consider the linear kernel if the RBF kernel is used with optimized parameters [98], which is the case here. Then, binary classification is performed according to which side of the hyperplane the test vector resides on. To use the SVM with more than two classes, a one-versus-one approach is followed where a binary SVM classifier is trained for each class pair. A test vector is classified with all pairs of classifiers and the classifier with the highest confidence makes the class decision [99]. The MATLAB

toolbox LibSVM is used for the implementation [100]. The two parameters of the SVM classifier, C and γ, are optimized jointly over all five cases, both cross-validation techniques, and all five datasets. The parameter C is the penalty parameter of the optimization problem of the SVM classifier (see Equation (2.1) in [101]) and γ is the parameter of the Gaussian RBF kernel described above. A two-level grid search is used to determine the parameter pair that performs the best over all cases, cross-validation techniques, and datasets. In the coarse grid, the parameters are selected as $(C, \gamma) \in \{10^{-5}, 10^{-3}, 10^{-1}, \ldots, 10^{15}\} \times \{10^{-15}, 10^{-13}, 10^{-11}, \ldots, 10^{3}\}$ and the best parameter pair is found to be $(C^{*}, \gamma^{*}) = (10^{1}, 10^{-1})$. Then, a finer grid around (C^{*}, γ^{*}) on the set $(C, \gamma) \in 100\,\mathcal{P} \times \mathcal{P}$, with $\mathcal{P} = \{0.01, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.7, 1, 3, 5\}$ reveals the best parameter pair $(C^{**}, \gamma^{**}) = (40, 0.2)$, which is used in all five cases, cross-validation techniques, and datasets considered in this chapter.

4. **Artificial Neural Networks (ANN):** An ANN consists of neurons, each of which produces an output that is a nonlinear function (called the activation function) of a weighted linear combination of multiple inputs and a constant. In this study, the sigmoid function, $g(x) = (1 + e^{-x})^{-1}$, is used as the activation function [97]. A multi-layer ANN consists of several layers of neurons. The inputs to the first layer are the elements of the feature vector. In the last layer, a neuron is allocated to each of the K classes. The number of hidden-layer neurons is selected as the nearest integer to $\frac{1}{2}\left(\frac{\ln(2K)}{\ln 2} + 2K - 1\right)$, depending on the number of classes K. (As a rule of thumb, each class is assumed to have two linearly separable subclasses. Then, the number of neurons in the hidden layer is selected as the average of the optimistic and pessimistic cases. In the former, $\frac{\ln(2K)}{\ln 2}$ neurons are required to have the hyperplanes intersect at different positions, whereas in the latter, $2K - 1$ neurons are required for parallel hyperplanes [102].) Training an ANN can be implemented in various ways and determines the weights of the linear combination for each neuron. The desired output is one for the neuron corresponding to the class of the input vector and zero for the output neurons of the other classes. The back-propagation algorithm is used for training, which iteratively minimizes

the errors in the neuron outputs in the least-squares sense, starting from the last layer and proceeding backwards [103]. The weights are initialized with a uniform random distribution in $[0, 0.2]$ and the learning rate is chosen as 0.3. An adaptive stopping criterion is used, which terminates the algorithm at the ith epoch (that is, when each training vector has been used exactly i times) if $\min\{\mathcal{E}_{i-9}, \mathcal{E}_{i-8}, \ldots, \mathcal{E}_i\} > \mathcal{E}_{i-10} - 0.01$, where $\mathcal{E}_i$ is the average of the sum of the squared errors over all the training vectors in the last layer's outputs at the ith epoch. In other words, the algorithm stops when the errors at (any of) the last 10 epochs are not significantly smaller than the error at the 11th epoch from the end. In classification, a test vector is given as the input to the ANN and the output neuron with the maximum output indicates the class decision.

2.3.2.3 Cross Validation

The accuracies of the classifiers are determined by two cross-validation techniques: P-fold and leave-one-subject-out (L1O). In P-fold cross validation, the dataset is randomly divided into $P = 10$ equal partitions and the data in each partition are classified with a classifier trained by the data in all the remaining partitions. L1O cross validation is similar to P fold, the main difference being that data are partitioned subject-wise so that each partition contains the data acquired from one of the subjects [97]. In L1O, feature vectors of a given subject are left out while training the classifier with the remaining subjects' feature vectors. The left out subject's feature vectors are then used for testing (classification). This process is repeated for each subject. L1O is highly affected by the variation in the data across the subjects because the training and test sets contain different subjects' data. Hence, it is more challenging than subject-unaware cross-validation techniques such as repeated random sub-sampling or P fold [104].

2.3.3 Comparative Evaluation Based on Accuracy

We naturally expect the accuracy achieved with the proposed transformations to be lower compared to the reference case because neither of the two transformations preserves the direction of the gravity vector detected by the accelerometers nor the direction of the Earth's magnetic North measured by the magnetometers. After transforming, absolute sensor orientations are no longer available. Removing this information is necessary to provide the user the flexibility to place the sensor units at any orientation.

The activity recognition accuracies for datasets A–E are shown in Figure 2.8, along with the standard deviations over the cross-validation iterations. For each dataset, the classifier accuracies are presented for the five cases for each cross-validation technique. We observe that when the standard activity recognition system is used with randomly oriented sensors (the random rotation case), the accuracy drops by 21.2% on the average, compared to the reference case. Using only the Euclidean norm improves the accuracy drop for datasets A–C, and causes an average degradation of 13.5% in accuracy compared to the reference case, over all datasets. We also observe that both of the proposed OIT techniques significantly improve the accuracy drop compared to the random rotation case in most situations. On the average, proposed methods 1 (with 9 elements) and 2 decrease the accuracy by 15.5% and 7.6%, respectively, compared to the reference case; hence, the latter is superior to the former most of the time. When the first three or the first six elements of the heuristic OIT are used, the performance depends on the dataset and the cross-validation technique used and is comparable to using all nine elements. The accuracy obtained by using the SVD-based OIT is comparable with the reference case for all datasets except for C for which it is lower.

The most accurate classifier, in general, is the SVM; its accuracy is especially greater than the other classifiers when the sensors are oriented randomly. This result shows that the SVM is robust against challenges associated with the classification problem and imperfections in the data, even though the same parameter values are used for the SVM classifier throughout the study. The

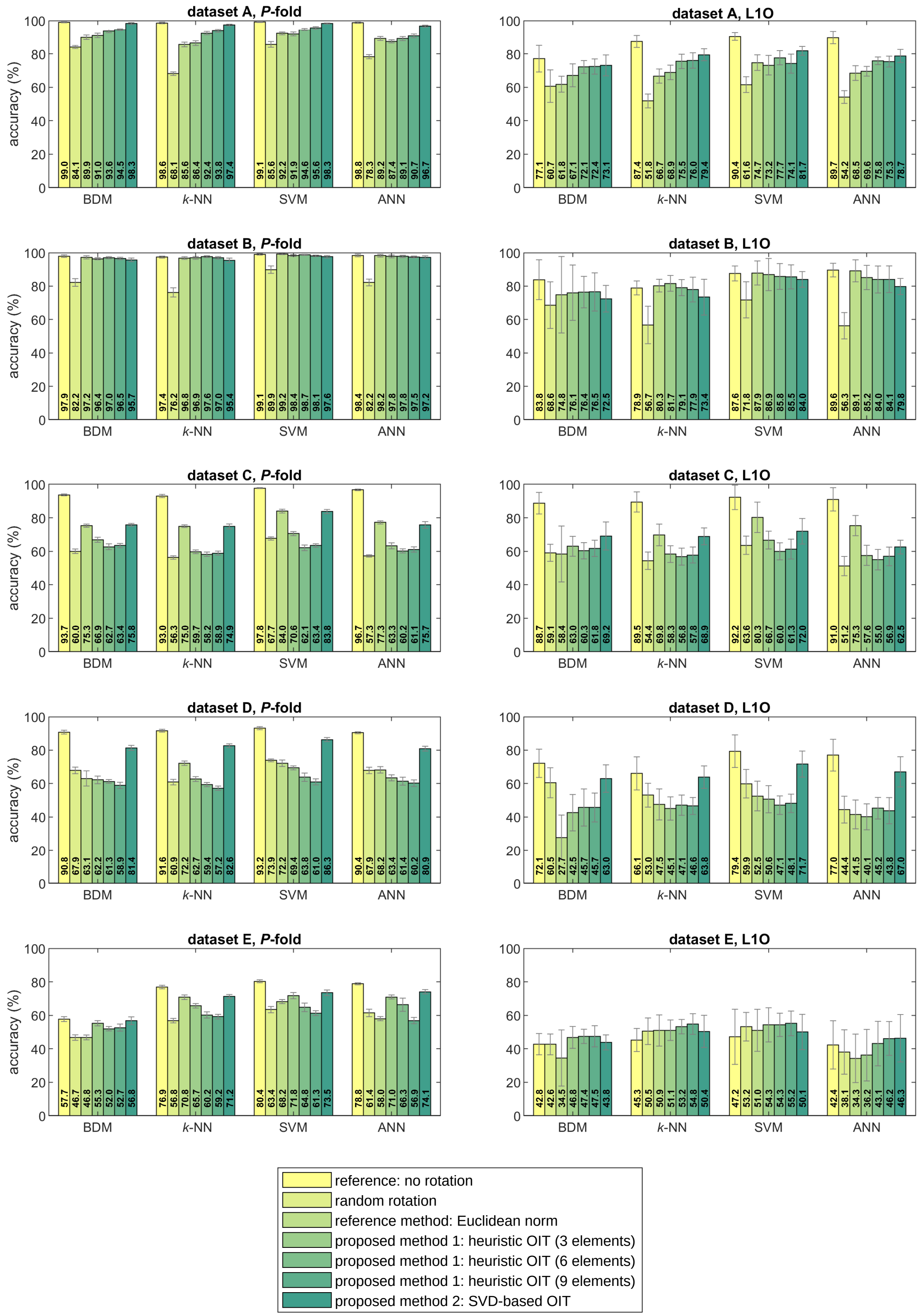


Figure 2.8: Accuracies shown as bars or horizontal lines for all the cases, datasets, classifiers, and cross-validation techniques. The vertical sticks indicate plus/minus two standard deviations around the mean over the cross-validation iterations.

robustness of the SVM in different problems is consistent with the results obtained in [47]. The second most robust classifier is BDM, which is also more accurate than most of the other classifiers for random rotation for all datasets. We attribute the robustness of BDM to its "coarseness" in classification, which improves the accuracy in classifying imperfect data. In other words, because each segment in the training and test data is rotated randomly and differently, the feature vectors are scattered in the feature space. In this case, one needs to train a classifier that will not separate the feature space haphazardly based on individual feature vectors, but rather consider the simple common properties of the feature vector constellations of the classes. Binary decision making realizes this successfully, fitting a smooth Gaussian distribution to the training data of each class. However, the k-NN classifier, for instance, partitions the feature space into regions with complicated boundaries and performs worse for randomly rotated data.

Since we use the same methodology to classify the activities in all datasets, we are able to make a fair comparison between the datasets. Referring to Figure 2.8, we observe that the activity recognition accuracy highly differs among the datasets even for the reference case where no transformation is applied: Datasets D and E result in lower accuracy than datasets A–C for all four classifiers. In particular, the classifiers perform poorly for dataset E, especially for L1O cross validation, where most of the segments are incorrectly classified. This result shows that a single tri-axial accelerometer worn on the chest is not sufficient to recognize relatively complicated activities, such as working at a computer (E_1) or talking while standing (E_7). Rotating or transforming the data does not have a significant effect on the results for dataset E and L1O cross validation, indicating that the recorded data do not contain sufficient information about the activities. We also observe in all datasets that the L1O cross-validation technique results in much lower accuracy than P fold because of the variations in the data across the subjects who perform the activities [17].

2.4 Discussion

We have not recorded a new dataset for incorrectly oriented sensor units in this study. The first reason for this choice is that it would not have been possible to compare the five cases based on the same dataset because we would not have been able to obtain the results in the reference case using a dataset recorded with different sensor orientations. Considering that there are usually significant variations in the data recorded from activities performed by different subjects and by the same subject at different times [17, 105], comparing the five cases based on different datasets would not be fair. The second reason is that the proposed OITs completely remove the absolute orientation information from the data, which means that the transformed sequences would be exactly the same if the sensor units were oriented differently. A third reason is the difficulty of selecting the incorrect sensor orientations considered in the new dataset because this would highly affect the results of random rotation.

We assume that each sensor unit may be placed at any orientation at a given position but the orientation on the body must remain the same in the short term. We make this assumption because we wish to preserve the information related to the rotational motion of the body related to the activities performed and only remove that related to the absolute orientation of the sensors. To this end, in the heuristic OIT, we extract some quantities from the sensor sequences and their time differences that are invariant to sensor orientation. If the sensor orientation with respect to the body changes over time, these difference sequences will be affected. However, the heuristic OIT uses differences spanning at most four consecutive time samples, which correspond to a duration of three sampling periods (0.12, 0.375, 0.06, 0.03, and 0.06 sec in datasets A–E, respectively). Thus, it is sufficient to maintain the sensor unit orientations for three sampling periods to obtain uncorrupted transformed sequences. This result translates into practice, where the sensor orientations are allowed to deviate slowly provided that the deviation over three sampling periods is negligible. This property is not valid for the SVD-based OIT, which requires that the sensor unit orientations with respect to the body remain the same throughout the time period the transformation is applied (one segment).

However, since each segment is transformed independently in both the training and test phases, the sensor unit orientations in each segment may be completely different. This result would have no effect on the transformed sequences nor the accuracy.

Unlike some studies that assume correct sensor placement in the training phase, such as [31], we allow users of wearable systems the flexibility to place the sensor units at any orientation during both the training and test phases for both OIT techniques. Many studies consider only a small and finite number of orientations, whereas in our approach, orientation angles can take values over a continuum. This method is advantageous because of the inevitable deviations in sensor placement over time. We also do not make any assumptions regarding the nature of the daily activities. For instance, in [67], to estimate the directions of the forward-backward and vertical axes of the human body, it is assumed that the long-term average of the acceleration provides the direction of the gravity vector, and most of the variations perpendicular to the vertical axis are along the forward-backward direction of the body. Similar assumptions are made in [106]. These assumptions are not valid in applications such as monitoring elderly, disabled, or injured people, and children who are more likely to place the sensor units incorrectly because of these users' limitations, or in evaluating physical therapy or sports exercises, where the subjects' body movements can be more vigorous and different than those in daily activities. Thus, we believe that the existing techniques are not applicable to the generic activity recognition framework and that the approaches proposed here allow more flexibility.

The most important advantage of our methodology is that the OIT techniques that we propose can be readily used without much effort at the beginning of the typical activity recognition paradigm (consisting of segmentation, feature extraction and reduction, and classification, Figure 2.6), provided that rule-based heuristic approaches that rely on the physical meanings of the raw sensor measurements are not used. The SVD-based OIT can be applied to the raw sensor measurements in any kind of system that processes multi-dimensional time-domain sequences. The only requirement to apply the heuristic OIT is that the system should be able to process up to 9D time-domain sequences instead of 3D ones.

2.5 Run-Time Analysis

To assess the computational cost of pre-processing the sequences, the run times of the proposed OIT techniques and the Euclidean norm method are provided in Table 2.2 for each dataset. We observe that the calculation of the heuristic OIT takes the longest, followed by the SVD-based OIT, and the Euclidean norm approach. As the number of elements included in heuristic OIT is increased from 1 (Euclidean norm) to 3 to 6 to 9, the run time naturally increases. The 3-element and 6-element versions of the heuristic OIT algorithm could be suitable for deployment on resource-limited platforms for which the calculation of an inverse cosine or a vector dot/cross product is a significant effort.

Table 2.2: Run times of the three OIT techniques (in sec) for datasets A–E.

method	**dataset**				
	A	**B**	**C**	**D**	**E**
Euclidean norm	6.60	2.34	5.52	4.12	3.51
proposed method 1: heuristic OIT (3 elements)	28.93	2.23	6.57	5.95	2.76
proposed method 1: heuristic OIT (6 elements)	191.41	10.10	44.06	49.24	21.01
proposed method 1: heuristic OIT (9 elements)	369.24	17.50	84.24	91.45	38.67
proposed method 2: SVD-based OIT	70.03	4.12	20.43	59.74	8.33

We also investigate the run times of the classifiers that show some variation. The classifiers' run times are presented separately for the five cases and the two cross-validation techniques for dataset A in Table 2.3.[7] In the rows entitled "run time," each entry is the sum of the training and classification times of all the test feature vectors in an average cross-validation iteration. It is observed that k-NN is significantly faster than the other classifiers, whereas the ANN and SVM are relatively slow. The variation in the run time across the five cases and the two cross-validation techniques is not as much as the variation across the classifiers.

[7]The processing was performed on 64-bit MATLAB® R2016a running on a laptop computer containing a quad-core processor Intel® Core™ i7-4720HQ with a clock speed of 2.6–3.6 GHz and 16 GB of RAM. For the heuristic OIT, run times of the version with nine elements is provided.

Table 2.3: Total run time (training and classification of all test feature vectors), average training time per single cross-validation iteration, and average classification time per feature vector for dataset A.

	classifier	reference	Euclidean norm	random rotation	proposed method 1	proposed method 2	reference	Euclidean norm	random rotation	proposed method 1	proposed method 2
		P-fold					L1O				
run time (s)	BDM	1.31	1.62	1.30	1.29	1.31	1.63	1.61	1.69	2.59	2.38
	k-NN	0.15	0.16	0.16	0.16	0.15	0.18	0.17	0.19	0.42	0.26
	SVM	13.24	36.05	12.23	30.50	13.65	12.07	28.42	11.70	34.53	17.50
	ANN	8.75	12.80	13.85	14.48	10.12	7.99	9.33	9.13	11.35	9.33
training time (s)	BDM	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
	k-NN	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	SVM	12.84	30.19	11.68	29.89	13.22	11.64	27.52	11.06	33.58	16.81
	ANN	8.56	12.77	13.83	14.45	10.10	7.97	9.30	9.10	11.31	9.30
classification time (ms)	BDM	1.42	1.76	1.42	1.40	1.42	1.42	1.40	1.47	2.25	2.08
	k-NN	0.16	0.17	0.17	0.17	0.16	0.15	0.15	0.16	0.37	0.22
	SVM	0.31	0.72	0.48	0.52	0.34	0.29	0.69	0.46	0.66	0.45
	ANN	0.02	0.02	0.02	0.03	0.02	0.02	0.02	0.02	0.03	0.02

Each entry in the rows entitled "training time" is the average duration of training a classifier in a *single* P-fold or L1O iteration. The ANN and SVM are about three orders of magnitude slower than the other classifiers in training, in exchange for higher accuracy. The k-NN classifier does not require any training because it only needs to store the training feature vectors for classification. The training time of BDM does not significantly depend on the data, hence, it is nearly the same for each of the five cases and the two cross-validation techniques. On the other hand, the training times of the SVM and ANN highly differ across the five cases and the two cross-validation techniques, and training is faster in the reference case and proposed method 2.

The rows entitled "classification time" contain the average durations of classifying *a single* test feature vector for each case and each cross-validation technique. In all cases, BDM has the longest classification time, whereas ANN has the shortest. The classification time of the SVM is case dependent, whereas the classification times of the other classifiers are comparable for each of the five cases.

2.6 Concluding Remarks

The aim of this chapter was to solve the generic problem of placing sensor units at incorrect orientations, instead of partially solving both the incorrect position and orientation problems under restrictive assumptions. The results show that both OIT techniques that we propose solve the issue of incorrect sensor unit orientation in activity recognition, with an average absolute reduction of 11.6% in accuracy. In particular, compared to the reference case, the SVD-based OIT causes an average accuracy degradation of 7.6%, whereas this value is 15.5% for the heuristic OIT. On the other hand, without any transformation, random sensor unit orientation decreases the accuracy by 21.2% on average, which shows the effectiveness of the transformations that we propose. The use of these transformations requires neither restrictive assumptions about the sensor and activity types nor about the sensor unit positions. The proposed methodology can be used in the pre-processing stage of existing wearable systems without much effort, making them invariant to sensor unit orientation.

Chapter 3

Invariance to Sensor Unit Orientation Based on Orientation Estimation

In Chapter 2, we have proposed two different geometrical transformations, heuristic OIT and SVD-based OIT, for invariance to sensor unit orientation. In this chapter, we develop an alternative OIT that exploits the data acquired by accelerometers, gyroscopes, and magnetometers. For orientation invariance, we represent the sensor data with respect to the Earth frame. We also calculate the sensor rotations between consecutive time samples and represent them by quaternions in the Earth frame. The overview of the proposed method is depicted in Figure 3.1. For this purpose, we need to estimate the sensor unit orientation with respect to the Earth frame. In this chapter, we employ an existing OEM based on the Gauss-Newton (GN) algorithm [107], whereas in Chapter 4, we develop a novel OEM and compare it with the existing methods. The work presented in this chapter has appeared in [19].

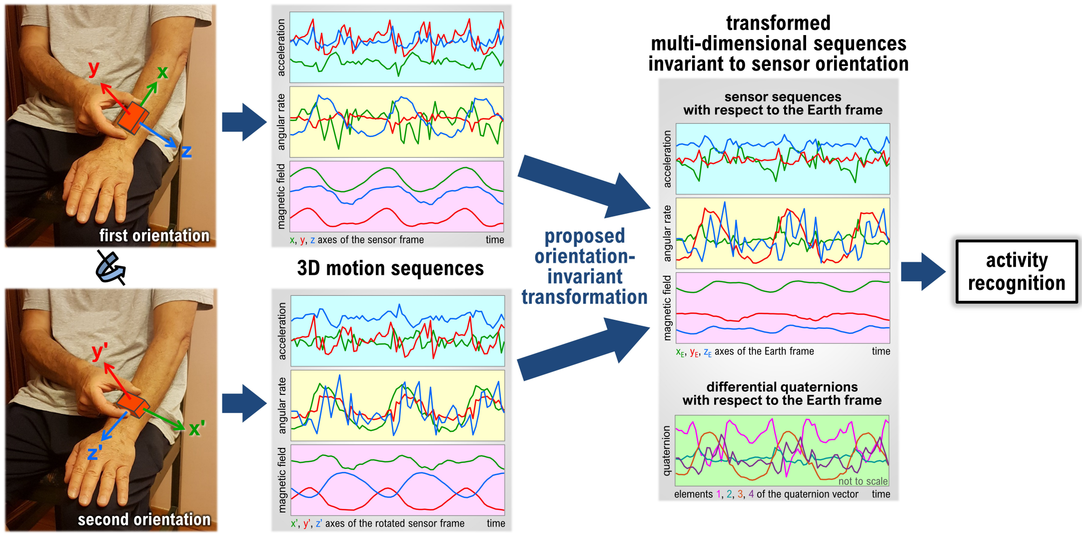
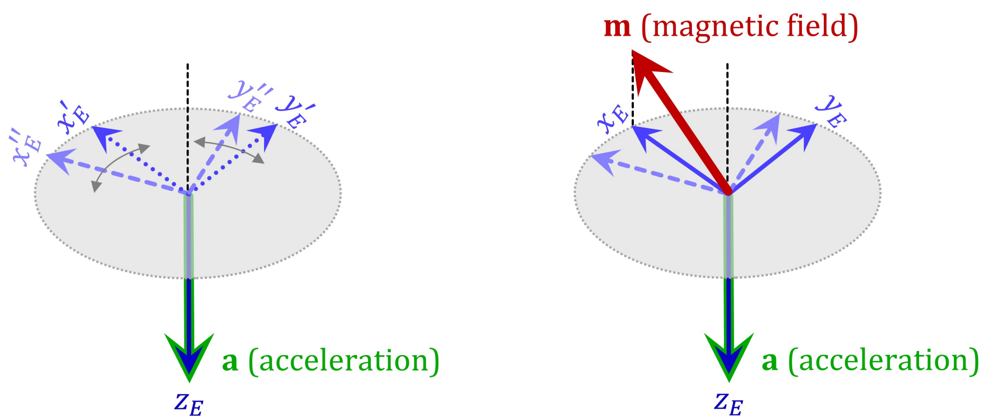


Figure 3.1: An overview of the proposed method for sensor unit orientation invariance.

3.1 Estimation of Sensor Unit Orientation

We capture the body motions using three types of tri-axial wearable motion sensors: accelerometer, gyroscope, and magnetometer. The acceleration vector acquired by an accelerometer approximately points in the down direction of the Earth frame, provided that the gravitational component of the total acceleration is dominant over the acceleration components resulting from the motion of the sensor unit. However, even if the acceleration vector consists of mainly the gravitational component, by itself it is not sufficient to estimate the sensor unit orientation because there exist infinitely many solutions to the sensor unit orientation, obtained by rotating the correct solution about the direction of the acquired acceleration vector (Figure 3.2(a)). Hence, we need to incorporate the magnetometer into the orientation estimation as well.

The magnetic field vector acquired by a magnetometer points in a fixed direction in the Earth frame (the magnetic North) (Figure 3.2(b)), provided that there are no external magnetic sources or distortion and the variation of the Earth's magnetic field is neglected. By taking the reference directions obtained from the accelerometer and the magnetometer as the vertical axis and the (magnetic) North axis of the Earth frame, respectively, we can estimate the *static orientation* of the sensor unit with respect to the Earth frame. However, this estimation is reliable only in the long term because the gravity component is superposed with the acceleration caused by the motion of the unit and the Earth's magnetic field is superposed with the external magnetic sources (if any). Hence, we also estimate the *dynamic orientation* by integrating the gyroscope angular rate output, which is reliable only in the short term because of the drift error [78]. To obtain an accurate orientation estimate both in the short and long term, we merge these two sources of information. Thus, we exploit the information provided by the three types of sensors to determine the sensor unit orientation with respect to the Earth frame as a function of time.



(a) (b)

Figure 3.2: (a) With only the acquired acceleration field vector **a**, there exist infinitely many solutions to the sensor unit orientation (two are shown); (b) the acquired magnetic field vector **m** uniquely identifies the sensor unit orientation.

Once we estimate the sensor unit orientation with respect to the Earth frame, we can transform the acquired data from the sensor frame to the Earth frame such that they become invariant to sensor unit orientation. In addition, to include the information about the rotational motion of the sensor unit, we represent the sensor unit rotation between consecutive time samples in the Earth frame by using a similarity transformation. We show that appending this rotational motion data to the sensor data and representing both in the Earth frame improves the activity recognition accuracy.

Existing orientation estimation techniques for the sensor units can be classified into deterministic, stochastic, and frequency-based approaches [108]. Since stochastic approaches are computationally intensive and frequency-based approaches are relatively not very robust [108], we focus on deterministic methods in this study. While most existing OEMs obtain the dynamic estimate by integrating the angular rate vector, they estimate the static orientation in different ways based on the acceleration and magnetic field vectors. Simple, non-iterative techniques such as TRIAD [109] and factored quaternion algorithm [110] rely on geometric approaches to respectively calculate the rotation matrices and quaternions that represent the static orientation. Since they do not utilize gyroscopic angular rate measurements, a dynamic orientation estimate is not involved. These approaches are mainly intended for slowly moving or static sensor units. On the other hand, there exist orientation estimation studies that are based on the use of iterative algorithms such as Gradient-Descent (GD) [111], GN [107], and Levenberg-Marquardt (LM) [112], the last of which is a blend of GD and GN. Linear and extended Kalman filters (KF) are also employed but with relatively high computational cost [113–115]. The iterative algorithms first estimate the static orientation by minimizing a cost function that decreases as the vertical and North directions of the Earth approach the acceleration and magnetic field vectors, respectively. The method proposed in [111] uses the GD algorithm to align the upward direction of the Earth frame with the acceleration vector and the North direction with the horizontal component of the magnetic field vector. To improve the computational efficiency, it uses an approximate solution and iterates the GD algorithm only once. The algorithm proposed in [107] uses the GN method to make an alignment similar to that in [111]; however, unlike [111], it calculates the solution without making any approximations. The same study also provides a brief comparison between GD and GN algorithms for which the number of iterations is limited to ten and three, respectively. Based on the results, it is stated that GN is faster and does not require as many iterations as GD to reach the minimum point and the estimated orientation angles do not fluctuate as much around their true values. These iterative algorithms may not always converge to the global minimum and are computationally intensive because they need to be iterated several times at each time sample. Once the static orientation estimate is obtained through a number of iterations, the static and dynamic orientation estimates are combined through weighted averaging at each time sample. Existing OEMs summarized above are reviewed in [108] in more detail.

We define the Earth's coordinate frame E according to the North-East-Down (NED) convention [116] such that the Earth's z axis, z_E, points downwards and the Earth's x axis, x_E, points in the direction of the component of the Earth's magnetic field that is perpendicular to the z axis, which is roughly the North direction, as illustrated in Figure 3.3.

Let S_n be the rotating sensor frame at time sample n. Estimating the sensor unit orientation involves calculating a 3×3 rotational transformation matrix $\mathbf{R}_{S_n}^{E}$ that describes the sensor frame S_n with respect to the Earth frame E at each time sample n. The Earth frame and the sensor frame at consecutive time samples n and $n+1$ are depicted in Figure 3.4 together with the rotation matrices relating these coordinate frames. In this chapter, we adopt the OEM in [107], which is explained in the Appendix. The short-term orientation estimate is calculated by integrating the angular rate acquired by the gyroscope. For the long-term orientation estimation, the GN method is used to minimize a cost function which decreases as the acceleration vector points downwards in the Earth frame and as the horizontal component of the magnetic field vector is aligned with the North direction of the Earth frame. Then, the short- and long-term orientation estimates are combined through weighted averaging [107].

3.2 Sensor Signals with Respect to the Earth Frame

The tri-axial data acquired on the x, y, and z axes of each sensor type in the sensor coordinate frame S_n naturally depend on the orientations of the sensor units. Our approach is based on transforming the acquired data from the sensor frame to the Earth frame.

Let $\mathbf{v}^S[n] = \left(v_x^S[n], v_y^S[n], v_z^S[n]\right)^T$ be the data vector in $\mathbb{R}^3$ acquired from the x, y, z axes of a tri-axial sensor at time sample n. To represent $\mathbf{v}^S[n]$ with respect to the Earth frame, we pre-multiply it by the estimated sensor unit orientation at that

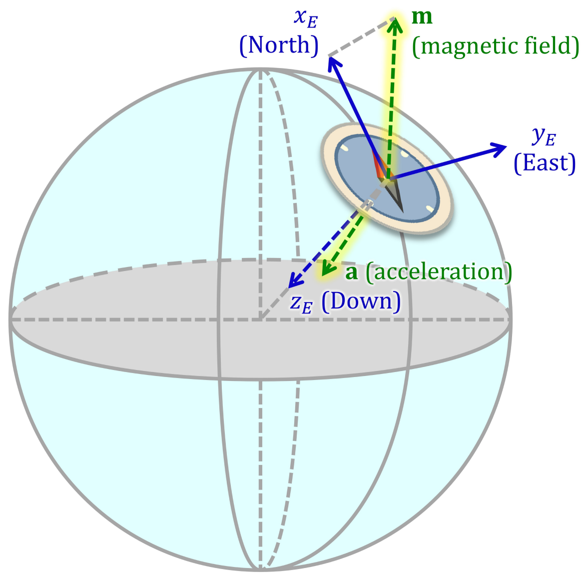


Figure 3.3: The Earth frame illustrated on an Earth model with the acquired reference vectors.

time sample, which is the rotation matrix relating the S_n frame to the E frame:

$$\mathbf{v}^E[n] = \mathbf{R}_{S_n}^E \, \mathbf{v}^S[n] \tag{3.1}$$

The components of the vector $\mathbf{v}^E[n] = \left(v_x^E[n], v_y^E[n], v_z^E[n]\right)^T$ are represented with respect to the x_E, y_E, z_E axes of the Earth frame and are invariant to the sensor unit orientation.

3.3 Differential Sensor Rotations with Respect to the Earth Frame

In addition to the data transformed to the Earth frame, we propose to incorporate the information contained in the change in the sensor unit orientation over time. While the sensor units can be placed at arbitrary orientations, we require that during data acquisition their orientations remain fixed with respect to the body

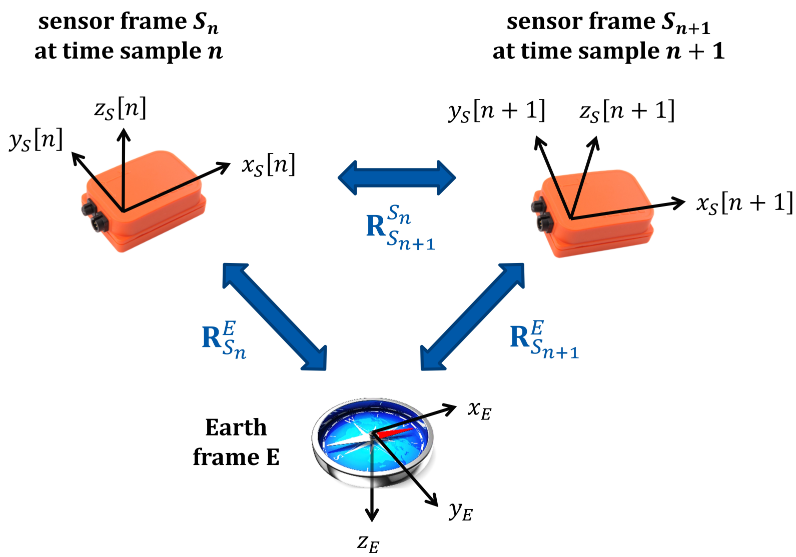


Figure 3.4: The Earth and the sensor coordinate frames at two consecutive time samples with the rotational transformations relating them.

part they are placed on. In other words, the sensor units need to be firmly attached to the body and are not allowed to rotate freely during the motion. However, this restriction is only necessary in the short term over one time segment (for dataset A used in this chapter). Under this restriction, the rotational motion of the body parts on which the sensor units are worn can be extracted from the acquired data correctly regardless of the initial orientations of the units.

Note that we can easily calculate the sensor unit orientation $\mathbf{R}_{S_{n+1}}^{S_n}$ at time sample $n+1$ relative to the sensor orientation at time sample n as

$$\mathbf{C}_n \triangleq \mathbf{R}_{S_{n+1}}^{S_n} = \mathbf{R}_E^{S_n}\,\mathbf{R}_{S_{n+1}}^{E} = \left(\mathbf{R}_{S_n}^{E}\right)^{-1}\mathbf{R}_{S_{n+1}}^{E} \tag{3.2}$$

for each n as shown in Figure 3.4. The matrix $\mathbf{C}_n$ is not invariant to sensor unit orientation because it represents the orientation of frame S_{n+1} with respect to S_n and depends on the orientation at which the sensor unit is fixed to the body. To observe this, let us assume that the sensor unit is placed at a different arbitrary orientation; that is, the sensor unit is rotated by an arbitrary rotation matrix $\mathbf{P}$ that is constant over time. Then, the acquired data are $\tilde{\mathbf{v}}^S[n] = \mathbf{P}^{-1}\,\mathbf{v}^S[n]$ for all n, represented with respect to the new sensor unit orientation $\tilde{S}_n$, and the sensor unit orientation with respect to the Earth is estimated as $\tilde{\mathbf{R}}_{S_n}^{E} = \mathbf{R}_{S_n}^{E}\,\mathbf{P}$ for all n. Note that the original rotation matrix is post-multiplied by $\mathbf{P}$ because $\mathbf{P}$ describes a rotational transformation with respect to the sensor frame, not the Earth frame [117]. For the new sensor unit orientation, the rotation of the sensor unit between time samples n and $n+1$ can be calculated as

$$
\begin{aligned}
\tilde{\mathbf{C}}_n &= \tilde{\mathbf{R}}_{S_{n+1}}^{S_n} \\
&= \tilde{\mathbf{R}}_{E}^{S_n}\,\tilde{\mathbf{R}}_{S_{n+1}}^{E} \\
&= \left(\tilde{\mathbf{R}}_{S_n}^{E}\right)^{-1}\,\tilde{\mathbf{R}}_{S_{n+1}}^{E} \\
&= \left(\mathbf{R}_{S_n}^{E}\,\mathbf{P}\right)^{-1}\,\left(\mathbf{R}_{S_{n+1}}^{E}\,\mathbf{P}\right) \\
&= \mathbf{P}^{-1}\,\left(\mathbf{R}_{S_n}^{E}\right)^{-1}\,\mathbf{R}_{S_{n+1}}^{E}\,\mathbf{P} \\
&= \mathbf{P}^{-1}\,\mathbf{R}_{E}^{S_n}\,\mathbf{R}_{S_{n+1}}^{E}\,\mathbf{P} \\
&= \mathbf{P}^{-1}\,\mathbf{R}_{S_{n+1}}^{S_n}\,\mathbf{P} \\
&= \mathbf{P}^{-1}\,\mathbf{C}_n\,\mathbf{P}
\end{aligned}
\tag{3.3}
$$

Since $\tilde{\mathbf{C}}_n \neq \mathbf{C}_n$ in general, $\mathbf{C}_n$ is *not* invariant to sensor unit orientation. We can make the rotational transformation $\mathbf{C}_n$ invariant to sensor unit orientation by representing it in the Earth frame. Hence, we transform $\mathbf{C}_n$ from the sensor frame S_n to the Earth frame E by using a similarity transformation [118]:

$$
\mathbf{D}_n = \left(\mathbf{R}_{E}^{S_n}\right)^{-1}\mathbf{C}_n\left(\mathbf{R}_{E}^{S_n}\right) = \mathbf{R}_{S_n}^{E}\,\mathbf{R}_{S_{n+1}}^{S_n}\,\mathbf{R}_{E}^{S_n} = \mathbf{R}_{S_{n+1}}^{E}\,\mathbf{R}_{E}^{S_n} \tag{3.4}
$$

We call this transformation $\mathbf{D}_n$ *differential sensor rotation with respect to the Earth frame.*

It is straightforward to show that $\mathbf{D}_n$ is invariant to sensor unit orientation. Using a constant arbitrary rotation matrix $\mathbf{P}$ that relates the original and modified sensor unit orientations as before, we have:

$$\begin{aligned}
\tilde{\mathbf{D}}_n &= \tilde{\mathbf{R}}^{E}_{S_{n+1}}\,\tilde{\mathbf{R}}^{S_n}_{E} \\
&= \tilde{\mathbf{R}}^{E}_{S_{n+1}}\left(\tilde{\mathbf{R}}^{E}_{S_n}\right)^{-1} \\
&= \left(\mathbf{R}^{E}_{S_{n+1}}\,\mathbf{P}\right)\left(\mathbf{R}^{E}_{S_n}\,\mathbf{P}\right)^{-1} \\
&= \mathbf{R}^{E}_{S_{n+1}}\,\underbrace{\mathbf{P}\,\mathbf{P}^{-1}}_{\mathbf{I}_{3\times 3}}\left(\mathbf{R}^{E}_{S_n}\right)^{-1} \\
&= \mathbf{R}^{E}_{S_{n+1}}\,\mathbf{R}^{S_n}_{E} \\
&= \mathbf{D}_n
\end{aligned} \tag{3.5}$$

Thus, we observe that the differential rotation $\tilde{\mathbf{D}}_n$ with respect to the Earth frame, calculated based on the rotated data, is the same as the one calculated based on the original data ($\mathbf{D}_n$).

3.4 Comparative Evaluation of Proposed and Existing Methodology on Orientation Invariance for Activity Recognition

To demonstrate our methodology, we use the publicly available daily and sports activities dataset acquired by our research group earlier [80], which is named as dataset A in Chapter 2.[1] To acquire the dataset, each subject wore five Xsens MTx sensor units [88] (see Figure 3.6), each unit containing three tri-axial devices: an accelerometer, a gyroscope, and a magnetometer. The sensor units are placed on the chest, on both wrists, and on the outer sides of both knees, as shown in Figure 3.5. Nineteen activities are performed by eight subjects. For each activity performed by each subject, there are 45 (= 5 units × 9 sensors) time-domain sequences of 5 min duration, sampled at 25 Hz, and consisting of 7500 time samples

[1]The remaining datasets are not used in this chapter since they do not include data from a magnetometer.

each. The attributes of the dataset and the types of activities are provided in the second column of Table 2.1.

The activities can be broadly grouped into two, as shown in the second column of Table 2.1: In stationary activities (A_1–A_4), the subject stays still without moving significantly, whereas non-stationary activities (A_5–A_{19}) are associated with some kind of motion.

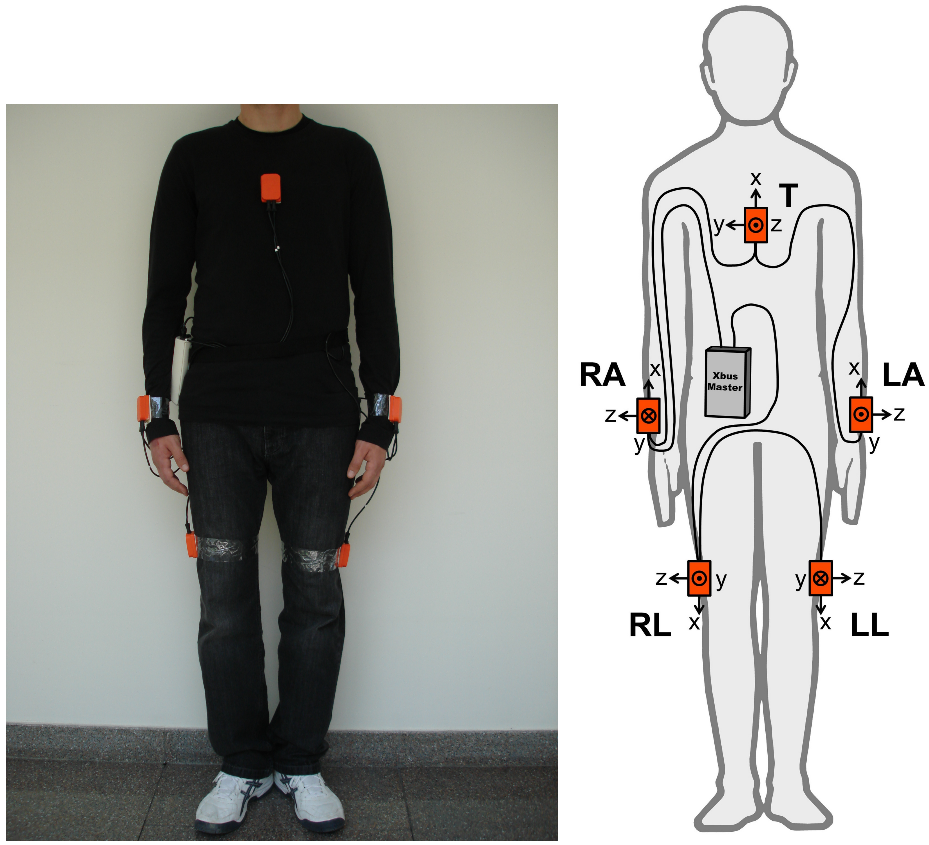
3.4.1 Activity Recognition

We employ the activity recognition procedure described in Section 2.3.2. The details are explained below:

3.4.1.1 Description of the Proposed and Existing Methodology on Orientation Invariance

In the pre-processing stage, seven data transformation techniques are considered to observe the effects of different sensor unit orientations on the accuracy and the improvement obtained with the existing and the proposed OITs:

- **Reference:** Data are not transformed and the sensor units are assumed to maintain their fixed positions and orientations during the whole motion. This corresponds to the standard activity recognition scheme, as in [86, 87, 92].
- **Random rotation:** This case is considered to assess the accuracy of the standard activity recognition scheme (without any OIT) when the sensor units are oriented randomly at their fixed positions. Instead of recording a new dataset with random sensor unit orientations, we randomly rotate the original data to make a fair comparison with the reference case. For this purpose, we randomly generate a rotational transformation matrix $\mathbf{R}$ as defined in Equation 2.15 independently for each time segment of each sensor unit (see Section 3.4.1.2 for segmentation). We pre-multiply each of the three



(a) (b)

Figure 3.5: (a) Positioning of the MTx units on the body; (b) connection diagram of the units (the body drawing in the figure is from `http://www.clker.com/clipart-male-figure-outline.html`; the cables, Xbus Master, and sensor units were added by the authors).

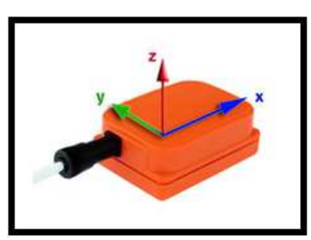


Figure 3.6: The Xsens MTx unit [88].

tri-axial sequences of that unit by the random rotation matrix corresponding to that segment of the unit: $\tilde{\mathbf{v}}[n] = \mathbf{R}\,\mathbf{v}^S[n]$. In this way, we simulate the situation where each sensor unit is placed at a possibly different random orientation in each time segment.

- **Euclidean norm method:** The Euclidean norm of the x, y, z components of the sensor sequences are taken at each time sample and used instead of using the original tri-axial sequences, as explained in Section 2.3.2.1. As reviewed in Section 1.1.2, this technique has been used in activity recognition to achieve invariance to the sensor unit orientation [26, 48, 63] or as an additional feature as in [32, 47, 64–66, 93].

- **Sequences along and perpendicular to the gravity vector:** In this method, the acceleration sequence in each time segment is averaged over time to approximately calculate the direction of the gravity vector. Then, for each sensor type, the sensor sequence's amplitude in this direction and the magnitude that is perpendicular to this direction are taken. This method has been used in [58, 67, 68] to achieve orientation invariance.

- **SVD-based transformation:** Sensory data are represented with respect to three principal axes that are calculated by SVD [18, 22], as explained in Section 2.2. The transformation is applied to each time segment of each sensor unit separately so that sensor units are allowed to be placed at different orientations for each segment.

To calculate the orientation-invariant transformations in the remaining two methods, we estimate the orientation $\mathbf{R}_{S_n}^{E}$ of each of the five sensor units as a function of time sample n as explained in the Appendix. For the algorithm to reach steady state rapidly, we append to the acquired signal a prefix signal of duration 1 s that consists of zero angular rate, a constant acceleration, and a constant magnetic field that are the same as the measurements at the first time sample.

- **Sensor sequences with respect to the Earth frame:** We transform the sensor sequences into the Earth frame using the estimated sensor orientations, as described by Equation (3.1). This method has been used in [69] to achieve invariance to sensor unit orientation in activity recognition. As an example, Figure 3.7(a) shows the accelerometer, gyroscope, and magnetometer data ($\mathbf{v}^S[n]$) acquired during activity A_{10} and Figure 3.7(b) shows the same sequences transformed into the Earth frame. We observe that the magnetic field with respect to the Earth frame does not significantly vary over time because the Earth's magnetic field is nearly constant in the Earth frame provided that there are no external magnetic sources in the vicinity of the sensor unit.

- **Proposed method: sensor sequences and differential quaternions, both with respect to the Earth frame:** We calculate the differential rotation matrix $\mathbf{D}_n$ with respect to the Earth frame for each sensor unit at each time sample n, as explained in Section 3.3. This rotation matrix representation is quite redundant because it has nine elements while any 3D rotation can be represented by only three angles. Since the representation by three angles has a singularity problem, we represent the differential rotation $\mathbf{D}_n$ compactly by a four-element quaternion $\mathbf{q}_n^{\mathrm{diff}}$ as

$$\mathbf{q}_n^{\mathrm{diff}} = \begin{bmatrix} q_1^{\mathrm{diff}} \\ q_2^{\mathrm{diff}} \\ q_3^{\mathrm{diff}} \\ q_4^{\mathrm{diff}} \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{1+d_{11}+d_{22}+d_{33}}}{2} \\ \frac{d_{32}-d_{23}}{4\sqrt{1+d_{11}+d_{22}+d_{33}}} \\ \frac{d_{13}-d_{31}}{4\sqrt{1+d_{11}+d_{22}+d_{33}}} \\ \frac{d_{21}-d_{12}}{4\sqrt{1+d_{11}+d_{22}+d_{33}}} \end{bmatrix} \tag{3.6}$$

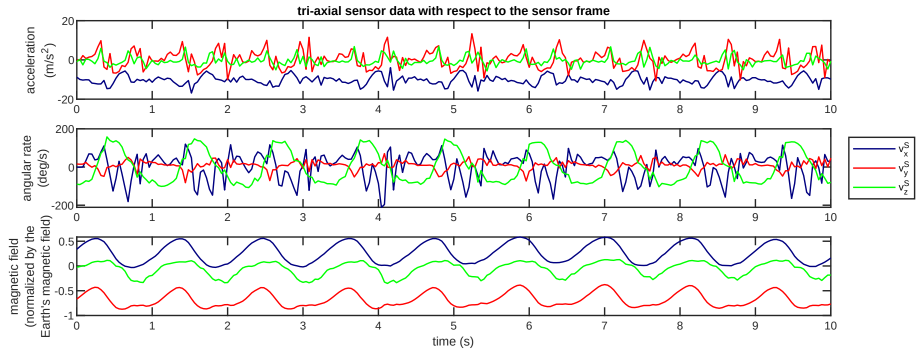
where $d_{ij}\ \ (i,j = 1,2,3)$ are the elements of $\mathbf{D}_n$ [119]. The vector $\mathbf{q}_n^{\mathrm{diff}}$ is called *differential quaternion with respect to the Earth frame* (the dependence of the elements of $\mathbf{q}_n^{\mathrm{diff}}$ and $\mathbf{D}_n$ on n has been dropped from the notation for simplicity). In the classification process, we use each element of $\mathbf{q}_n^{\mathrm{diff}}$ as a function of n, as well as the sensor sequences with respect to the Earth frame. Hence, there are four time sequences for the differential quaternion in addition to the three axes each of accelerometer, gyroscope, and magnetometer data for each of the five sensor units. Therefore, the transformed data comprises $(4+3+3+3)$ sequences $\times$ 5 sensor units $=$ 65 sequences in total.

We have observed that the joint use of the sensor sequences and differential quaternions, both with respect to the Earth frame, achieves the highest activity recognition accuracy compared to the other combinations. Representing rotational transformations by rotation matrices instead of quaternions degrades the accuracy. Omitting magnetometer sequences with respect to the Earth frame causes a slight reduction in the accuracy.

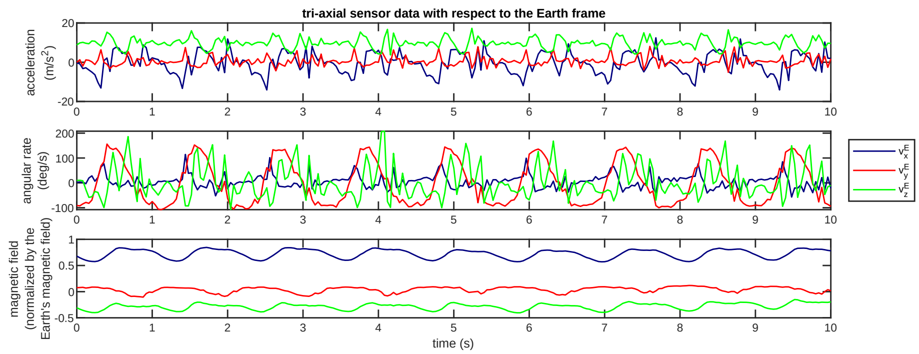
Figure 3.7(c) shows the nine elements of the differential rotation matrix $\mathbf{D}_n$ with respect to the Earth frame over time, which are calculated based on the sensor data shown in Figure 3.7(a). Figure 3.7(d) shows the elements of the differential quaternion $\mathbf{q}_n^{\text{diff}}$ as a function of n. The almost periodic nature of the sensor sequences (Figure 3.7(a)) is preserved in $\mathbf{D}_n$ and $\mathbf{q}_n^{\text{diff}}$ (Figure 3.7 (c) and (d)). The differential rotation is calculated between two consecutive time samples that are only a fraction of a second apart, hence the amplitudes of the elements of $\mathbf{D}_n$ and $\mathbf{q}_n^{\text{diff}}$ do not vary much. Since differential rotations involve small rotation angles (close to $0°$), the $\mathbf{D}_n$ matrices are close to the 3×3 identity matrix ($\mathbf{I}_{3\times3}$) because they can be expressed as the product of three rotation matrices as in Equation (2.15) where each of the basic rotation matrices (as well as their product) is close to $\mathbf{I}_{3\times3}$ because of the small angles. Hence, the diagonal elements which are close to one and the upper- and lower-diagonal elements which are close to zero are plotted separately in Figure 3.7(c) for better visualization. When $\mathbf{D}_n$ is close to $\mathbf{I}_{3\times3}$, the $\mathbf{q}_n^{\text{diff}}$ vectors calculated by using Equation (3.6) are close to $(1, 0, 0, 0)^T$, as observed in Figure 3.7(d).

3.4.1.2 Classification

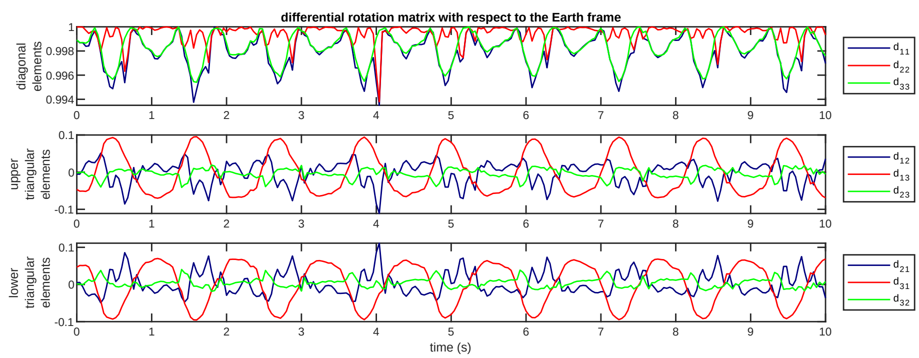
A procedure similar to that in [86, 87, 92] is followed for activity recognition. The sensor sequences are divided into 9120 (= 60 feature vectors per 5 min recording × 19 activities × 8 subjects) non-overlapping segments of 5-s duration each and transformed according to one of the seven approaches described in Section 3.4.1.1. Then, statistical features are extracted for each segment of each axis of each sensor type, as described in Section 2.3.2.1, resulting in a total of 26 features per segment



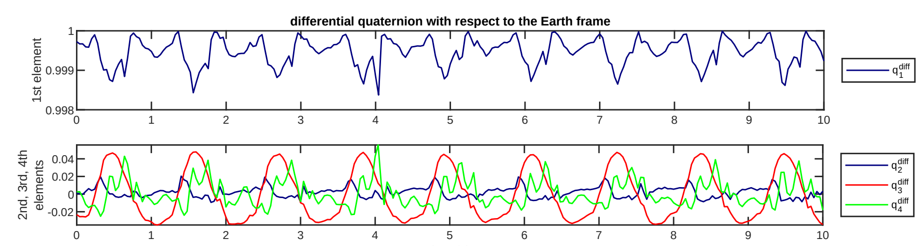
(a)



(b)



(c)



(d)

Figure 3.7: Original and orientation-invariant sequences from a walking activity plotted over time. (a) Original sensor sequences; (b) sensor sequences; elements of (c) the differential rotation matrix and (d) the differential quaternion. Sequences in (b)–(d) are represented in the Earth frame and are invariant to sensor orientation.

of each axis. For the reference approach that does not involve any transformation, there are 5 sensor units $\times$ 9 axes $\times$ 26 features per axis = 1170 features that are stacked to form a 1170-element feature vector for each segment. The number of axes as well as the number of features vary depending on the transformation technique; however, the total number of feature vectors is fixed (9120). For instance, in the Euclidean norm, there is a three-fold decrease in the number of axes and hence in the number of features. The features are normalized to the interval $[0, 1]$ over all the feature vectors for each subject.

The number of features is reduced through PCA, as in Section 2.3.2.1. This allows one to consider only a certain number of features that exhibit the largest variances to reduce the dimensionality. Thus, for each approach, the eigenvalues of the covariance matrix of the feature vectors are calculated, sorted in descending order, and plotted in Figure 3.8. Using the first 30 eigenvalues appears to be suitable for most of the approaches; hence, we reduce the dimensionality down to $F = 30$.

We perform activity classification with seven state-of-the-art classifiers that are briefly described below.

- **Support Vector Machines (SVM):** This classifier is described in Section 2.3.2.2. We optimize the SVM parameters in the same way as in Section 2.3.2.2 over all approaches and cross-validation techniques in this chapter. In the coarse grid, we get the same optimal parameter values as in Section 2.3.2.2; hence we use the same fine grid. The optimal parameter values in the fine grid are obtained as $(C^{**}, \gamma^{**}) = (5, 0.1)$, which are used throughout this chapter.
- **Artificial Neural Networks (ANN):** The ANN classifier, explained in Section 2.3.2.2, is used here with the same parameter selection method.
- **Bayesian Decision Making (BDM):** This classifier is explained in Section 2.3.2.2.
- **Linear Discriminant Classifier (LDC):** This classifier is the same as BDM except that the average of the covariance matrices individually

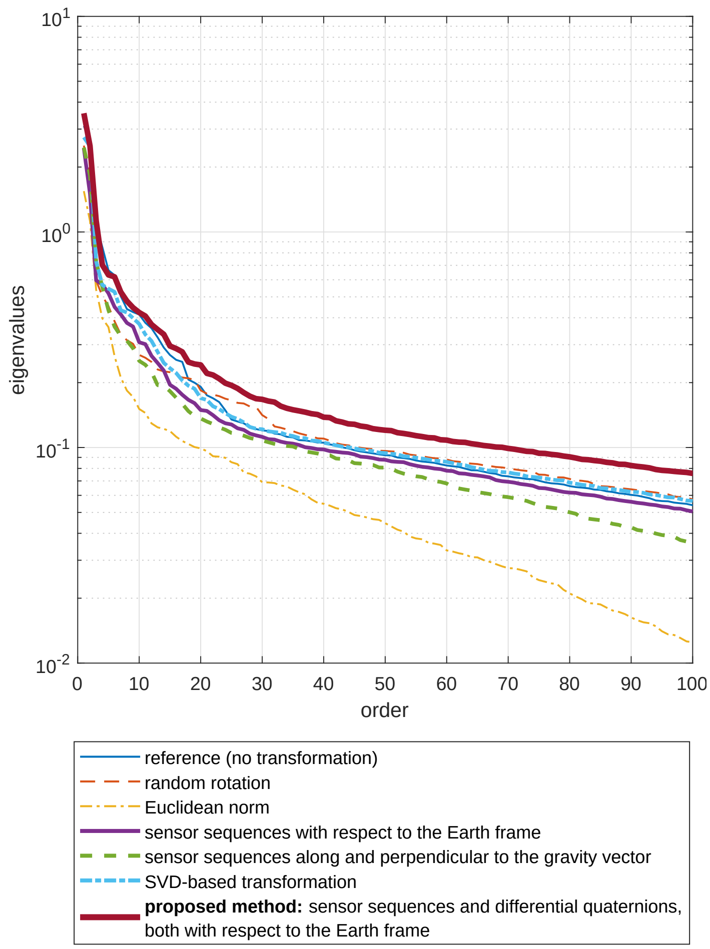


Figure 3.8: The first 100 eigenvalues of the covariance matrix of the feature vectors sorted in descending order, calculated based on the features extracted from the data transformed according to the seven approaches.

calculated for each class is used for all of the classes. Since the Gaussian distributions fitted to the different classes have different mean vectors but the same covariance matrix in this case, the classes have identical probability density functions centered at different points in the feature space. Hence, the classes are linearly separated from each other, and the decision boundaries in the feature space are hyperplanes [97].

- ***k*-Nearest Neighbor (*k*-NN):** The k-NN classifier, explained in Section 2.3.2.2, is used here with the same parameter selection as in that section.

- **Random Forest (RF):** A random forest classifier is a combination of multiple decision trees [120]. In the training phase, each decision tree is trained by randomly and independently sampling the training data. Normalized information gain is used as the splitting criterion at each node. In the classification phase, the decisions of the trees are combined by using majority voting. The number of decision trees is selected as 100 because we have observed that using a larger number of trees does not significantly improve the accuracy while increasing the computational cost considerably.

- **Orthogonal Matching Pursuit (OMP):** The training phase consists of only storing the training vectors with their class labels. In the classification phase, each test vector is represented as a linear combination of a very small portion of the training vectors with a bounded error, which is called the sparse representation. The vectors in the representation are selected iteratively by using the OMP algorithm [121] where an additional training vector is selected at each iteration. The algorithm terminates when the desired representation error level is reached, which is selected to be 10^{-3}. Then, a residual for each class is calculated as the representation error when the test vector is represented as a linear combination of the training vectors of only that class, and the class with the minimum residual error is selected.

3.4.1.3 Cross Validation

To determine the accuracies of the classifiers, L1O cross-validation technique is used, as explained in Section 2.3.2.3. Thus, in our implementation, the dataset is partitioned into eight and there are 1140 feature vectors in each partition.

3.4.2 Comparative Evaluation Based on Accuracy and Run Time

The activity recognition performance of the different data transformation techniques and classifiers is shown in Figure 3.9. In the figure, the lengths of the bars correspond to the classification accuracies and the thin horizontal sticks indicate plus/minus one standard deviation about the accuracies averaged over the cross-validation iterations.

In the lower part of Figure 3.9, the accuracy values averaged over the seven classifiers are also provided for each approach and compared with the reference case, as well as with the proposed method. Referring to this part of the figure, the standard system that we take as reference, with fixed sensor orientations, provides an average accuracy of 87.2%. When the sensor units are randomly oriented, the accuracy drops by 31.8% on average with respect to the standard reference case. This shows that the standard system is not robust to incorrectly or differently oriented sensor units. The existing methods for orientation invariance result in a more acceptable accuracy reduction compared to the reference case: The accuracy drop is 18.8% when the Euclidean norms of the tri-axial sensor sequences are taken, 12.5% when the sensor sequences are transformed to the Earth frame, 12.2% when the sensor sequences are represented along and perpendicular to the gravity vector, and 8.4% when the SVD-based transformation is applied.

Our approach that uses the sensor sequences together with differential quaternions, both with respect to the Earth frame, achieves an average accuracy of 82.5% over all activities with an average accuracy drop of only 4.7% compared to the reference case. Such a decrease in the accuracy is expected when the sensor units are allowed to be placed freely at arbitrary orientations because this flexibility entails the removal of fundamental information such as the direction of the gravity vector measured by the accelerometers and the direction of the Earth's magnetic field detected by the magnetometers. Hence, the average accuracy drop of 4.7% is considered to be acceptable when such information related to the sensor unit orientations is removed inevitably.

In the lower part of Figure 3.9, we also provide the improvement achieved by each method compared to the random rotation case which corresponds to the standard system using random sensor unit orientations. The method that we newly propose in this chapter performs the best among all the methods considered in this study when the sensor units are allowed to be attached at arbitrary orientations.

The activity recognition accuracy highly depends on the classifier. According to Figure 3.9, in almost all cases, the SVM classifier performs the best among the seven classifiers compared. SVM outperforms the other classifiers especially in approaches targeted to achieve orientation invariance where the classification problem is more challenging. The robustness of SVM in such non-ideal conditions is consistent with other studies [47, 87]. Besides the SVM classifier, ANN and LDC also obtain high classification accuracy. Although reference [69] states that k-NN has been shown to perform remarkably well in activity recognition, it is not the most accurate classifier that we have identified.

To observe the recognition rates of the individual activities, a confusion matrix associated with the SVM classifier is provided in Table 3.1 for the proposed method. It is apparent that the proposed transformation highly misclassifies the stationary activities A_1–A_4. These activities contain stationary postures, namely, sitting, standing, and two types of lying, which are misclassified probably because we remove the information about sensor unit orientation from the data. In particular, activity A_1 (sitting) is mostly misclassified and confused with activities A_3 (lying on back side) and A_7 (standing still in an elevator). The remaining stationary activities are also misclassified as A_7. Among the 15 non-stationary activities, activities A_{10} and A_{11} (walking on a treadmill in flat and 15° inclined position, respectively) are

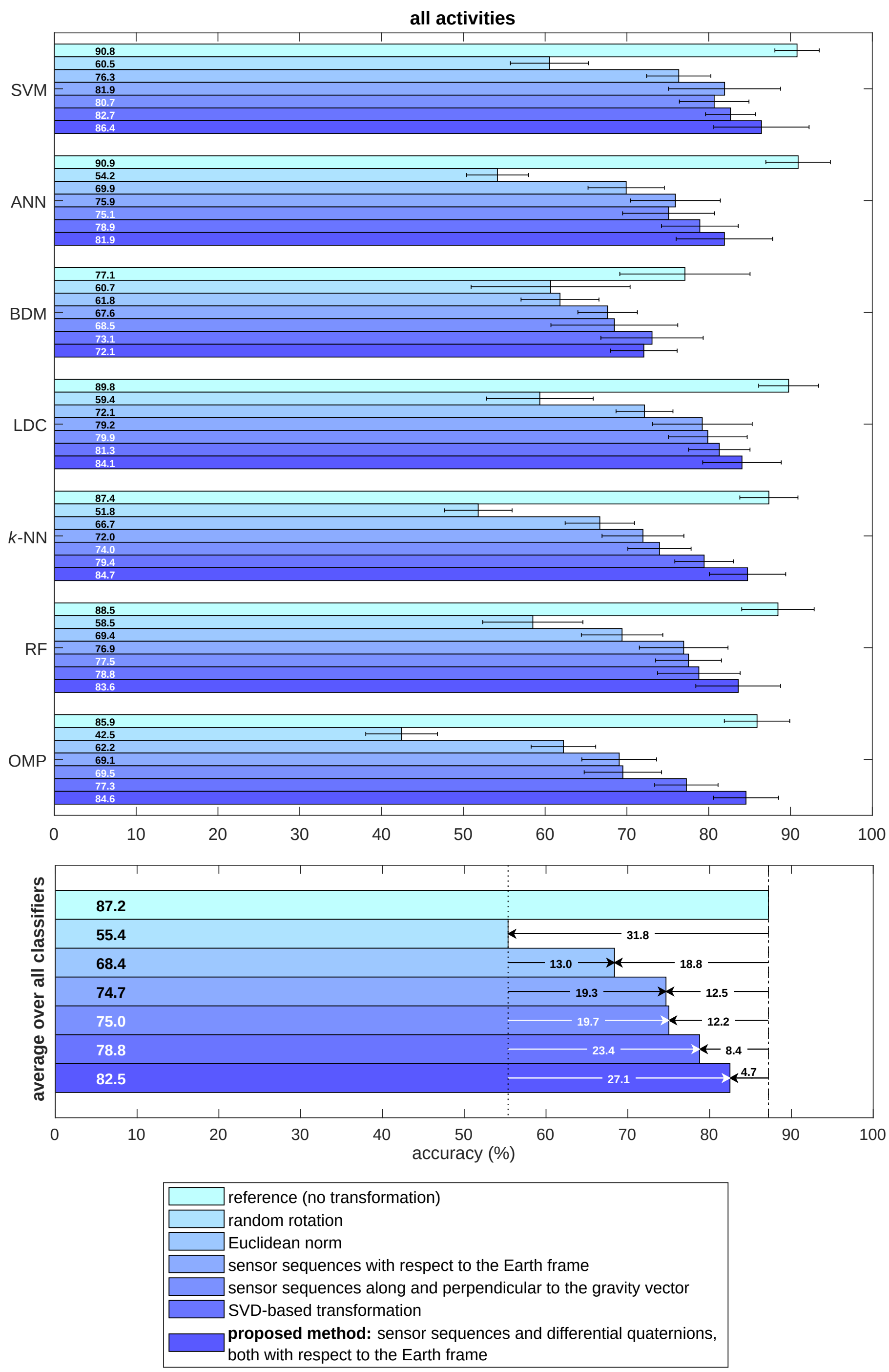


Figure 3.9: Activity recognition performance for all the data transformation techniques and classifiers over all activities. The lengths of the bars represent the accuracies and the thin horizontal sticks indicate plus/minus one standard deviation over the cross-validation iterations.

confused with each other because of the similarity between the body movements in the two activities. Other misclassifications occur between activity pairs that have similarities such as A_7/A_8, A_8/A_7, A_2/A_8, A_{18}/A_6, and A_{13}/A_9, although rarely. Activities A_{12} (running on a treadmill at a speed of 8 km/h) and A_{17} (rowing) are perfectly classified by SVM for the proposed method, probably because they are associated with unique body movements and do not resemble any of the other activities.

We present the classification performance separately for stationary and non-stationary activities in Figure 3.10. For each classifier and each approach, we calculate the accuracy values by averaging out the accuracies of the stationary activities (A_1–A_4) and non-stationary activities (A_5–A_{19}).

For stationary activities (see Figure 3.10(a)), an average accuracy of 81.2% is obtained for fixed sensor unit orientations. When the sensor units are oriented randomly, the average accuracy drops to 42.6%. The existing orientation-invariant methods exhibit accuracies between 31.7% and 62.2%, some of them being higher and some being lower than the accuracy for random rotation. The Euclidean norm method performs particularly poorly in this case. The proposed method achieves an average accuracy of 66.8%, which is considerably higher than random rotation and all the existing OITs. Although two of the existing transformations provide some improvement compared to the random rotation case, their accuracies are much lower than the standard reference system. Hence, removing the orientation information from the data makes it particularly difficult to classify stationary activities.

For non-stationary activities (see Figure 3.10(b)), the accuracy decreases from 88.8% to 58.8% on average when the sensor units are placed randomly and no transformation is applied. The existing orientation-invariant methods obtain accuracies ranging from 78.2% to 83.2%, which are comparable to the reference case with fixed sensor unit orientations. The method we propose obtains an average accuracy of 86.7%, which is higher than all the existing methods and only 2.1% lower than the reference case. This shows that when the sensor units are fixed to the body at arbitrary orientations, the proposed method can classify non-stationary activities with a performance similar to that of fixed sensor unit orientations. In the last two rows of the confusion matrix provided in Table 3.1, the average accuracy of the stationary activities (A_1–A_4) and non-stationary activities (A_5–A_{19}) are provided separately for the proposed method, again using the SVM classifier.

Referring to Figure 3.10(a), we observe that the recognition rate of stationary activities highly depends on the classifier. On average, the best classifier is LDC, probably because the recognition of stationary activities is quite challenging and the LDC classifier separates the classes from each other linearly and smoothly in the feature space. For the proposed method, the OMP classifier performs much better than the remaining six classifiers. On the other hand, for non-stationary activities (see Figure 3.10(b)), the classifiers obtain comparable accuracy values, unlike the case for stationary activities. In this case, SVM is the most accurate classifier, both on average and for the proposed method.

3.5 Run-Time Analysis

The average run times of the data transformation techniques per 5-s time segment are provided in Table 3.2. All the processing in this work was performed on 64-bit MATLAB® R2017b running on a laptop computer whose specifications are provided in Section 2.5. The proposed method has an average run time of about 61 ms per 5-s time segment and can be executed in near real time since the run time is much shorter than the duration of the time segment.

The run times of the classifiers are presented in Table 3.3 for each of the seven data transformation techniques. Table 3.3(a) contains the total run times of the classifiers for an average cross-validation iteration, including the training phase and classification of all the test feature vectors. We observe that k-NN, LDC, and BDM are much faster than the other classifiers for all of the data transformation techniques. Table 3.3(b) contains the average training times of the classifiers for a single cross-validation iteration. The k-NN and OMP classifiers only store the training feature vectors in the training phase; therefore, their training time is negligible. Among the remaining classifiers, training of BDM is the fastest.

Table 3.1: Confusion matrix of the SVM classifier for the proposed method over all activities.

estimated	true labels																			
labels	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	A_{12}	A_{13}	A_{14}	A_{15}	A_{16}	A_{17}	A_{18}	A_{19}	**total**
A_1	286	1	68	20	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	383
A_2	0	330	0	0	0	0	26	1	0	0	0	0	0	0	0	0	0	0	0	357
A_3	81	0	372	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	453
A_4	1	0	0	367	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	368
A_5	0	0	0	0	477	0	0	2	11	0	12	0	0	0	0	0	0	0	1	503
A_6	0	0	0	0	0	453	2	5	0	0	0	0	6	0	0	0	0	42	0	508
A_7	97	102	33	83	0	0	354	61	0	0	0	0	0	0	0	0	0	0	0	730
A_8	15	47	6	10	1	27	90	409	1	0	1	0	9	2	2	0	0	0	8	628
A_9	0	0	0	0	2	0	0	1	416	19	4	0	36	0	0	1	0	0	0	479
A_{10}	0	0	0	0	0	0	0	0	13	354	84	0	0	0	0	0	0	0	0	451
A_{11}	0	0	0	0	0	0	0	0	38	105	374	0	11	0	0	0	0	0	0	528
A_{12}	0	0	0	0	0	0	0	0	0	0	0	480	0	0	0	0	0	0	0	480
A_{13}	0	0	0	0	0	0	0	1	1	2	5	0	399	7	0	0	0	0	1	416
A_{14}	0	0	0	0	0	0	0	0	0	0	0	0	19	471	0	0	0	0	0	490
A_{15}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	478	1	0	0	0	479
A_{16}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	477	0	0	0	477
A_{17}	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	480	0	1	482
A_{18}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	438	0	439
A_{19}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	469	469
total	480	480	480	480	480	480	480	480	480	480	480	480	480	480	480	480	480	480	480	9120
accuracy	59.6	68.8	77.5	76.5	99.4	94.4	73.8	85.2	86.7	73.8	77.9	100.0	83.1	98.1	99.6	99.4	100.0	91.3	97.7	**86.5**
(%)	**70.6** (for stationary activities)				**90.7** (for non-stationary activities)															(overall)

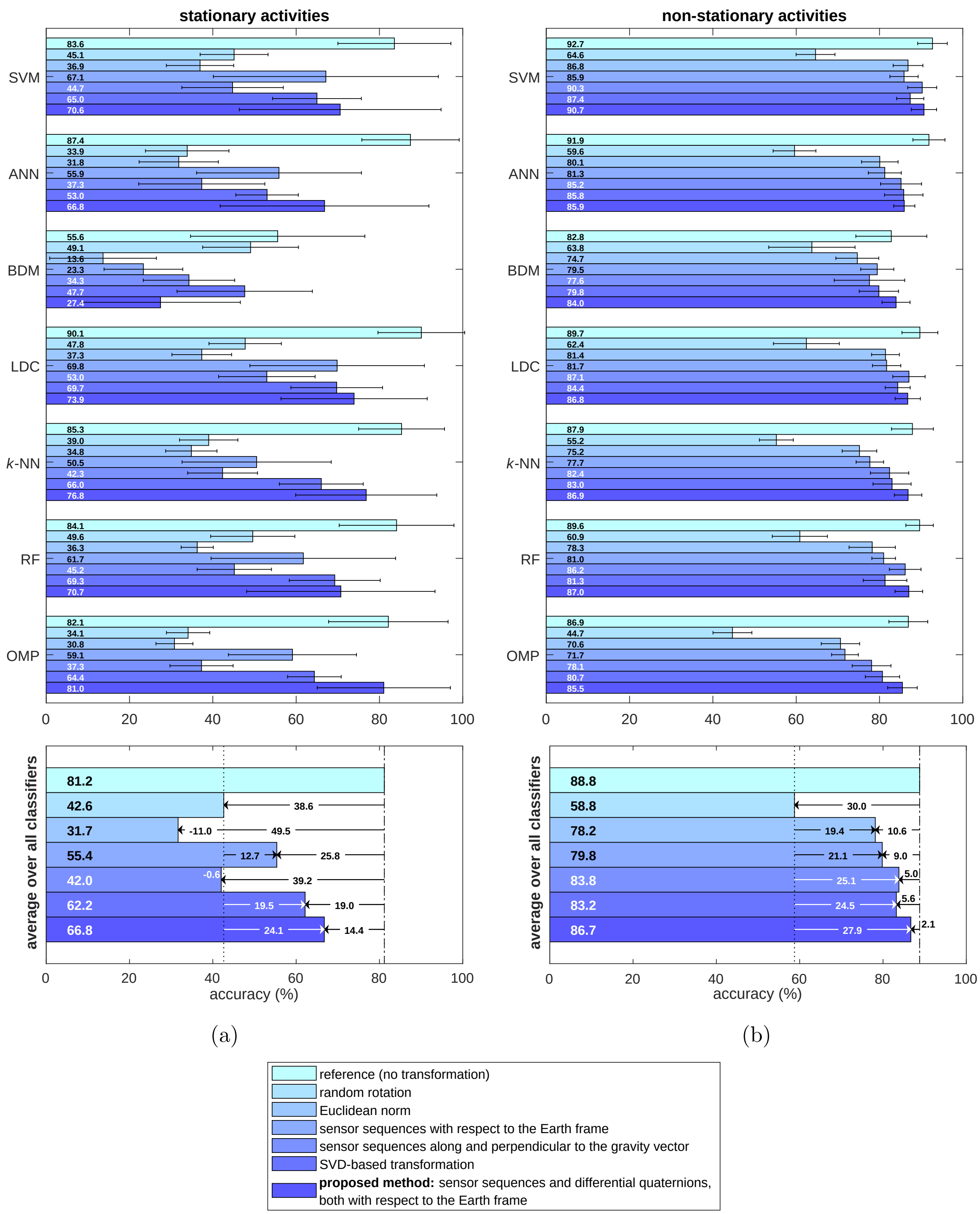


Figure 3.10: Activity recognition performance for all the data transformation techniques and classifiers for (a) stationary and (b) non-stationary activities. The lengths of the bars represent the accuracies and the thin horizontal sticks indicate plus/minus one standard deviation over the cross-validation iterations.

Table 3.2: Average run times of the data transformation techniques per 5-s time segment.

data transformation technique	**run time (ms)**
Euclidean norm	0.69
sensor sequences with respect to the Earth frame	56.25
sensor sequences along and perpendicular to the gravity vector	1.09
SVD-based transformation	8.94
proposed method: sensor sequences and differential quaternions, both with respect to the Earth frame	61.08

Table 3.3(c) contains the average classification time of a single test feature vector, extracted from a segment of 5-s duration. ANN and LDC are about an order of magnitude faster than the others in classification. The classification time of OMP is the largest. Note that, because of programming overheads, the total classification times provided in Table 3.3(a) are greater than the sum of the training and classification times (Table 3.3(b,c), respectively) multiplied by 1140 (the number of feature vectors per L1O iteration).

3.6 Discussion

Overall, the recognition rates of non-stationary activities are considerably better than those of stationary ones for all the approaches considered in this study. This is because in non-stationary activities, the activity type is encoded in the body motion whereas in stationary activities, since there is no significant body motion, the removal of sensor unit orientation information to achieve orientation invariance has a major impact on the accuracy. The classification of stationary activities is a more challenging problem and it is clear that sensor unit orientations provide essential information for this purpose.

The direction of the gravity vector measured by the accelerometer and the direction of the magnetic field vector determined by the magnetometer provide essential information about the orientation of the sensor unit. When the sensor sequences are represented with respect to the Earth frame to achieve orientation

Table 3.3: (a) Total run time (including training and classification of all test feature vectors) and (b) training time in an average L1O iteration; (c) average classification time of a single test feature vector.

	classifier	**reference (no transformation)**	**random rotation**	**Euclidean norm**	**sensor sequences with respect to the Earth frame**	**sensor sequences along and perpendicular to the gravity vector**	**SVD-based transformation**	**proposed method: sensor sequences and differential quaternions, both with respect to the Earth frame**
	SVM	6.42	14.20	7.22	11.71	8.19	6.24	10.05
	ANN	7.37	8.49	8.54	6.58	12.04	7.91	6.14
(a)	BDM	1.67	1.61	1.59	1.55	2.12	1.48	1.69
total run time	LDC	1.10	0.87	0.84	1.52	0.84	0.93	1.51
(s)	k-NN	0.24	0.12	0.12	0.21	0.19	0.12	0.22
	RF	16.81	22.51	26.40	24.34	19.05	19.71	23.98
	OMP	1018.27	798.90	92.32	99.41	96.48	75.18	114.68
	SVM	6.01	13.39	6.61	10.31	7.58	5.36	8.60
	ANN	7.35	8.47	8.52	6.57	12.01	7.89	6.12
(b)	BDM	0.01	0.01	0.01	0.01	0.01	0.01	0.01
training time	LDC	0.33	0.23	0.22	0.38	0.22	0.26	0.33
(s)	k-NN	–	–	–	–	–	–	–
	RF	15.20	20.90	24.11	21.75	17.45	17.87	21.25
	OMP	–	–	–	–	–	–	–
	SVM	0.26	0.60	0.42	0.39	0.40	0.24	0.31
	ANN	0.02	0.02	0.01	0.01	0.02	0.01	0.01
(c)	BDM	1.46	1.41	1.39	1.35	1.85	1.29	1.47
classification time	LDC	0.04	0.03	0.03	0.05	0.03	0.03	0.04
(ms)	k-NN	0.21	0.11	0.11	0.19	0.16	0.11	0.19
	RF	0.71	0.73	0.99	0.83	0.72	0.74	0.87
	OMP	892.55	700.17	80.55	86.38	84.20	65.43	99.69

invariance, this information is lost because the gravity and the magnetic field of the Earth are roughly in the fixed z_E and x_E directions of the Earth frame, respectively. Hence, in our proposed method, we incorporate the change in the sensor unit orientation over time by calculating differential quaternions with respect to the Earth, which represent the rotation between consecutive time samples invariantly to the sensor unit orientation. The use of differential quaternions increases the accuracy considerably because they effectively represent the rotational motion of the sensor unit related to the activities. When the rotational transformation is represented with respect to the Earth frame, it is invariant to sensor unit orientation, as desired.

For all the methods compared in this chapter, we use the same dataset which was acquired by placing the sensor units on the body at fixed orientations. This enables us to make a fair comparison between all of the seven approaches considered in this work. In the random rotation case, we rotate the data arbitrarily for each time segment and each sensor unit; hence, we obtain new data that simulate random sensor unit orientations and match exactly the same level of difficulty of the original data except for the rotational difference. In the last five approaches that correspond to orientation-invariant methods, it is mathematically guaranteed that the transformed data are exactly invariant to sensor unit orientations; hence, they can be directly compared with the reference and random rotation cases. Had we recorded an additional dataset with different sensor unit orientations, we would not be able to fairly compare the accuracies obtained with the two datasets because it is not possible to guarantee the same level of difficulty in activity recognition in different experiments. This fact can be observed even within the current dataset from the non-negligible standard deviations in the activity recognition accuracy over the cross-validation iterations (see Figures 3.9 and 3.10). This shows that the variation among the subjects is significant, as also observed in [17].

3.7 Concluding Remarks

In this chapter, we have demonstrated that the standard activity recognition paradigm cannot handle incorrectly or differently oriented sensor units when the position remains fixed. To overcome this problem, we have proposed a transformation that we apply on the sensor data at the pre-processing stage to increase the robustness of the system to errors in the orientations at which the sensor units are attached to the body. The method we have proposed extracts the activity-related information from the sensor sequences while removing the information associated with the absolute sensor unit orientations. This way, we ensure that the transformed sequences do not depend on the absolute sensor unit orientations. The transformed sequences have the same form as the original sequences except the number of axes, which enables us to apply this method in the pre-processing stage of any system that can handle multi-axial data, including systems that directly use time-domain data in its raw form as well as those that use extracted features. We have shown that our method significantly reduces the accuracy degradation caused by incorrect/different sensor unit orientations. The proposed method performs substantially better than the existing methods developed specifically for this problem and achieves nearly the same accuracy level as the fixed orientation case for non-stationary activities. The transformation we propose can be computed in a time much shorter than the duration of one segment of the data, therefore, it can be efficiently implemented and used in near real time.

Chapter 4

Novel Non-Iterative Orientation Estimation Method for Wearable Motion Sensor Units

In Chapter 3, we have employed an existing OEM based on GN [107]. In this chapter, we propose a novel non-iterative OEM for wearable motion sensor units acquiring accelerometer, gyroscope, and magnetometer measurements. We integrate the proposed method into the OIT proposed in Chapter 3. The overview of the method is shown in Figure 4.1.

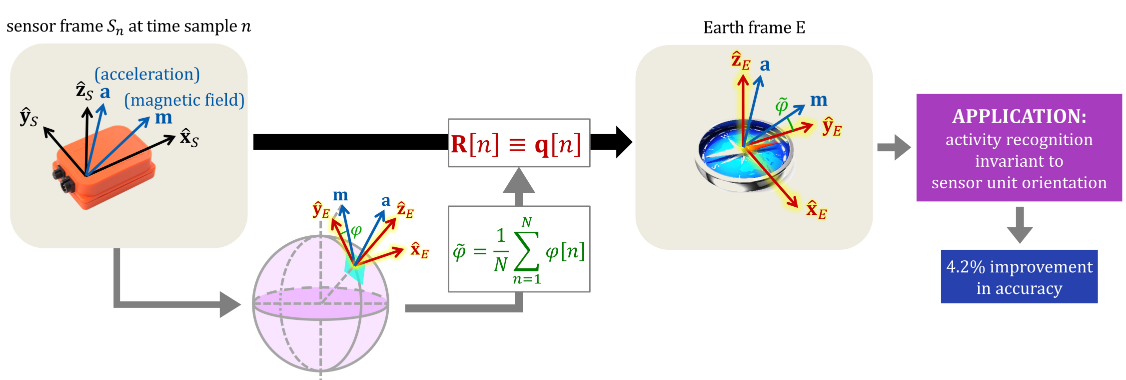


Figure 4.1: An overview of the proposed OEM.

4.1 Notation and Representation of Sensor Unit Orientation

Data acquired from the accelerometer, gyroscope, and magnetometer at time sample n are represented by 3×1 vectors $\mathbf{a}[n], \boldsymbol{\omega}[n]$, and $\mathbf{m}[n]$, respectively. For a time segment of recorded data that contains N time samples, the discrete-time index n takes values between 1 and N, and is omitted for simplicity where needed. The hat notation is used for vectors normalized by their magnitudes (unit vectors): $\hat{\mathbf{a}} \triangleq \mathbf{a}/\|\mathbf{a}\|$ and $\hat{\mathbf{m}} \triangleq \mathbf{m}/\|\mathbf{m}\|$.

According to the East-North-Up (ENU) convention,[1] the x, y, z axes of the Earth's coordinate frame E point in the East, North, and upward directions, respectively (Figure 4.2). The transformation between E and the sensor frame S_n at the time sample n can be represented by a 3×3 rotation matrix $\mathbf{R}[n]$ or equivalently by a 4×1 quaternion $\mathbf{q}[n]$ [119].[2] The columns of $\mathbf{R}[n]$ correspond to the unit vectors $\hat{\mathbf{x}}_E$, $\hat{\mathbf{y}}_E$, $\hat{\mathbf{z}}_E$ of frame E with respect to S_n. Note that the transpose of $\mathbf{R}[n]$ (which is the same as its inverse since $\mathbf{R}[n]$ is an orthonormal matrix), represents the orientation of the sensor frame S_n with respect to the Earth frame E.

4.2 Proposed Methodology to Estimate Sensor Unit Orientation

Given the current angular rate vector, $\boldsymbol{\omega}[n] = (\omega_x[n],\ \omega_y[n],\ \omega_z[n])^T$, the dynamic orientation $\mathbf{q}_d[n]$ for $n = 1, \ldots, N$ is estimated based on the combined orientation estimate $\mathbf{q}[n-1]$ at the previous time sample [see Equation (4.6)] and the

[1]For the Earth frame, we have used the NED convention in Chapter 3, as in the original implementation of GN-based OEM (see [107] and the Appendix), whereas we use the ENU convention in this chapter. The estimated sensor unit orientations for the two conventions are related to each other by a fixed coordinate transformation. Both conventions yield exactly the same activity recognition results.

[2]The matrix $\mathbf{R}[n]$ in this chapter is the same as the matrix $\mathbf{R}_E^{S_n}$ in Chapter 3.

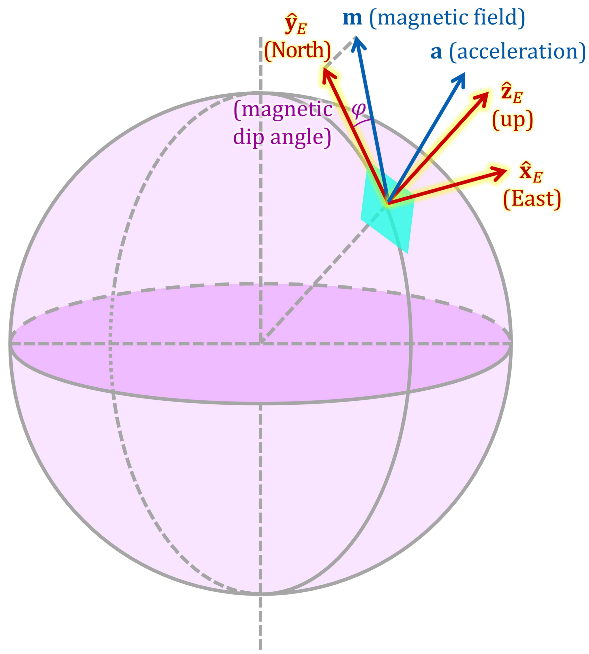


Figure 4.2: The Earth frame illustrated on an Earth model illustrating the unit vectors of the Earth frame, the two reference vectors $\mathbf{a}$ and $\mathbf{m}$, and the magnetic dip angle φ.

augmented angular rate vector $\boldsymbol{\omega}'[n] \triangleq (0,\ \omega_x[n],\ \omega_y[n],\ \omega_z[n])^T$ as

$$\mathbf{q}_d[n] = \mathbf{q}[n-1] + \Delta t \left(\frac{1}{2} \mathbf{q}[n-1] \otimes \boldsymbol{\omega}'[n] \right) \tag{4.1}$$

where the symbol $\otimes$ denotes the quaternion product operator and Δt is the time step.

Assuming that the acceleration components resulting from the motion of the sensor unit average out to zero, gravity stands as the dominant component of $\mathbf{a}$ in the long term. Consequently, averaging the acquired acceleration vectors provides an estimate of the direction of the gravity vector which points in the vertical direction of the Earth. Based on this assumption that the average of the $\mathbf{a}$ vectors points to the vertical, we can estimate the magnetic dip angle φ by averaging the angle between $\mathbf{m}[n]$ and the horizontal plane (perpendicular to $\mathbf{a}[n]$) over a short time segment:

$$\tilde{\varphi} = \frac{1}{N}\sum_{n=1}^{N} \varphi[n] \quad \text{where} \quad \varphi[n] = \frac{\pi}{2} - \angle\big(\mathbf{a}[n], \mathbf{m}[n]\big) \tag{4.2}$$

Here, $\angle(\cdot, \cdot) \in [0, \pi)$ denotes the angle between two 3D vectors and N is the number of time samples over a short time segment.

If $\tilde{\varphi}$ were zero, then we could have taken the upward ($\hat{\mathbf{z}}_E$) and the North ($\hat{\mathbf{y}}_E$) axes in the same direction as the detected $\mathbf{a}$ and $\mathbf{m}$ vectors, respectively, as in existing work. Since this is not the case in general, we select $\hat{\mathbf{z}}_E$ and $\hat{\mathbf{y}}_E$ orthogonally to simultaneously meet the two objectives,

O1: $\hat{\mathbf{z}}_E$ is as close as possible to $\hat{\mathbf{a}}$ and

O2: the angle between $\hat{\mathbf{y}}_E$ and $\hat{\mathbf{m}}$ is as close as possible to $\tilde{\varphi}$.

We geometrically determine $\hat{\mathbf{z}}_E$ and $\hat{\mathbf{y}}_E$ to satisfy the objectives O1 and O2 directly without the use of any iterative OEMs such as GD or GN as follows:

To satisfy O1 only, the up and North directions ($\hat{\mathbf{z}}_E$ and $\hat{\mathbf{y}}_E$) can be taken as the $\hat{\mathbf{a}}$ vector and the normalized component $\hat{\mathbf{m}}_\perp$ of $\mathbf{m}$ perpendicular to $\mathbf{a}$, respectively, as in the TRIAD algorithm [109]:

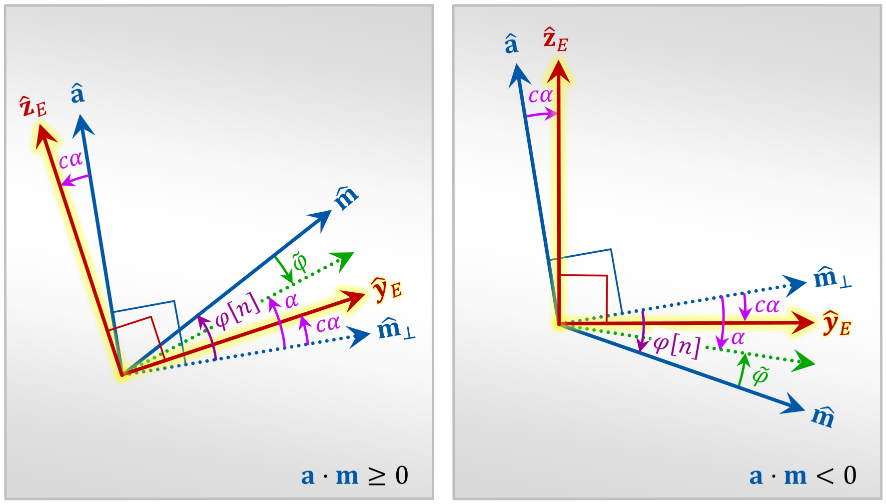
$$\hat{\mathbf{m}}_\perp = \frac{\mathbf{m}_\perp}{\|\mathbf{m}_\perp\|} \quad \text{where} \quad \mathbf{m}_\perp = \mathbf{m} - (\hat{\mathbf{a}} \cdot \mathbf{m})\,\hat{\mathbf{a}} \tag{4.3}$$

To satisfy O2 only, we may rotate the $\hat{\mathbf{z}}_E$ and $\hat{\mathbf{y}}_E$ axes on the **a-m** plane about the axis $\hat{\mathbf{m}} \times \hat{\mathbf{a}}$ by the angle

$$\alpha = \operatorname{sign}(\mathbf{a} \cdot \mathbf{m})\,(\angle(\mathbf{m}, \mathbf{m}_\perp) - |\tilde{\varphi}|) = \operatorname{sign}(\mathbf{a} \cdot \mathbf{m})\,(\varphi[n] - |\tilde{\varphi}|) \tag{4.4}$$

where $\operatorname{sign}(\cdot)$ denotes the signum function. This rotation is depicted in Figure 4.3 for the two cases.

Since the objectives O1 and O2 cannot be satisfied at the same time (unless $\tilde{\varphi} = 0$), we consider a solution which tries to meet both objectives simultaneously by rotating the vectors $\hat{\mathbf{a}}$ and $\hat{\mathbf{m}}_\perp$ through an angle $c\alpha$, where $c \in [0, 1]$ is a parameter



(a) (b)

Figure 4.3: Selection of the $\hat{\mathbf{z}}_E$ and $\hat{\mathbf{y}}_E$ axes to estimate the static orientation for the cases where (a) $\mathbf{a} \cdot \mathbf{m} \geq 0$ and (b) $\mathbf{a} \cdot \mathbf{m} < 0$.

of the algorithm. Then,

$$\begin{aligned} \hat{\mathbf{z}}_E &= \hat{\mathbf{a}} \cos(c\alpha) - \hat{\mathbf{m}}_\perp \sin(c\alpha) \\ \hat{\mathbf{y}}_E &= \hat{\mathbf{a}} \sin(c\alpha) + \hat{\mathbf{m}}_\perp \cos(c\alpha) \end{aligned} \tag{4.5}$$

We select the remaining axis that points to the East as $\hat{\mathbf{x}}_E = \hat{\mathbf{y}}_E \times \hat{\mathbf{z}}_E$ and represent the static orientation estimate by the quaternion $\mathbf{q}_s[n]$ corresponding to the rotational transformation $\mathbf{R}_s[n] = [\hat{\mathbf{x}}_E \quad \hat{\mathbf{y}}_E \quad \hat{\mathbf{z}}_E]$.

We finally merge the dynamic and static estimates through weighted averaging to obtain the combined orientation estimate:

$$\mathbf{q}[n] = \mathcal{K}\mathbf{q}_d[n] + (1 - \mathcal{K})\, \mathbf{q}_s[n] \tag{4.6}$$

where $\mathcal{K} \in [0, 1]$ is the weight parameter of the algorithm. The flowchart of the algorithm is shown in Figure 4.4.

We optimize the parameters c and $\mathcal{K}$ of the newly proposed OEM through a 2D grid search to maximize classification accuracy. On a coarse grid where both parameters vary between zero and one with 0.1 increments, the optimal values

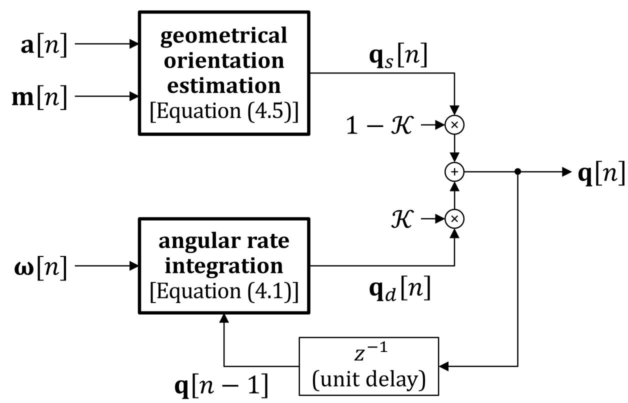


Figure 4.4: The flowchart of the proposed algorithm.

are $(c^*, \mathcal{K}^*) = (0.40, 0.98)$. On a fine grid where $c \in \{0.30, 0.32, \ldots, 0.80\}$ and $\mathcal{K} \in \{0.90, 0.91, \ldots, 1.00\}$, the optimal parameter pair is $(c^{**}, \mathcal{K}^{**}) = (0.36, 0.98)$, which is the parameter pair used in this study. When c and $\mathcal{K}$ are both set equal to zero, the proposed OEM reduces to the TRIAD algorithm.

4.3 Implementation of Existing OEMs and Initialization

We implement the existing iterative OEMs as follows: For KF-based OEM, we use the function `ahrsfilter` that is available in the Sensor Fusion and Tracking Toolbox of MATLAB R2018b [122]. This method relies on an indirect complementary KF model. The term complementary indicates that the KF balances orientation estimates coming from (i) the accelerometer and magnetometer and (ii) from the gyroscope [123]. The term indirect indicates that the KF operates on the error vector rather than the state vector itself [123]. The error process is modeled through the 12×1 state vector

$$\mathbf{x}_{\epsilon,k} = \begin{bmatrix} \boldsymbol{\theta}_{\epsilon,k} \\ \boldsymbol{\omega}_{\epsilon,k} \\ \mathbf{a}_{\epsilon,k} \\ \mathbf{m}_{\epsilon,k} \end{bmatrix} \tag{4.7}$$

where $\boldsymbol{\theta}_{\epsilon,k}$ is the 3×1 orientation error vector, $\boldsymbol{\omega}_{\epsilon,k}$ is the 3×1 gyroscope offset vector, $\mathbf{a}_{\epsilon,k}$ is the 3×1 acceleration error vector measured in the sensor frame, and $\mathbf{m}_{\epsilon,k}$ is the 3×1 magnetic disturbance error vector measured in the sensor frame, all measured at iteration k [123]. The 6×1 observation vector is defined as

$$\mathbf{z}_{\epsilon,k} = \begin{bmatrix} \mathbf{g}_{d,k} - \mathbf{g}_{s,k} \\ \mathbf{m}_{d,k} - \mathbf{m}_{s,k} \end{bmatrix} \tag{4.8}$$

where $\mathbf{g}_{d,k}$ and $\mathbf{g}_{s,k}$ are the dynamic and static estimates of the gravity vector at iteration k, whereas $\mathbf{m}_{d,k}$ and $\mathbf{m}_{s,k}$ are their counterparts for the magnetic field of the Earth [123]. The state and observation equations are expressed as

$$\mathbf{x}_k = \mathbf{A}_k \mathbf{x}_{k-1} + \mathbf{w}_k \tag{4.9}$$

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k \tag{4.10}$$

where $\mathbf{w}_k$ and $\mathbf{v}_k$ are additive noise vectors, $\mathbf{A}_k = \mathbf{0}$, and $\mathbf{H}_k$ is a matrix calculated based on the dynamic orientation estimate [123]. In this way, the static and dynamic estimates are adaptively combined, as in the remaining orientation estimation techniques (other than TRIAD). The noise variances of the accelerometer, gyroscope, and magnetometer sensors are provided by the manufacturer as $0.0110\ (\mathrm{m/s^2})^2$, $9.6328 \times 10^{-5}\ (\mathrm{rad/s})^2$, and $0.01581\ (\mu\mathrm{T/s})^2$, respectively [88]. We optimized these input parameters through a 3D grid search where we multiplied each by the factors 0.25, 0.5, 1, 2, or 4 and considered all $5^3 = 125$ combinations of these parameters. The highest accuracy is obtained where the values provided by the manufacturer are multiplied by 0.25, 0.25, and 4, respectively.

In the GD-based OEM, we use a single, approximated GD iteration at each time sample, as in its original implementation [111]. We implement the GN and LM algorithms without imposing any limit to the number of iterations, and terminate them when the change in the cost function is smaller than 10^{-3}. For LM, we use the algorithm provided on page 438 in [112]: We initialize the damping parameter with 0.5 for the first iteration and adaptively change it by a multiplicative factor of two in the iterations that follow.

We initialize the iterative and proposed OEMs as follows: Because of the dependence of the dynamic orientation estimate $\mathbf{q}_d[n]$ on the combined estimate $\mathbf{q}[n-1]$ at the previous time sample and since such a combined estimate is not available at the first time sample, Equation (4.1) is not evaluated for $n = 1$. Thus, the combined estimate in Equation (4.6) at $n = 1$ is calculated solely based on the static estimate without using the dynamic estimate: $\mathbf{q}[1] = \mathbf{q}_s[1]$. The iterative methods (GD, GN, and LM) are executed at each time sample n to estimate the static orientation $\mathbf{q}_s[n]$. Since there is no information about the orientation at $n = 1$, they are initialized with the quaternion estimated by the TRIAD algorithm at the first time sample. The KF is also initialized with an algorithm that is equivalent to TRIAD. (Note that both TRIAD and the proposed algorithm can already make an orientation estimate at the very first time sample.) For $n = 2, \ldots, N$, the combined orientation estimate $\mathbf{q}[n-1]$ at the previous time sample is used as the initial condition [107, 111]. We apply the OEMs to each time segment (5-s duration) of the recorded data separately.

4.4 Comparative Evaluation of the Proposed and Existing OEMs

In Chapter 3, we proposed a methodology for recognizing daily and sports activities that requires accurate sensor unit orientation estimates to allow the units to be worn on the body at any orientation. In that chapter, we employed the GN algorithm [107] to estimate the orientation of the sensor units. Here, we demonstrate that the activity recognition accuracy can be considerably improved by only replacing the GN algorithm with the newly proposed OEM.

We use the publicly available dataset acquired by our research group, comprised of 19 daily and sports activities [80, 81, 86]. The dataset (referred as dataset A in

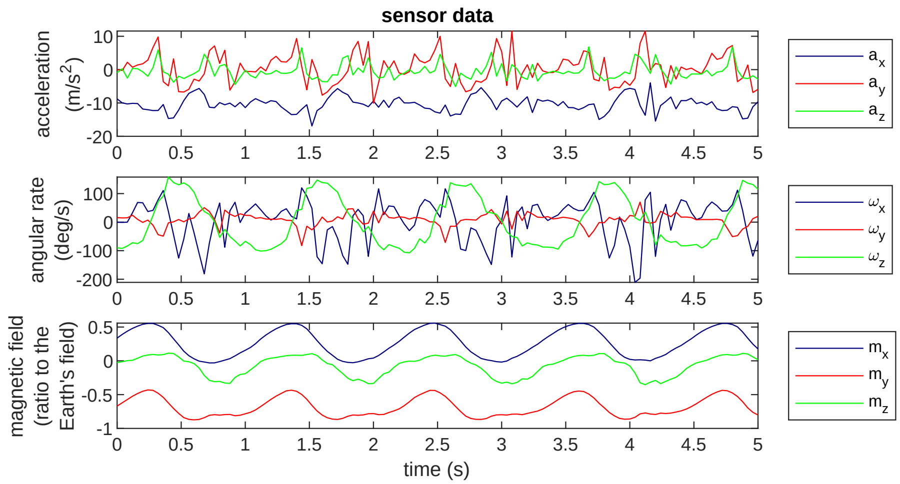
Section 2.3.1) is described in Section 3.4 and its attributes are provided in the second column of Table 2.1[3].

Figure 4.5(a) shows the data acquired from the sensor unit on the right leg of a subject during the activity of walking on a flat treadmill. The estimated elements of the quaternions $\mathbf{q}[n]$ representing the sensor unit orientations using the existing and proposed algorithms are plotted as a function of time in part (b) of the same figure.

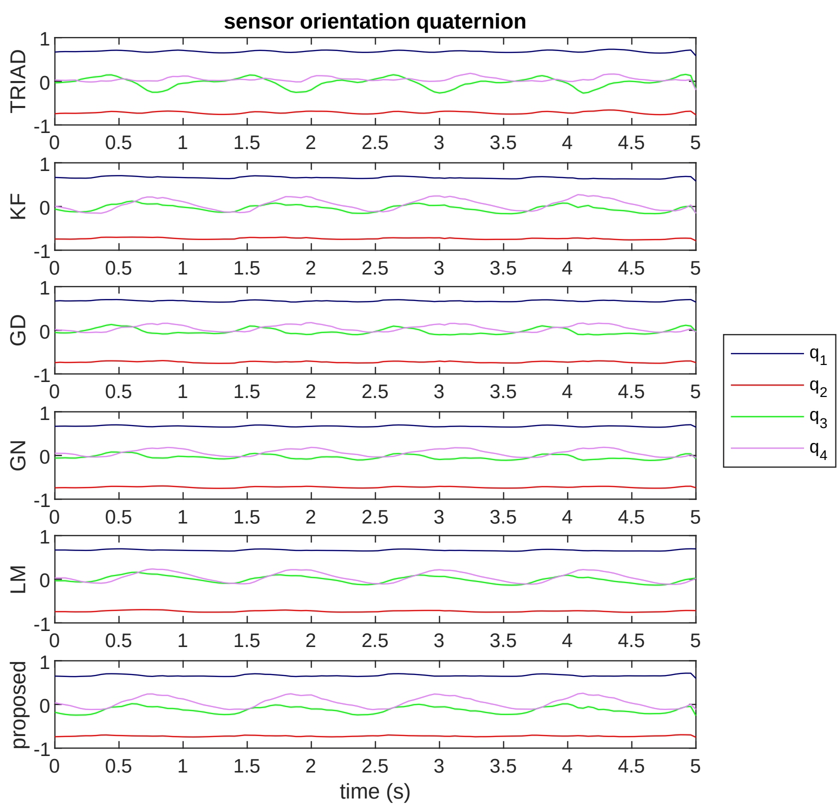
We have implemented eight approaches: The REF method is the standard activity recognition scheme with sensor units fixed to the body at proper orientations and does not transform the acquired data in any way (see Section 3.4.1.1). In ROT, we simulate arbitrarily oriented sensor units by randomly rotating the acquired data vectors through a rotational transformation, independently generated for each time segment of each sensor unit, as explained in the random rotation approach in Section 3.4.1.1. The OIT approach allows the units to be fixed to the body at any orientation by representing the acquired data in frame E together with the use of differential quaternions, as in the proposed approach that is described in Section 3.4.1.1. The OIT requires accurate estimation of sensor unit orientation. In this chapter, we estimate the sensor unit orientation by using five existing OEMs (TRIAD [109], KF [122], GD [111], GN (Appendix A and [107]), LM [112]) and the non-iterative method that we propose here. The six variations of the OIT using these OEMs are respectively denoted by OIT-TRIAD, OIT-KF, OIT-GD, OIT-GN, OIT-LM, and OIT-proposed.

Next, we follow the activity recognition scheme that is explained in Section 3.4.1, which involves the basic stages of feature extraction, feature reduction, feature normalization, and classification of the (transformed) data. For the OIT approach described in Section 3.4.1.1, 13 axes are used instead of nine axes of the raw sensor data (see Section 3.4.1.2), and thus, there exist 1,690 features instead of 1,170. We reduce the total number of features from 1,170 to 30 for REF and ROT and from 1,690 to 30 for OIT through the use of PCA, as explained in Section 2.3.2.1.

[3]The remaining datasets are not used in this chapter since they do not include data from a magnetometer.



(a)



(b)

Figure 4.5: (a) Original sensor data and (b) the estimated elements of the orientation quaternions plotted as a function of time.

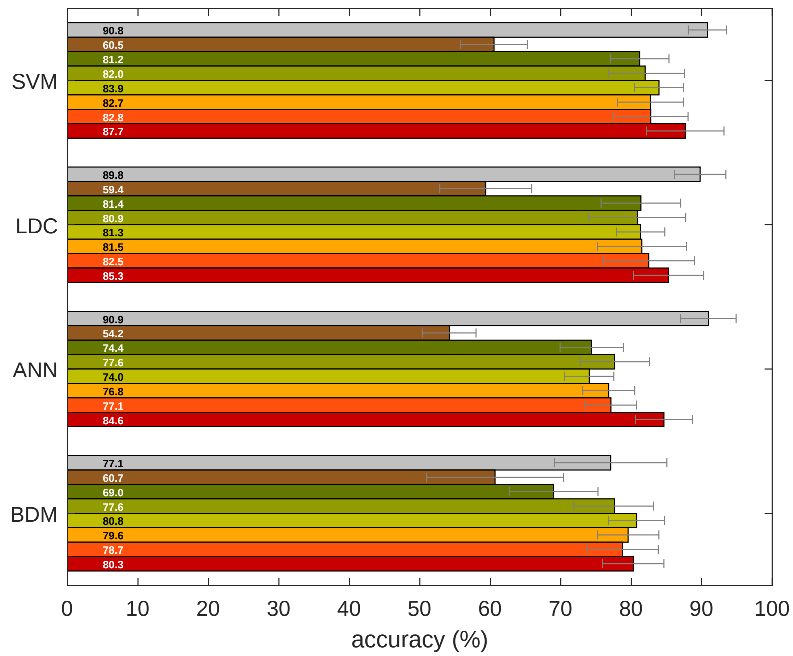
In Section 3.4.2, we have considered seven classifiers among which SVM usually showed outstanding performance, followed by LDC, ANN, and BDM. In this part of the thesis, we limit the number of classifiers to these best-performing four, select their parameters as in Section 3.4.2, and evaluate their accuracies through L1O cross validation that is explained in Section 2.3.2.3.

Activity recognition accuracies for the eight approaches that use the four selected classifiers are provided in Figure 4.6. As expected, the highest accuracy is obtained with REF that uses properly oriented sensor units and the lowest with ROT where the units are randomly oriented without the use of any OIT. All six OEMs, when integrated into the OIT, improve the accuracy compared to ROT. However, the proposed OEM is superior to the other five, achieving an average accuracy 8.0%, 5.0%, 4.5%, 4.3%, and 4.2% higher than OIT-TRIAD, OIT-KF, OIT-GD, OIT-GN, and OIT-LM, respectively (Figure 4.6(b)). Compared to REF, the average accuracy of OIT-proposed is 2.6% lower, which is naturally expected. The thin horizontal sticks in both parts of the figure indicate plus/minus one standard deviation over the cross-validation iterations and the classifiers, respectively.

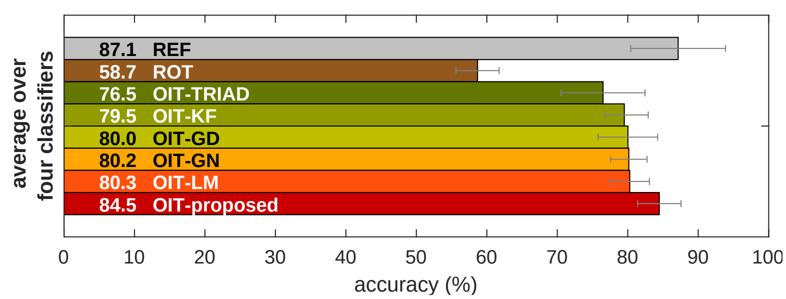
Referring to Figure 4.6(a), SVM usually performs the best among the four classifiers, demonstrating its robustness to variations in the data. For all six variations of the OIT, it achieves an accuracy noticeably higher than the remaining classifiers. LDC is the second best classifier on the average. The BDM classifier when used with OIT-KF, OIT-GD, OIT-GN, OIT-LM, and OIT-proposed unexpectedly obtains an accuracy higher than REF, despite that the sensor units are allowed to be arbitrarily oriented on the body.

4.5 Run-Time Analysis

We have determined the run times of the OEMs by running them stand alone (that is, not as part of an OIT but externally). According to the run times provided in Table 4.1, the proposed OEM is computationally more efficient than KF, GN, and LM by factors of 4.6, 2.9, and 5.9 and less efficient than



(a)



(b)

Figure 4.6: Activity recognition accuracy for the data transformation techniques and classifiers. (a) Individual results of the four selected classifiers and (b) their average accuracy.

TRIAD and GD by factors of 1.02 and 1.5, respectively. Since the computationally efficient approaches (GD and TRIAD) are not very accurate and the slightly more accurate algorithms (GN and LM) have much longer run times, the newly proposed method achieves a satisfactory compromise between accuracy and run time. For comparison, it is stated in [115] that linear and extended Kalman filter based approaches take 3.1 and 5.5 times more processing, respectively, compared to the approximated GD as in [111]. The average classification times of the four classifiers are 0.38, 0.04, 0.01, and 1.49 ms per time segment, which can be neglected compared to the run times of the OEMs.

Table 4.1: Average run times of the OEMs compared in this study.

OEM	**run time per 5-s time segment (ms)**
TRIAD	19.45
KF	91.13
GD	13.36
GN	57.66
LM	115.93
proposed	19.82

4.6 Concluding Remarks

We have demonstrated that among the five state-of-the-art OEMs, the simpler and computationally efficient TRIAD and GD are not very accurate (within the context of activity recognition) whereas GN and LM are computationally expensive, despite being slightly more accurate. The KF method is neither very accurate nor computationally efficient in the proposed activity recognition scheme. We have developed a non-iterative OEM based on physical and geometric properties of two reference vectors that is simple to implement and efficient for real-time execution. We have evaluated the effectiveness of our method in a real-world scenario of daily and sports activity recognition where the motion sensor units can be worn on the body at arbitrary orientations, as proposed in Chapter 3. By only replacing the OEM in this scheme with the newly proposed one, accuracy is improved and the run time is considerably reduced.

Chapter 5

Invariance to Sensor Unit Position

In this chapter, we develop techniques that provide flexibility in the positioning of wearable motion sensor units. For this purpose, we achieve position invariance within the same body part, allow the interchangeability of the units, and perform classification based on a single sensor unit. We assume that the sensor unit orientations are fixed in this chapter and consider simultaneous position and orientation invariance in Chapter 6.

Throughout this chapter, we employ the publicly available dataset acquired by our research group, comprised of 19 daily and sports activities [80, 81, 86]. The dataset (referred as dataset A in Section 2.3.1) is described in Section 3.4 and its attributes are provided in the second column of Table 2.1[1]. The sensor unit configuration is shown in Figure 3.5.

To assess the performance of existing and proposed methods, we apply the activity recognition scheme that is described in Section 3.4.1.2: Time-domain data are divided into non-overlapping time segments of 5-s duration. Then, existing and proposed transformation techniques are applied to the data for robustness to sensor

[1]The remaining datasets are not used in this chapter since they do not include data from a magnetometer which are required to implement some of the techniques proposed here.

unit positioning. Statistical features are extracted for each segment of each axis of each sensor type, as described in Section 2.3.2.1. The features are normalized and reduced through PCA (see Section 2.3.2.1). Seven state-of-the-art classifiers that are explained in Section 3.4.1.2 are considered and their accuracies are assessed using P-fold and L1O cross-validation techniques, as described in Section 2.3.2.3.

5.1 Position Invariance within the Same Body Part

In this section, we focus on techniques that achieve invariance to sensor unit positioning within the same body part as a first step to achieve position invariance.

Measurements of motion sensors are related directly to the linear and angular motion of the rigid body at which they are attached. We assume that the body part on which the sensor unit is placed, such as the lower arm, is considered to be rigid so that the relative position of any point with respect to another point remains constant in time. In other words, the distance between any two arbitrary points is preserved. The motion of a rigid body at any time instant can be described by a translation and a rotation in 3D space [124]. The linear velocity of all points within the rigid body is the same and can be represented by a 3×1 column vector $\boldsymbol{v}$. The angular velocity of all the points on the rigid body is also the same and can be represented by a 3×1 angular velocity (rate) vector $\boldsymbol{\omega}$. The vector $\boldsymbol{\omega}$ points along the axis of rotation and its magnitude represents the rate of rotation. The direction of rotation can be found using the right-hand rule. A gyroscope directly measures the angular rate vector $\boldsymbol{\omega}$ associated with the rigid body.

A magnetometer measures the vector sum $\mathbf{m}$ of the Earth's magnetic field and external magnetic sources, if any. The Earth's magnetic field is approximately constant within the human body and does not change much with the position of the sensor unit. Hence, the magnetometer data depend only on the orientation of the body part and not the sensor unit position on it.

According to the Coriolis theorem, an accelerometer measures the vector sum $\mathbf{a}$ of multiple acceleration components [124]:

$$\mathbf{a} = \underbrace{\dot{\boldsymbol{v}} + \mathbf{g}}_{\mathbf{a}_{\text{linear}}} + \underbrace{\boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r})}_{\mathbf{a}_{\text{centripetal}}} + \underbrace{\dot{\boldsymbol{\omega}} \times \mathbf{r}}_{\mathbf{a}_{\text{Euler}}} + \underbrace{2\boldsymbol{\omega} \times \dot{\mathbf{r}}}_{\mathbf{a}_{\text{Coriolis}}} \tag{5.1}$$

where $\dot{\boldsymbol{v}}$ is translational acceleration due to linear motion, $\mathbf{g}$ is the gravitational acceleration, $\dot{\boldsymbol{\omega}}$ is the angular acceleration, and $\mathbf{r}$ is the vector pointing from an arbitrary point (the origin) on the axis of rotation to the sensor unit as illustrated in Figure 5.1. The dot accent ($\dot{}$) in Equation 5.1) represents the first-order time derivative.

When the sensor unit is worn at a different position on the same body part, the vector $\mathbf{r}$ becomes $\mathbf{r}' = \mathbf{r} + \boldsymbol{\Delta}\mathbf{r}$, where $\boldsymbol{\Delta}\mathbf{r}$ is the sensor unit displacement (Figure 5.1). The acceleration $\mathbf{a}'$ of the displaced sensor unit can be expressed in terms of the acceleration $\mathbf{a}$ at the original sensor unit position and $\boldsymbol{\Delta}\mathbf{r}$ as follows:

$$\begin{aligned}
\mathbf{a}' &= \dot{\boldsymbol{v}} + \mathbf{g} + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}') + \dot{\boldsymbol{\omega}} \times \mathbf{r}' + 2\boldsymbol{\omega} \times \dot{\mathbf{r}'} \\
&= \dot{\boldsymbol{v}} + \mathbf{g} + \boldsymbol{\omega} \times [\boldsymbol{\omega} \times (\mathbf{r} + \boldsymbol{\Delta}\mathbf{r})] + \dot{\boldsymbol{\omega}} \times (\mathbf{r} + \boldsymbol{\Delta}\mathbf{r}) + 2\boldsymbol{\omega} \times \left(\dot{\mathbf{r}} + \dot{\boldsymbol{\Delta}\mathbf{r}}\right) \\
&= \mathbf{a} + \underbrace{\boldsymbol{\omega} \times (\boldsymbol{\omega} \times \boldsymbol{\Delta}\mathbf{r})}_{\boldsymbol{\Delta}\mathbf{a}_{\text{centripetal}}} + \underbrace{\dot{\boldsymbol{\omega}} \times \boldsymbol{\Delta}\mathbf{r}}_{\boldsymbol{\Delta}\mathbf{a}_{\text{Euler}}} + \underbrace{2\boldsymbol{\omega} \times \dot{\boldsymbol{\Delta}\mathbf{r}}}_{\boldsymbol{\Delta}\mathbf{a}_{\text{Coriolis}}}
\end{aligned} \tag{5.2}$$

We assume that once the subject places the sensor unit on his/her body, its position with respect to the body remains fixed over time in the short term. We model this by keeping the sensor unit displacement $\boldsymbol{\Delta}\mathbf{r}$ constant during each time segment ($\dot{\boldsymbol{\Delta}\mathbf{r}} = 0$). Thus, the Coriolis acceleration $\mathbf{a}_{\text{Coriolis}}$ is not affected by the change in the sensor unit position on the same body part. This is also true for the component $\mathbf{a}_{\text{linear}}$ since both $\dot{\boldsymbol{v}}$ and $\mathbf{g}$ are the same everywhere on the body part, provided that $\dot{\boldsymbol{\Delta}\mathbf{r}} = \mathbf{0}$. Hence, positioning the sensor unit differently on the same body part introduces the two additional components: $\boldsymbol{\Delta}\mathbf{a}_{\text{centripetal}}$ and $\boldsymbol{\Delta}\mathbf{a}_{\text{Euler}}$.

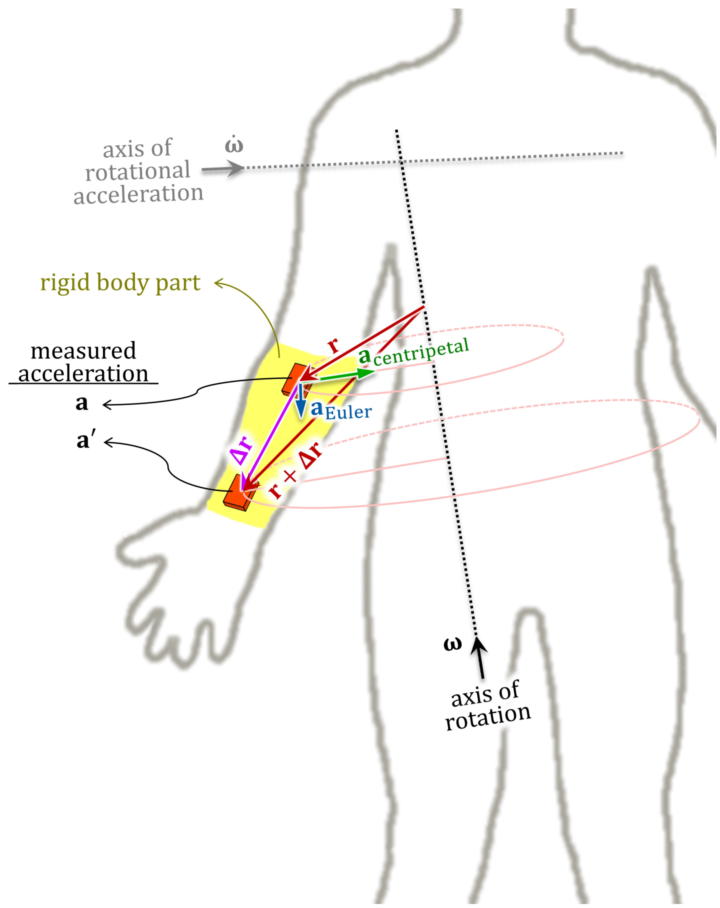


Figure 5.1: Sensor unit positioning within the same rigid body part. The displacement between two arbitrary positions and the centripetal and Euler components of the acquired acceleration vector are shown.

5.1.1 Impact of Sensor Unit Positioning within the Same Body Part on the Activity Recognition Accuracy

To observe the effects of sensor unit positioning on the accuracy, we first simulate the scenario where the sensor units are randomly positioned on the body parts at which they are originally placed. We generate a constant random displacement (RD) vector $\mathbf{\Delta r}$ independently for each sensor unit for each time segment of the data. Then, we calculate the acceleration vector $\mathbf{a}'$ for the displaced sensor unit based on the original measurement $\mathbf{a}$ by using Equation (5.2) (with the last term being zero).

We consider that each sensor unit is positioned on a disk with a given radius $d_{\max}$ centered at its ideal position and the displacement $\mathbf{\Delta r}$ is restricted to reside on the plane where the unit makes contact with the body part it is attached to. All of the five sensor units make contact with the x-y plane (see Figure 3.5). We consider three different simulation models where the direction of $\mathbf{\Delta r}$ is selected circularly symmetrically in all of them but its magnitude is determined differently:

- **RD-conc (concentrated):** The magnitude of $\mathbf{\Delta r}$ (which is d) is uniformly distributed between 0 and $d_{\max}$ so that $d \sim \mathcal{U}\,[0, d_{\max}]$. Its angular direction ϑ also has uniform distribution: $\vartheta \sim \mathcal{U}\,[0, 2\pi)$. Then, the vector $\mathbf{\Delta r}$ can be calculated as $\mathbf{\Delta r} = [d \sin\vartheta,\ d\cos\vartheta,\ 0]^T$.

- **RD-trun (truncated Gaussian):** The distribution of the points on the x-y plane is selected as a bi-variate Gaussian random vector. For this purpose, we generate two independent and identically distributed Gaussian random variables Δr_x, Δr_y with zero mean and standard deviation $0.4\, d_{\max}$ so that $\Delta r_x,\ \Delta r_y \sim \mathcal{N}\,(0,\ 0.16\, d_{\max}^2)$. Then, we generate the displacement vector as $\mathbf{\Delta r} = [\Delta r_x,\ \Delta r_y,\ 0]^T$. To ensure that $\mathbf{\Delta r}$ is on the disk centered at the origin with radius $d_{\max}$, we repeat this process as many times as necessary until $\mathbf{\Delta r}$ resides inside the disk.

- **RD-uni (uniformly distributed per unit area):** The displacement points $\mathbf{\Delta r}$ are generated to have uniform distribution per unit area on the x-y plane. Two independent and identically distributed random variables Δr_x, Δr_y are generated uniformly in the interval $[-d_{\max}, d_{\max}]$ and this process is repeated as many times as needed until $\mathbf{\Delta r}$ resides inside the disk centered at the origin with radius $d_{\max}$.

For the RD-conc and RD-trun models, the displacement points ($\mathbf{\Delta r}$) are concentrated around the origin, corresponding to the case where the units are more likely to be placed close to their correct/ideal positions. On the other hand, for the RD-uni model, the $\mathbf{\Delta r}$ points are uniformly distributed on the disk with equal density per unit area. Note that the displacement distance is bounded by $d_{\max}$ in all of the three models.

To analyze the effect of randomly displacing the sensor unit positions on the activity recognition accuracy, we apply one of the transformations RD-conc, RD-trun, and RD-uni to the test data in each cross-validation iteration, while keeping the original training data that are associated with the correctly placed sensor units. We execute the activity recognition scheme for different $d_{\max}$ values ranging from 0.5 cm to 100 cm. We provide the accuracy values in Figures 5.2–5.4 for the three RD simulation models. The classification accuracy is presented for each classifier separately at the top and by averaging over the classifiers at the bottom in the figures. The standard deviation sticks at the top and bottom parts of the figures indicate plus/minus one standard deviation about the accuracies over the cross-validation iterations and over the classifiers, respectively. Parts (a) and (b) of the figures correspond to the P-fold and L1O cross-validation techniques, respectively.

Referring to Figures 5.2–5.4, we observe that the activity recognition accuracy naturally decreases when the sensor units are fixed to different positions within the body part they are supposed to be put on. Displacements up to a few centimeters can be tolerated by the standard activity recognition scheme, whereas displacements by more than 10 cm significantly degrade the accuracy. The approaches RD-trun and RD-conc have similar trends with each other, whereas RD-uni exhibits a greater degradation in the accuracy when the units are displaced. This is expected because the distribution of the displacement points $\mathbf{\Delta r}$ is concentrated around the origin for RD-trun and RD-conc unlike RD-uni where the distribution is equal throughout the disk. The drop in the accuracy is expected because training data are associated with correctly positioned sensor units while the test data are displaced and the classifiers are not trained for this displacement. In particular, the distance of 100 cm has a higher accuracy for L1O than for P-fold because the training data in L1O have wider variations among the partitions (as each partition contains data acquired from a *different* subject) and the classifiers are more prepared for possible variations in the test data.

5.1.2 Proposed Methods for Robustness to Displacement within the Same Body Part

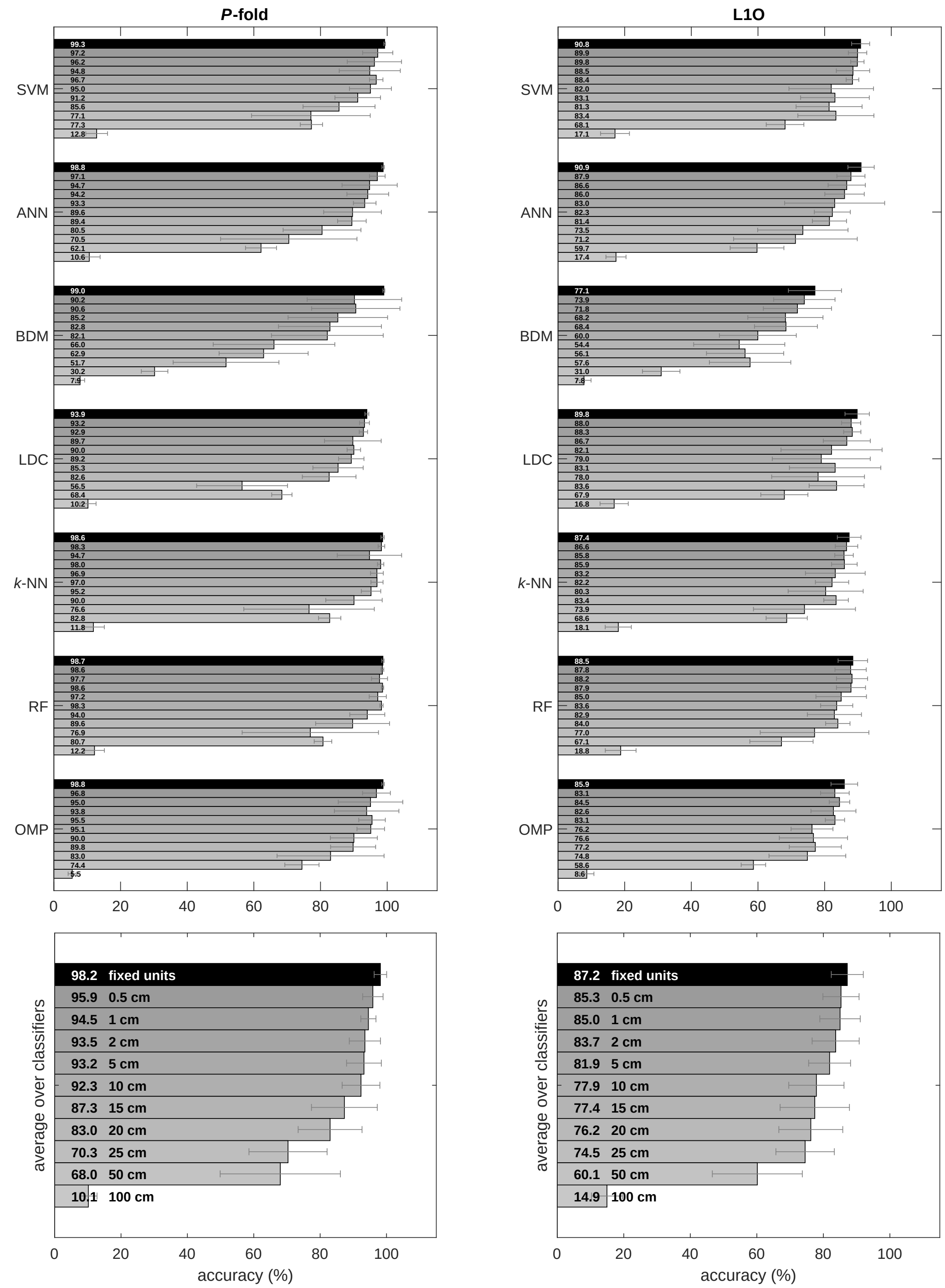
The gyroscope ($\boldsymbol{\omega}$) and magnetometer ($\mathbf{m}$) sequences are invariant to the positioning of the sensor unit within the same body part (which is considered to be rigid), and thus, are used for classification without making any modifications. On the other hand, the acceleration sequences ($\mathbf{a}$) depend on the position of the unit and the classification accuracy is degraded when they are directly used in the classification process, as shown in Section 5.1.1. Hence, we propose to extract sequences that are functions of time and robust to the positioning of the sensor unit within the same body part and to use these sequences in in the classification process instead of the raw acceleration data.

To extract position-invariant sequences, we analyze the two components caused by displacing the sensor unit according to Equation (5.2):

$$\boldsymbol{\Delta}\mathbf{a}_{\text{Euler}} = \dot{\boldsymbol{\omega}} \times \boldsymbol{\Delta}\mathbf{r} \tag{5.3}$$

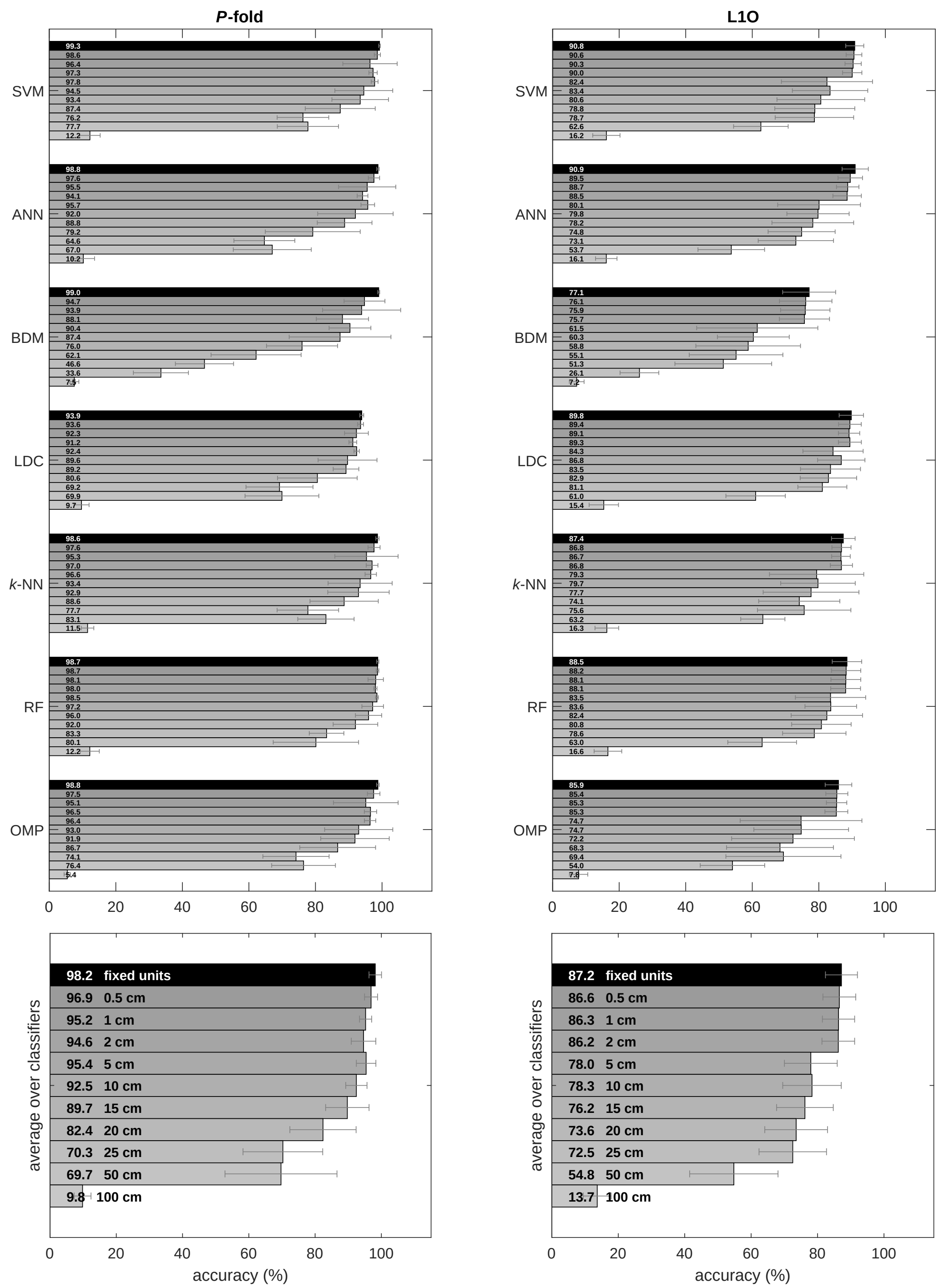
$$\boldsymbol{\Delta}\mathbf{a}_{\text{centripetal}} = \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \boldsymbol{\Delta}\mathbf{r}) \tag{5.4}$$

The components $\boldsymbol{\Delta}\mathbf{a}_{\text{Euler}}$ and $\boldsymbol{\Delta}\mathbf{a}_{\text{centripetal}}$ are perpendicular to $\dot{\boldsymbol{\omega}}$ and $\boldsymbol{\omega}$, respectively, for a given displacement vector $\boldsymbol{\Delta}\mathbf{r}$. Their magnitudes are calculated



(a) (b)

Figure 5.2: Activity recognition accuracy for fixed (reference) and randomly displaced units with the RD-conc approach for (a) P-fold and (b) L1O cross validation. The lengths of the bars indicate the accuracy values. The thin sticks represent plus/minus one standard deviation over the cross-validation iterations and over the classifiers at the top and bottom parts of the figure, respectively.



(a) (b)

Figure 5.3: Activity recognition accuracy for fixed (reference) and randomly displaced units with the RD-trun approach for (a) P-fold and (b) L1O cross validation.

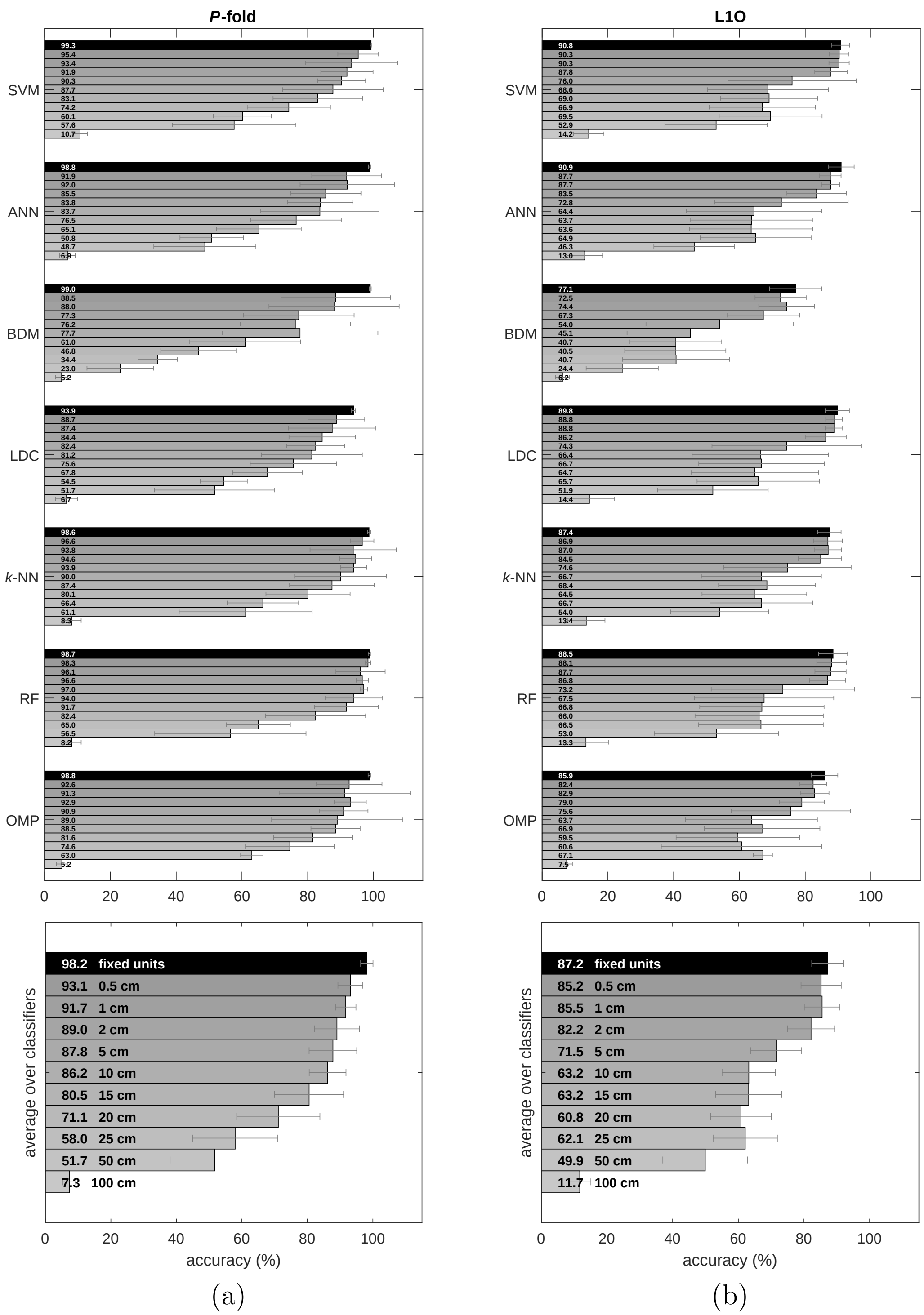


Figure 5.4: Activity recognition accuracy for fixed (reference) and randomly displaced units with the RD-uni approach for (a) P-fold and (b) L1O cross validation.

as follows:

$$\|\mathbf{\Delta a}_{\text{Euler}}\| = \|\dot{\boldsymbol{\omega}}\| \;\; \|\mathbf{\Delta r}\| \, \sin\left(\angle(\dot{\boldsymbol{\omega}}, \mathbf{\Delta r})\right) \tag{5.5}$$

$$\|\mathbf{\Delta a}_{\text{centripetal}}\| = \|\boldsymbol{\omega}\|^2 \, \|\mathbf{\Delta r}\| \, \sin\left(\angle(\boldsymbol{\omega}, \mathbf{\Delta r})\right) \tag{5.6}$$

To observe which of the two components is dominant, we define the ratio

$$\rho \triangleq \frac{\|\mathbf{\Delta a}_{\text{Euler}}\|}{\|\mathbf{\Delta a}_{\text{centripetal}}\|} = \frac{\|\dot{\boldsymbol{\omega}}\|}{\|\boldsymbol{\omega}\|^2} \frac{\sin\left(\angle(\dot{\boldsymbol{\omega}}, \mathbf{\Delta r})\right)}{\sin\left(\angle(\boldsymbol{\omega}, \mathbf{\Delta r})\right)} \tag{5.7}$$

By defining the following variables:

$$\sigma \triangleq \frac{\|\dot{\boldsymbol{\omega}}\|}{\|\boldsymbol{\omega}\|^2} \tag{5.8}$$

$$\beta \triangleq \angle(\dot{\boldsymbol{\omega}}, \mathbf{\Delta r}) \tag{5.9}$$

$$\Upsilon \triangleq \angle(\boldsymbol{\omega}, \mathbf{\Delta r}) \tag{5.10}$$

$$\lambda \triangleq \frac{\sin\beta}{\sin\Upsilon} \tag{5.11}$$

the ratio in Equation (5.7) may be expressed as $\rho = \sigma\lambda$. Then, we may neglect $\mathbf{\Delta a}_{\text{centripetal}}$ when $\rho \gg 1$. In this case, the projection

$$\mathrm{p} \triangleq \mathbf{a} \cdot \frac{\dot{\boldsymbol{\omega}}}{\|\dot{\boldsymbol{\omega}}\|} \tag{5.12}$$

of acceleration onto the direction of $\dot{\boldsymbol{\omega}}$ is independent of the sensor unit displacement $\mathbf{\Delta r}$ because the only component $\mathbf{\Delta a}_{\text{Euler}}$ (we consider) that originates from the random displacement is orthogonal to $\dot{\boldsymbol{\omega}}$. Hence, we calculate the component p of $\mathbf{a}$ along the direction of $\dot{\boldsymbol{\omega}}$ which is approximately invariant to sensor unit position within the same body part.

The orientation of the sensor unit with respect to the Earth frame can also be included as position-invariant feature. For this purpose, the orientation of the sensor unit is estimated with respect to the fixed Earth frame based on the accelerometer, gyroscope, and magnetometer data by using the OEM proposed in Chapter 4. According to the ENU convention, the x, y, z axes of the Earth frame point to the East, North, and up directions, respectively. The sensor unit orientation is represented by a 4×1 quaternion vector $\mathbf{q}$ for each time sample, as a feature that is invariant to the position of the unit within the same body part.

We propose two approaches where different combinations of the position-invariant sequences $\boldsymbol{\omega}$, $\mathbf{m}$, p, and $\mathbf{q}$ are used for classification: $\boldsymbol{\omega}\mathbf{m}$p and $\boldsymbol{\omega}\mathbf{m}p\mathbf{q}$. To assess the position invariance of these two approaches, we randomly displace the sensor unit positions as follows: For the training data, we only apply $\boldsymbol{\omega}\mathbf{m}$p (or $\boldsymbol{\omega}\mathbf{m}p\mathbf{q}$), whereas for the test data, we first randomly displace the unit positions and then apply $\boldsymbol{\omega}\mathbf{m}$p (or $\boldsymbol{\omega}\mathbf{m}p\mathbf{q}$). In this way, we simulate the case where we implement the $\boldsymbol{\omega}\mathbf{m}$p (or $\boldsymbol{\omega}\mathbf{m}p\mathbf{q}$) to achieve robustness to sensor unit positioning where the units are placed at different positions within the pre-determined body parts during the activity recognition scheme.

We statistically analyze the quantities σ, λ, and ρ in our dataset as follows:

- Among all 5-s time segments, the minimum ratio of time samples where $\sigma > 1$ is 68.8%. The histogram for the percentage of time samples in a time segment is shown in Figure 5.5(a).
- The average value of σ over all the 5,700,000 time samples in the dataset is $\bar{\sigma} = 897.9$. We have $\sigma > 1$ for 97.3% of these time samples. The histogram for σ is shown in Figure 5.5(b).
- The ratio λ is plotted as a function of the angles β and Υ in Figure 5.5(c). The angles depend on the direction of the displacement $\mathbf{\Delta r}$. The ratio λ increases as Υ approaches to 0 or π rad and decreases as β approaches to 0 or π rad.
- When the direction of $\mathbf{\Delta r}$ is selected uniformly, the distribution of ρ can be empirically calculated. The histogram for ρ is shown in Figure 5.5(d). We have $\rho > 1$ for 97.8% of the time samples in the dataset.

These statistics indicate that ρ is much greater than one; that is, $\|\mathbf{\Delta a}_{\text{Euler}}\| \gg \|\mathbf{\Delta a}_{\text{centripetal}}\|$ for almost all of the time samples in the dataset. Hence, we can neglect the component $\mathbf{\Delta a}_{\text{centripetal}}$ and rely on this fact to use p as a position-invariant feature within the same body part.

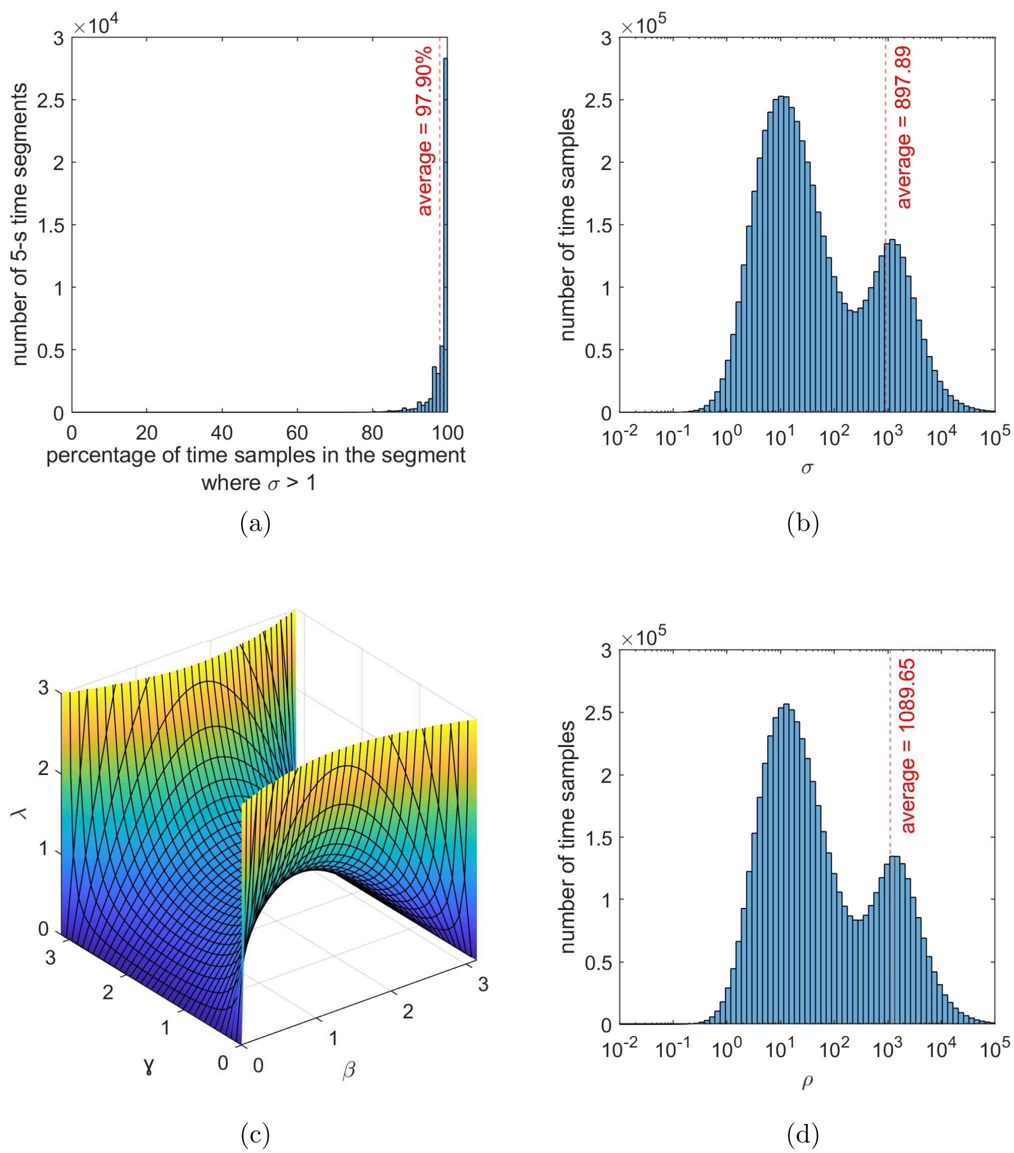
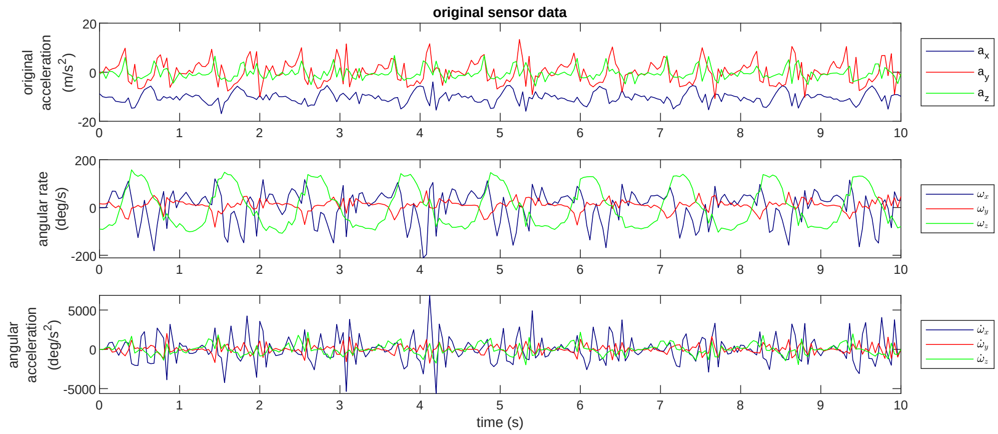


Figure 5.5: Statistics of the quantities σ, λ, and ρ that are related to the centripetal and Euler components of the acceleration. (a) Histogram of the percentage of time samples in a segment where $\sigma > 1$, (b) histogram of σ over time samples, (c) surface plot for ρ on the β-Υ plane, and (d) histogram of ρ over time samples.

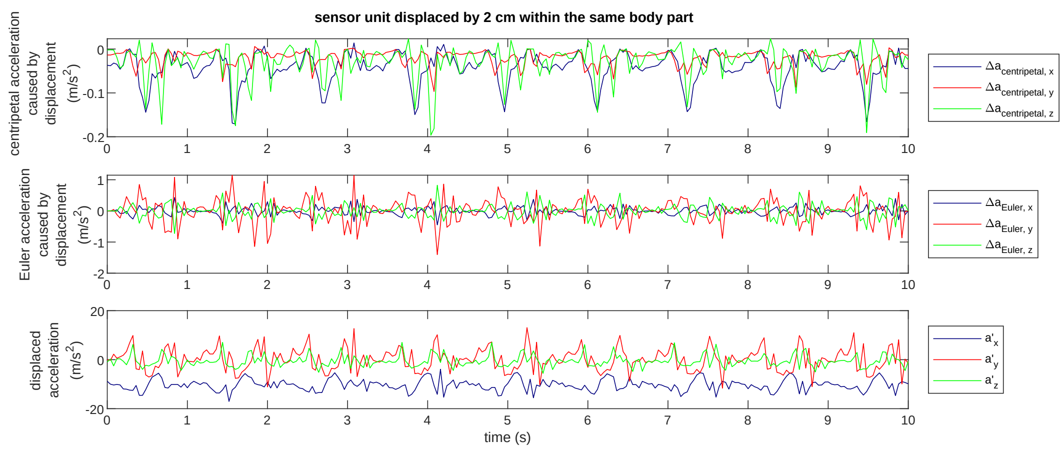
The x, y, z components of the original acceleration $\mathbf{a}$, angular rate $\boldsymbol{\omega}$, and angular acceleration $\dot{\boldsymbol{\omega}}$ vectors are plotted as functions of time in Figure 5.6(a) for the sensor unit on the right leg of a subject during activity A_{10} (see Section 2.3.1). The vectors $\boldsymbol{\Delta}\mathbf{a}_{\text{centripetal}}$ and $\boldsymbol{\Delta}\mathbf{a}_{\text{Euler}}$ caused by the sensor displacement as well as the acceleration $\mathbf{a}'$ for the displaced sensor unit are plotted as functions of time for $d_{\max} = 2$ cm and $d_{\max} = 15$ cm in Figure 5.6(b) and (c), respectively. We observe that $\boldsymbol{\Delta}\mathbf{a}_{\text{Euler}}$ has a magnitude greater than $\boldsymbol{\Delta}\mathbf{a}_{\text{centripetal}}$ most of the time and thus has a stronger effect on the acceleration $\mathbf{a}'$ measured by the displaced sensor unit. The acceleration component p and the orientation quaternion $\mathbf{q}$ are plotted as a function of time in parts (a) and (b) of Figure 5.7, respectively, for the same recording as in Figure 5.6. The periodicity of the motion is apparent in Figure 5.7(b).

The activity recognition accuracies for the $\boldsymbol{\omega}\mathbf{m}$p approach along with the three random displacement types RD-conc, RD-trun, and RD-uni are provided in Figures 5.8–5.10, respectively. We observe in Figure 5.8 that when the units are fixed, the $\boldsymbol{\omega}\mathbf{m}$p approach yields almost the same accuracy as the reference case for P-fold cross validation and a similar accuracy with the reference for L1O (see Figure 5.2). The accuracy of the $\boldsymbol{\omega}\mathbf{m}$p approach is not affected by RD-disk up to 50-cm displacement unlike the reference case (compare Figures 5.2–5.4 with 5.8–5.10), whereas a maximum sensor unit displacement of 100 cm causes a noticeable reduction in accuracy, especially for RD-uni. Nevertheless, the position-invariant feature p performs much better than the raw acceleration $\mathbf{a}$ when the units are displaced.

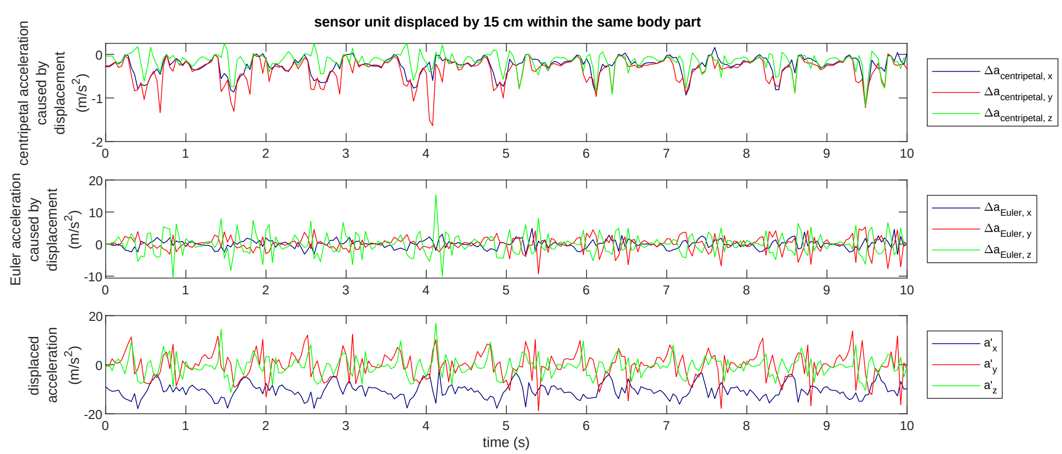
The activity recognition accuracies for the $\boldsymbol{\omega}\mathbf{m}p\mathbf{q}$ approach along with the three random displacement types RD-conc, RD-trun, and RD-uni are provided in Figures 5.11–5.13, respectively. Similar to $\boldsymbol{\omega}\mathbf{m}$p, the $\boldsymbol{\omega}\mathbf{m}p\mathbf{q}$ approach is robust to the displacement of the sensor units within the pre-determined body parts. The accuracy of $\boldsymbol{\omega}\mathbf{m}p\mathbf{q}$ is higher than $\boldsymbol{\omega}\mathbf{m}$p on the average (compare Figures 5.11–5.13 with 5.8–5.10).



(a)

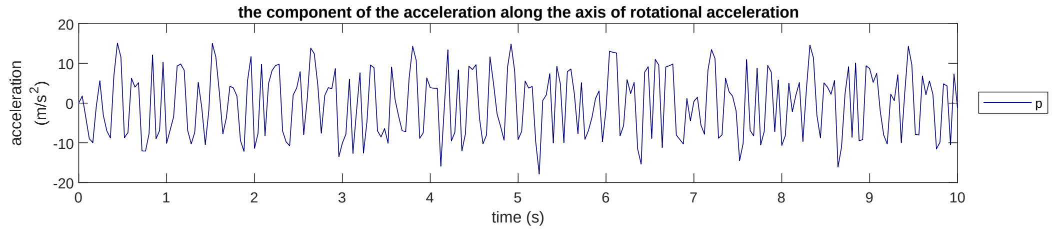


(b)

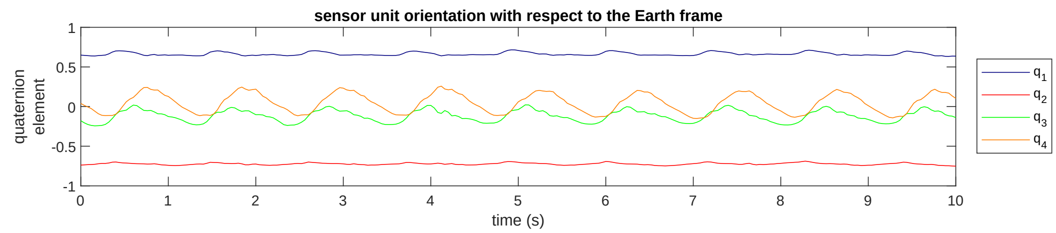


(c)

Figure 5.6: The original and displaced acceleration data. (a) The acceleration, angular rate, and angular acceleration sequences acquired from the sensor unit at the original position, (b)–(c) the centripetal, Euler, and displaced acceleration sequences calculated for the sensor unit that is displaced by 2 and 15 cm.



(a)



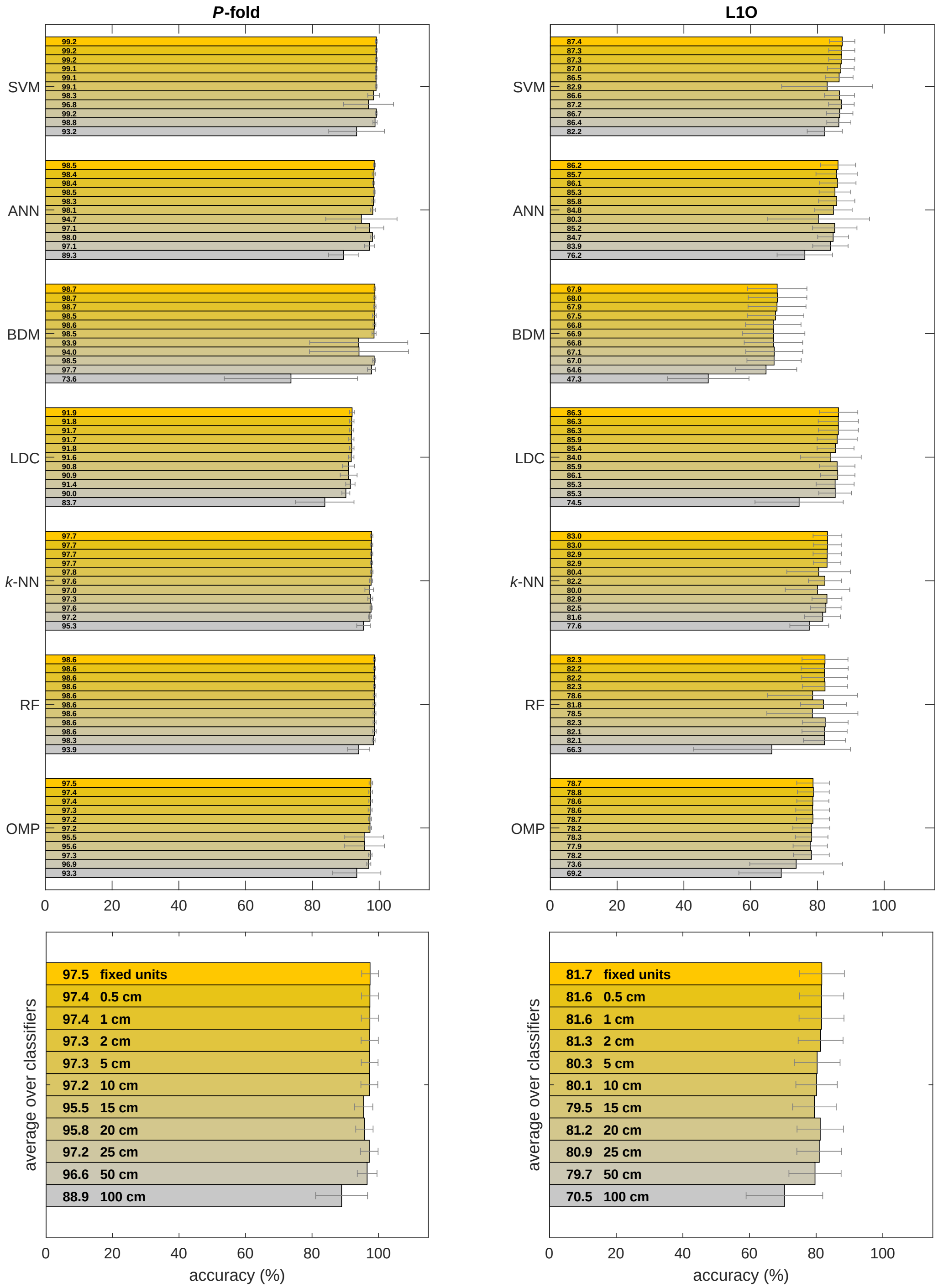
(b)

Figure 5.7: The position-invariant quantities extracted from the sensor data: (a) The component of acceleration along the direction of $\dot{\boldsymbol{\omega}}$ and (b) the quaternion that represents the sensor unit orientation with respect to the Earth frame.

5.1.3 Comparison of the Proposed and Existing Methods for Position Invariance within the Same Body Part

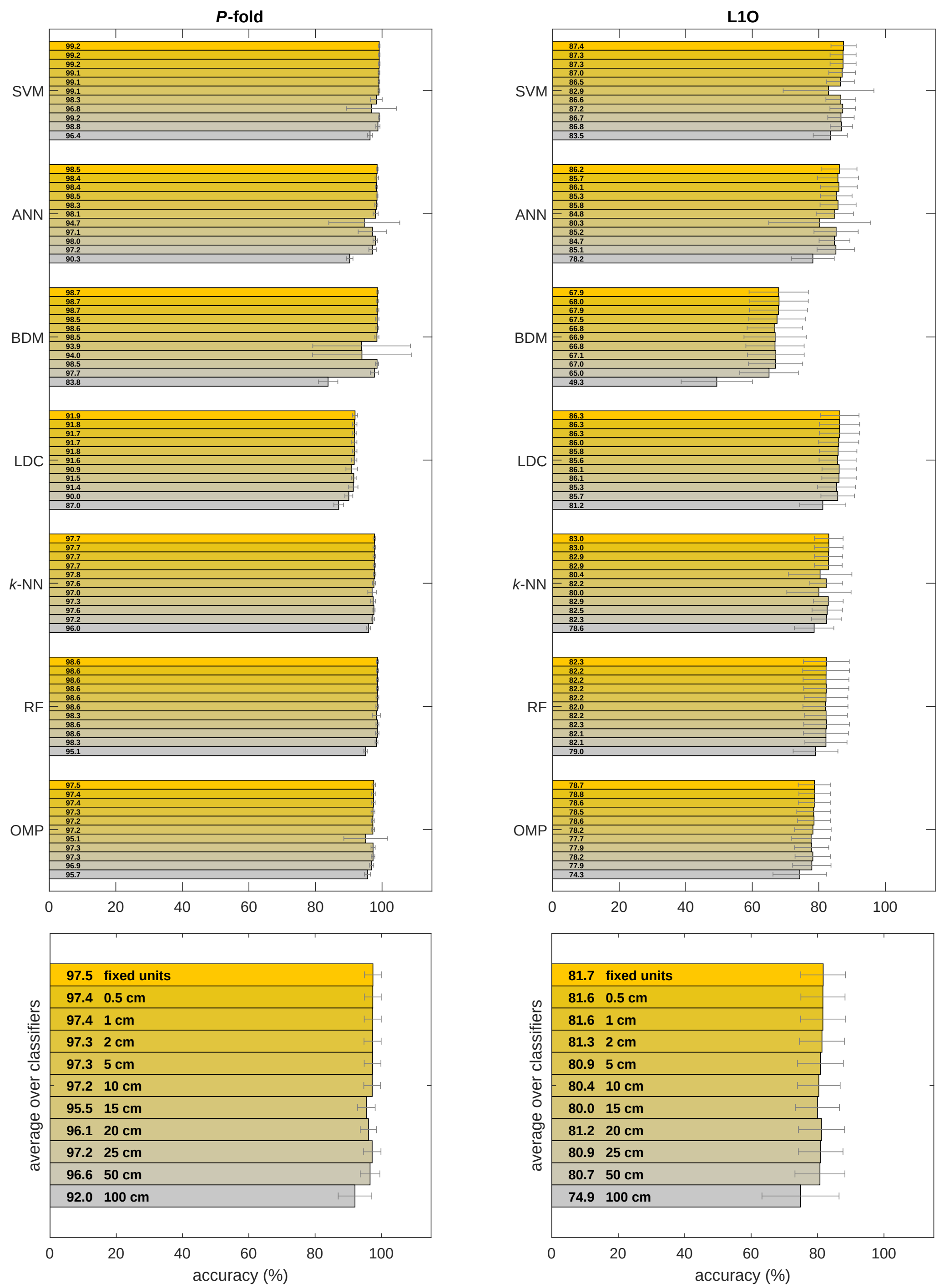
A straightforward approach to achieve position invariance within the same body part is to omit the acceleration data and to rely on the gyroscope and magnetometer data. This approach is called $\boldsymbol{\omega}\mathbf{m}$ for which the activity recognition accuracies are shown in Figure 5.14. Compared to the proposed approach $\boldsymbol{\omega}\mathbf{m}p\mathbf{q}$, the $\boldsymbol{\omega}\mathbf{m}$ approach preforms slightly worse for small displacement distances and slightly better for large displacement distances.

To our knowledge, the only existing approach that is applicable to our framework except $\boldsymbol{\omega}\mathbf{m}$ is to low-pass filter the acceleration data as proposed in [37, 38]. The acceleration sequences contain gravitational and motion-originated components, the former of which can be separated from the latter in the frequency domain for most human activities and is invariant to the sensor unit position within the same body part. The acceleration data $\mathbf{a}$ are low-pass filtered to make the gravitational



(a) (b)

Figure 5.8: Activity recognition accuracy for the $\boldsymbol{\omega}\mathbf{m}$p approach for fixed and randomly displaced units with the RD-conc approach for (a) P-fold and (b) L1O cross validation.



(a) (b)

Figure 5.9: Activity recognition accuracy for the $\boldsymbol{\omega}\mathbf{m}$p approach for fixed and randomly displaced units with the RD-trun approach for (a) P-fold and (b) L1O cross validation.

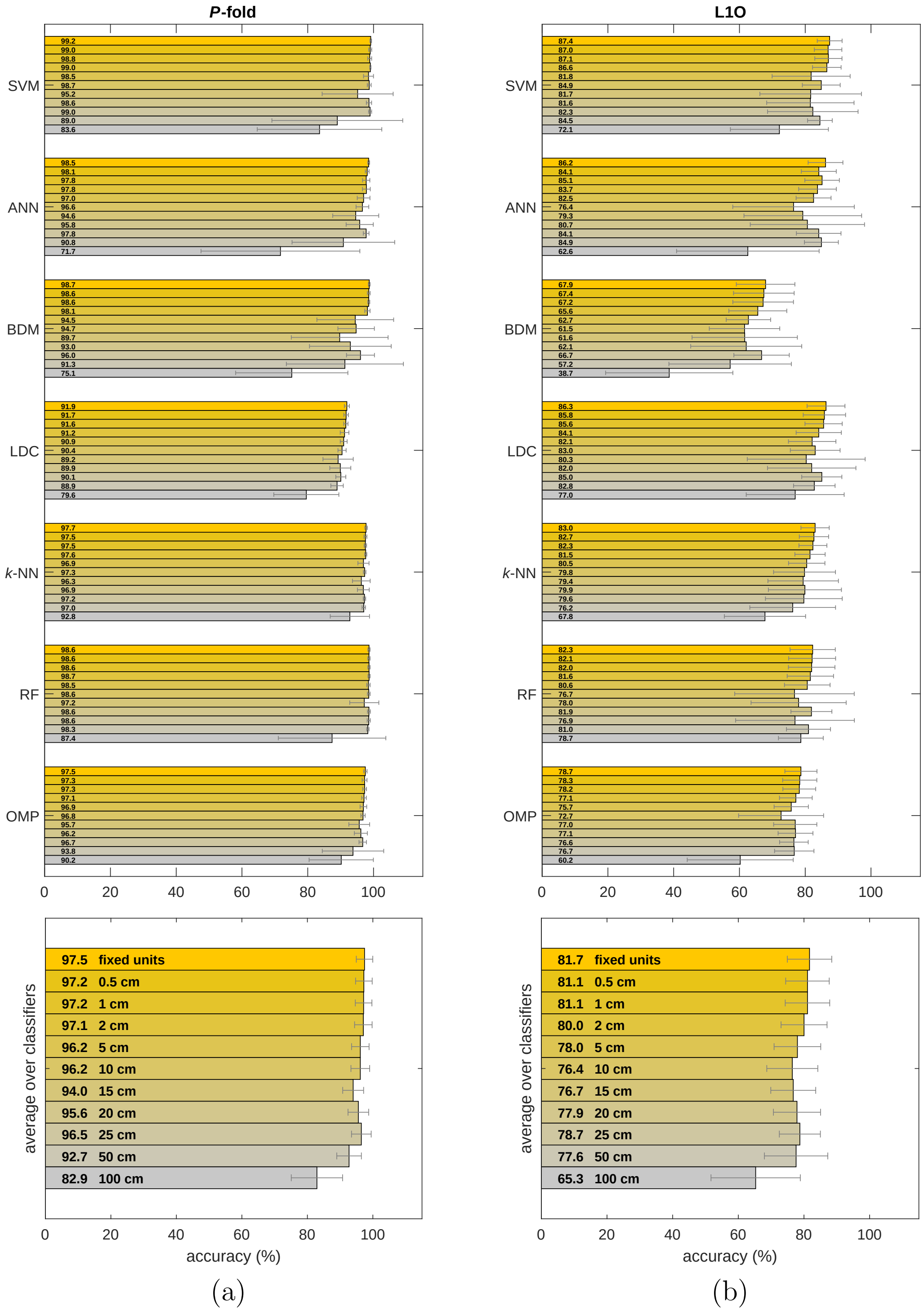


Figure 5.10: Activity recognition accuracy for the $\boldsymbol{\omega}\mathbf{m}$p approach for fixed and randomly displaced units with the RD-uni approach for (a) P-fold and (b) L1O cross validation.

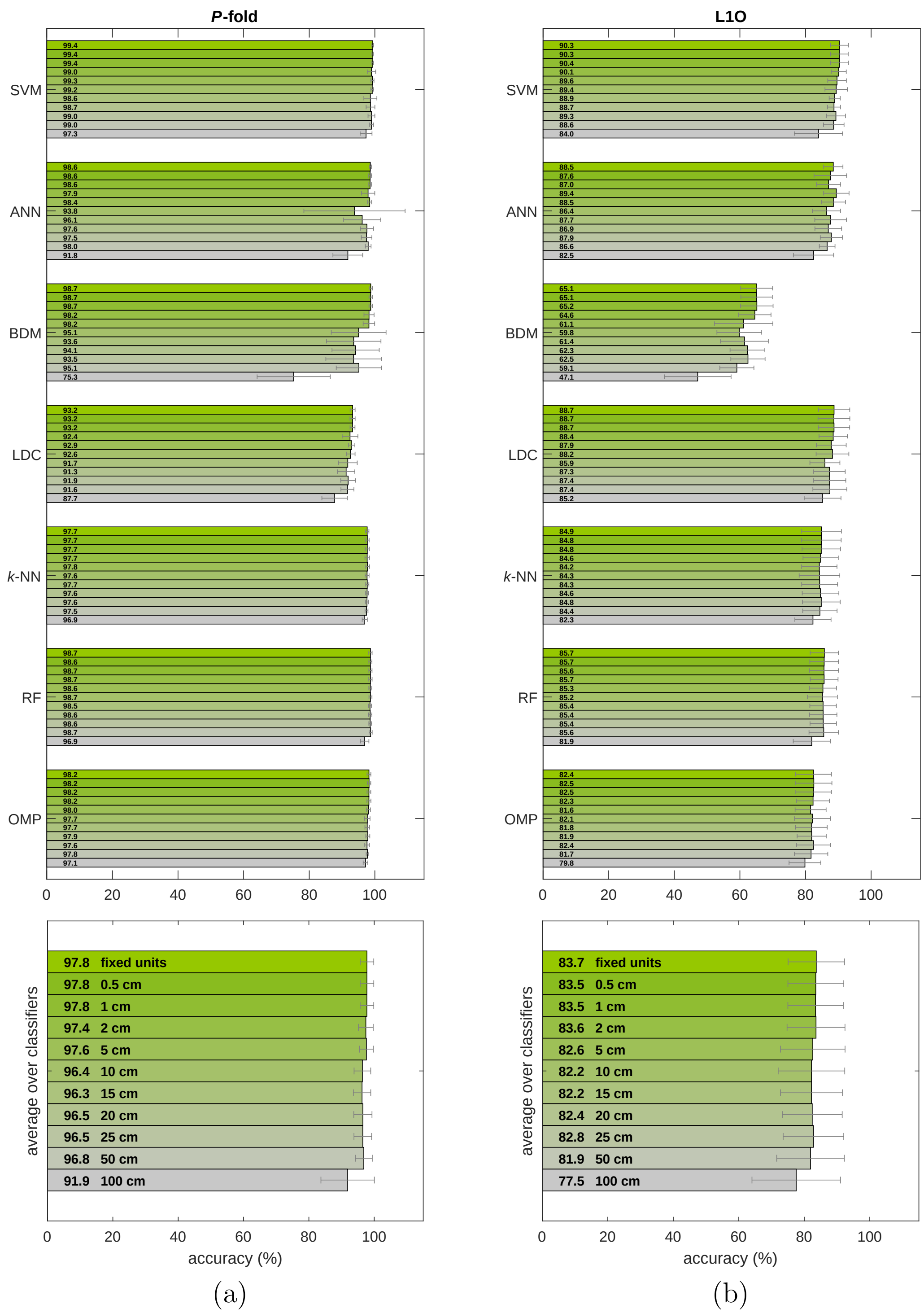
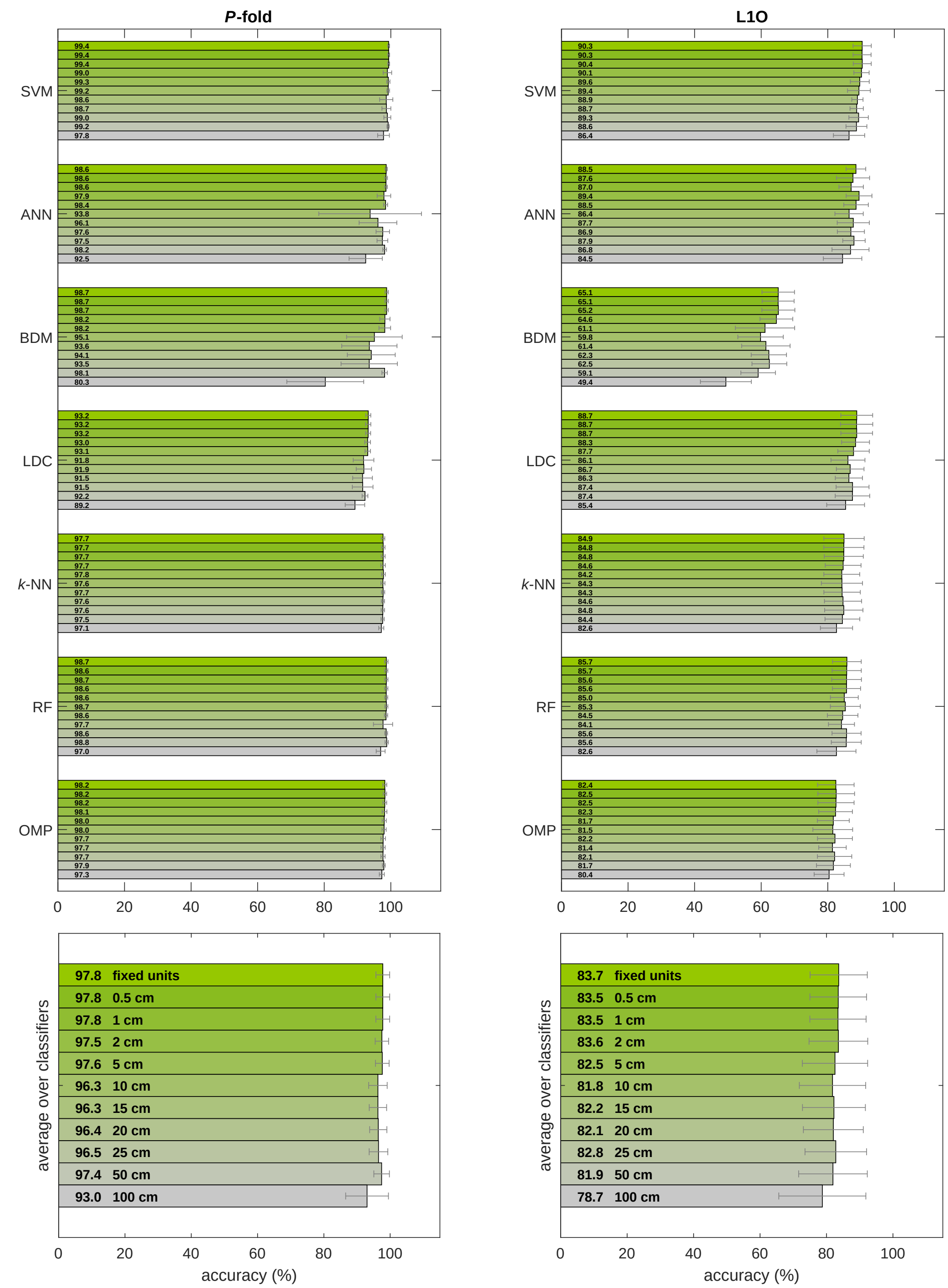


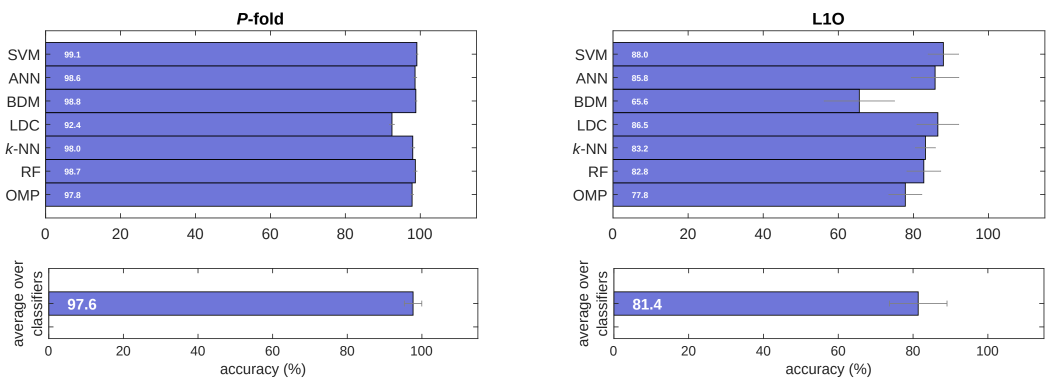
Figure 5.11: Activity recognition accuracy for the $\boldsymbol{\omega}\mathbf{m}\mathrm{pq}$ approach for fixed and randomly displaced units with the RD-conc approach for (a) P-fold and (b) L1O cross validation.



(a) (b)

Figure 5.12: Activity recognition accuracy for the $\boldsymbol{\omega}\mathbf{m}\mathrm{p}\mathbf{q}$ approach for fixed and randomly displaced units with the RD-trun approach for (a) P-fold and (b) L1O cross validation.

Figure 5.13: Activity recognition accuracy for the $\boldsymbol{\omega}\mathbf{m}\mathrm{p}\mathbf{q}$ approach for fixed and randomly displaced units with the RD-uni approach for (a) P-fold and (b) L1O cross validation.



(a) (b)

Figure 5.14: Activity recognition accuracy for the $\boldsymbol{\omega}\mathbf{m}$ approach for (a) P-fold and (b) L1O cross validation.

component more dominant over the other; thus, to improve the robustness to the sensor unit positioning. For this purpose, a zero-phase Chebyshev type-II infinite impulse response low-pass filter with a cut-off frequency of 10 Hz is applied to the acceleration sequences, as proposed in [38]. In addition to the filtered acceleration data (denoted as $\tilde{\mathbf{a}}$), the gyroscope and magnetometer sequences, $\boldsymbol{\omega}$ and $\mathbf{m}$, are also used in the classification process because they are already invariant to the positioning of the sensor unit within the same body part.

Figures 5.15–5.17 show the activity recognition rates for the aforementioned existing approach $\boldsymbol{\omega}\mathbf{m}\tilde{\mathbf{a}}$. It obtains a higher accuracy than the proposed approaches $\boldsymbol{\omega}\mathbf{m}$p and $\boldsymbol{\omega}\mathbf{m}p\mathbf{q}$ for displacement distances up to a few centimeters; however, its accuracy significantly decreases when the displacement exceeds several centimeters, which shows that it is not as robust as the newly proposed methods to the positioning of the sensor units. In particular, for the maximum sensor displacement of 100 cm, the existing approach $\boldsymbol{\omega}\mathbf{m}\tilde{\mathbf{a}}$ performs poorly, whereas the proposed approaches $\boldsymbol{\omega}\mathbf{m}$p and $\boldsymbol{\omega}\mathbf{m}p\mathbf{q}$ perform fairly well.

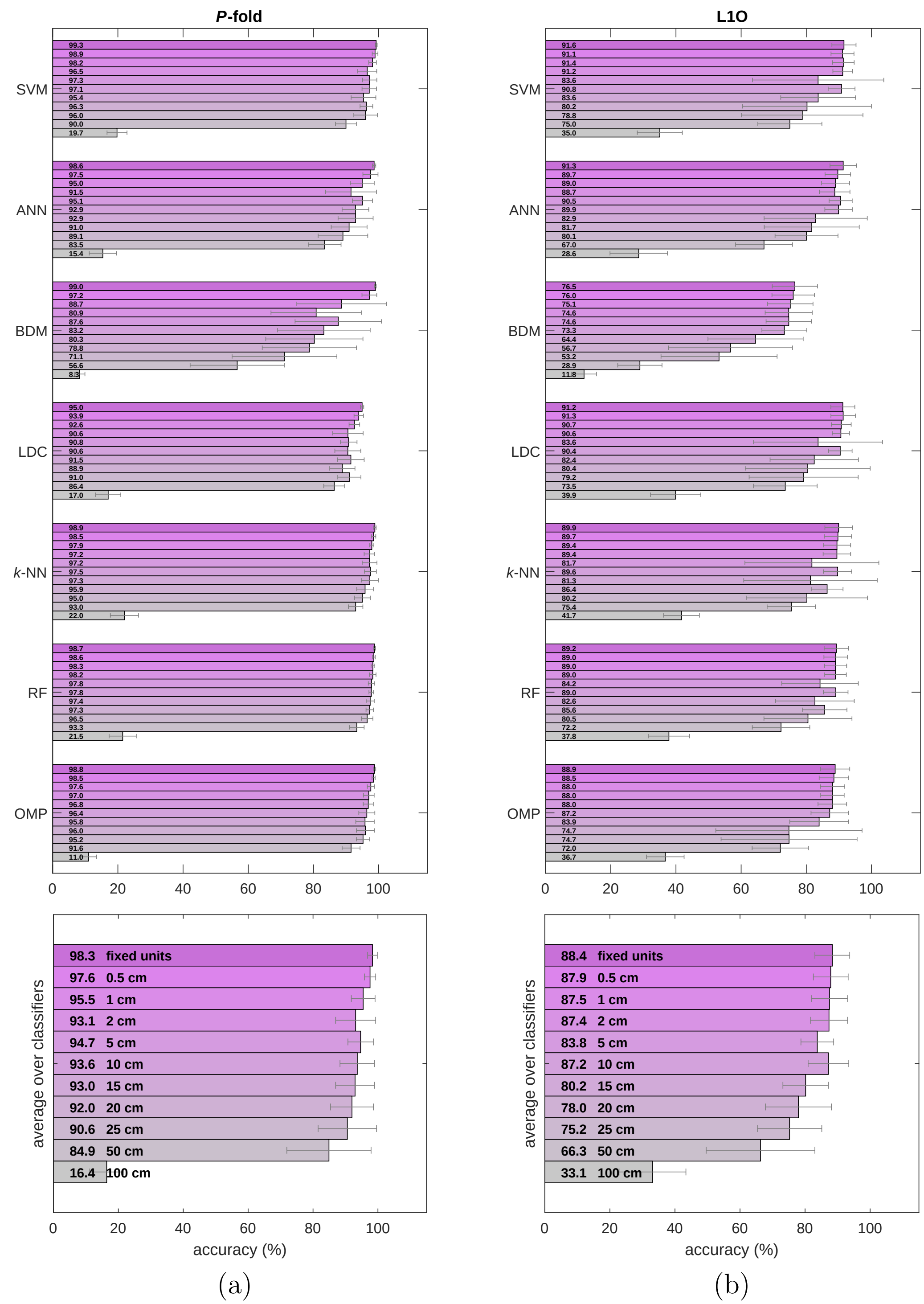
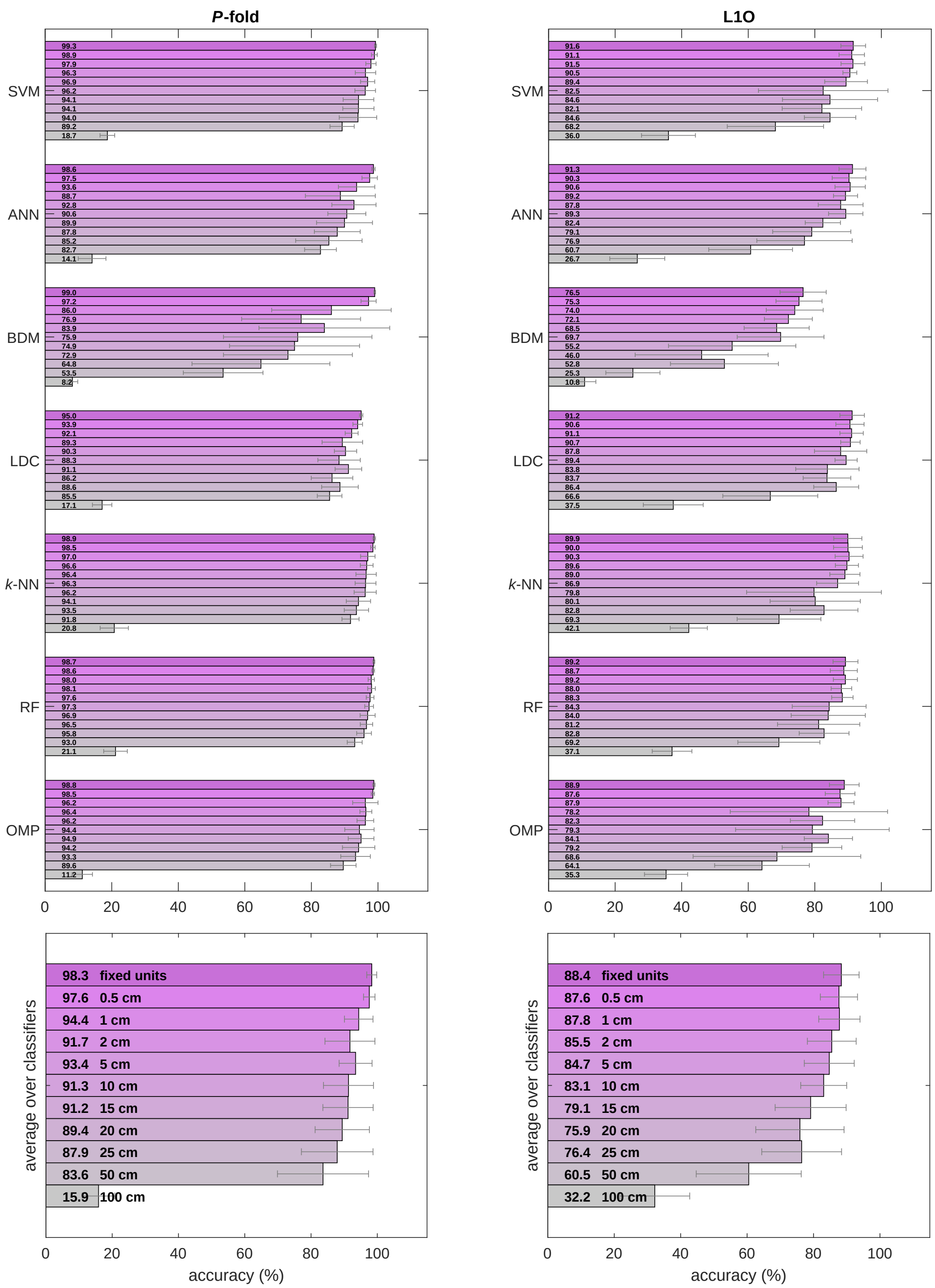


Figure 5.15: Activity recognition accuracy for the $\boldsymbol{\omega}\mathbf{m}\tilde{\mathbf{a}}$ approach for fixed and randomly displaced units with the RD-conc approach for (a) P-fold and (b) L1O cross validation.



(a) (b)

Figure 5.16: Activity recognition accuracy for the $\boldsymbol{\omega}\mathbf{m}\tilde{\mathbf{a}}$ approach for fixed and randomly displaced units with the RD-trun approach for (a) P-fold and (b) L1O cross validation.

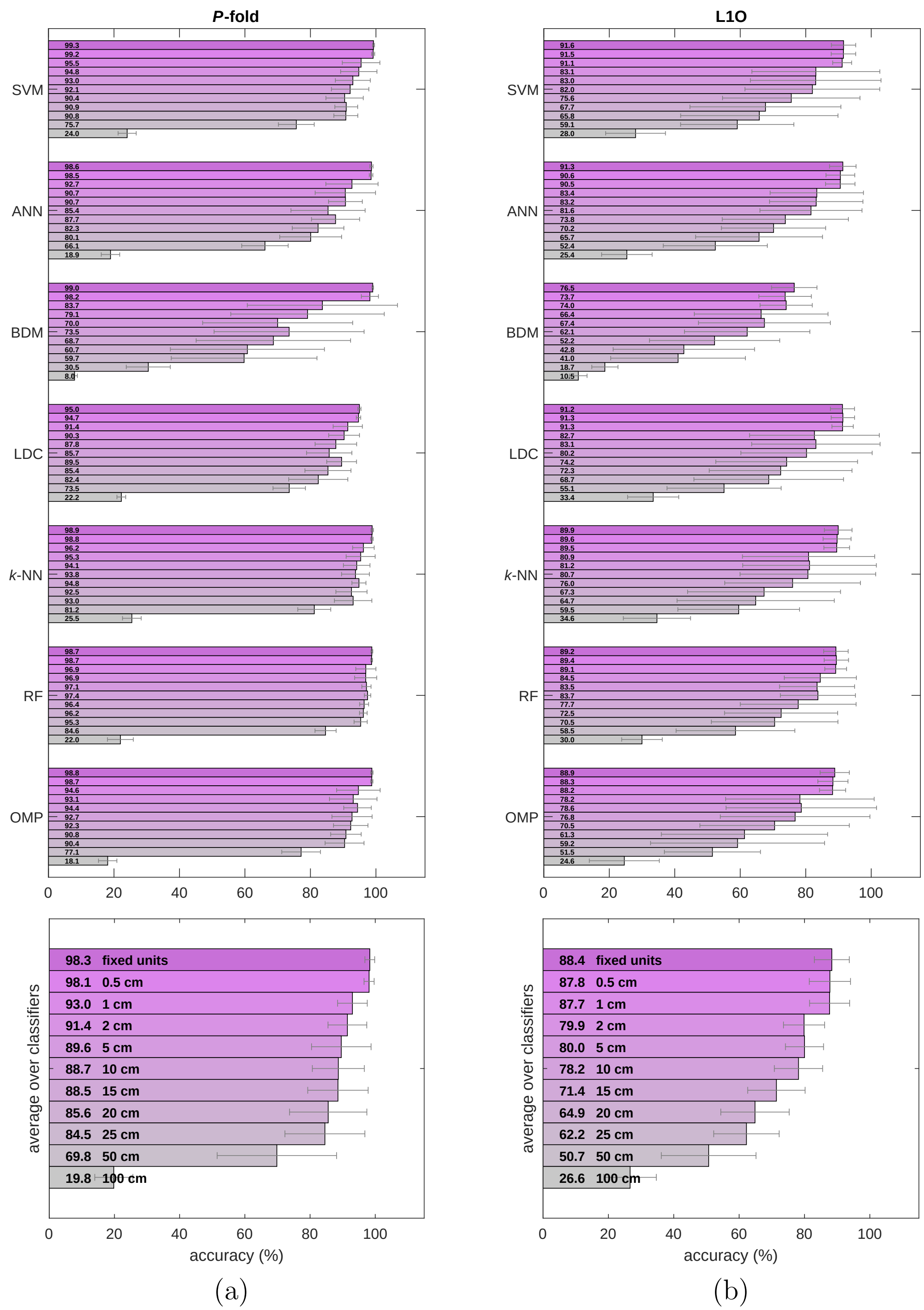


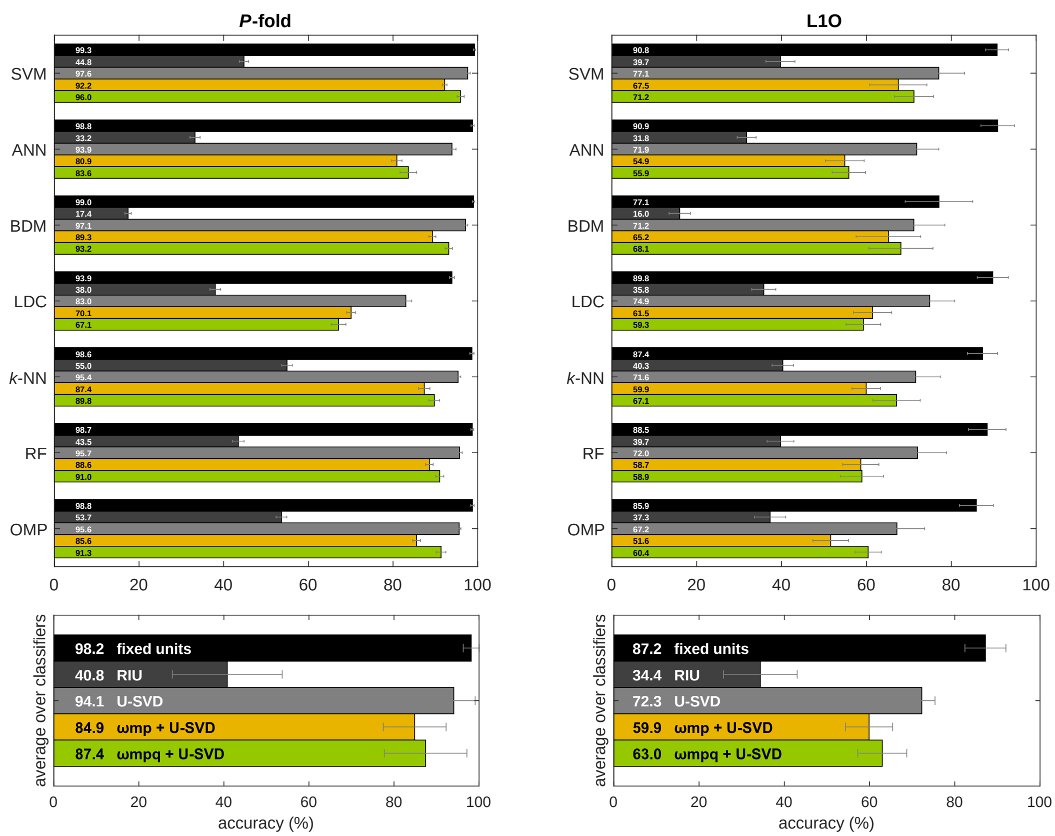
Figure 5.17: Activity recognition accuracy for the $\boldsymbol{\omega}\mathbf{m}\tilde{\mathbf{a}}$ approach for fixed and randomly displaced units with the RD-uni approach for (a) P-fold and (b) L1O cross validation.

5.2 Interchangeable Sensor Units

Many wearable systems require subjects to place more than one sensor unit at pre-determined positions on their body. This type of sensor configuration is obtrusive not only because they need to attach multiple sensor units but also because they need to identify the units to place each of them at its correct position on the body. The first level of flexibility provided for this purpose is to allow the units to be interchanged with each other and the second is to perform the classification based on a single unit. In this section, we consider the former, leaving the latter to Section 5.3, and propose a transformation technique called unit-based singular value decomposition (U-SVD) for interchangeable units. When U-SVD is applied to the sensor data in the pre-processing stage, the transformed data are no longer affected from the ordering of the sensor units.

5.2.1 Impact of Interchanged Sensor Units on the Activity Recognition Accuracy

When the sensor units are interchanged, the axes of the time-domain signal corresponding to different units are shuffled. This translates into a different ordering of the features in the feature vectors. When a test feature vector is obtained from sensor units that are ordered differently than the training data, the classification accuracy is expected to drop significantly because the indices of the features in the feature vectors will not match. To observe the impact of randomly interchanged sensor units on the activity recognition accuracy, we randomly interchange the time-domain sequences associated with the five sensor units with each other, independently for each 5-s time segment in the test data. We name this approach as randomly interchanged units (RIU) and provide its activity recognition accuracy in Figure 5.18. We observe that the accuracy of RIU abruptly decreases compared to the reference approach where the units are correctly ordered.



(a) (b)

Figure 5.18: Activity recognition accuracy for randomly interchanged sensor units (RIU) and the proposed U-SVD approach employed on its own and together with the $\boldsymbol{\omega}\mathbf{m}$p or $\boldsymbol{\omega}\mathbf{m}$**pq** approaches for (a) P-fold and (b) L1O cross validation.

5.2.2 Proposed Unit-Based SVD Method for Interchangeable Sensor Units

The proposed U-SVD transformation technique takes a linear combination of the time-domain sequences acquired by the different sensor units, independently for each 5-s time segment, so that interchanging the units during one time segment does not affect the transformed data at all. The U-SVD method comprises the following steps:

1. We normalize the time-domain sequences such that each sensor type (accelerometer, gyroscope, and magnetometer) has unit variance. The normalized sequences are respectively denoted by the column vectors $\hat{\mathbf{a}}[n]$, $\hat{\boldsymbol{\omega}}[n]$, and $\hat{\mathbf{m}}[n]$ of size 3×1 each, where $n = 1, \dots, N$ is the time sample index and N is the number of time samples in a time segment, which is 125 for our dataset. Each of the three vectors have x, y, z components, for instance, $\hat{\mathbf{a}}[n] = [\hat{a}_x[n], \hat{a}_y[n], \hat{a}_z[n]]^T$.

2. We form a data matrix $\mathbf{V}$ associated with each time segment as follows: With N_u being the number of units and $i = 1, \dots, N_u$ being the sensor unit index, we stack the measurements of each unit to form a row vector $\mathbf{v}_i$ of length $N_v = 3 \times 3 \times N$ as

$$\mathbf{v}_i = \left[\hat{\mathbf{a}}^T[1], \dots, \hat{\mathbf{a}}^T[N], \quad \hat{\boldsymbol{\omega}}^T[1], \dots, \hat{\boldsymbol{\omega}}^T[N], \quad \hat{\mathbf{m}}^T[1], \dots, \hat{\mathbf{m}}^T[N]\right] \tag{5.13}$$

   and vertically concatenate them to form a matrix of size $N_u \times N_v$:

$$\mathbf{V} = \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \vdots \\ \mathbf{v}_{N_u} \end{bmatrix} \tag{5.14}$$

   If the sensor units are interchanged, then the rows of $\mathbf{V}$ are re-ordered.

3. We decompose the matrix $\mathbf{V}$ into three matrices through the compact form of the SVD transformation [76] as $\mathbf{V} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{W}^T$ (see Section 2.2). Then, we

calculate the U-SVD transformation as

$$\mathcal{T}_{\text{U-SVD}} : \mathbf{V} \rightarrow \mathbf{\Sigma}\mathbf{W}^T \tag{5.15}$$

The transformed data matrix $\mathbf{V}_{\text{U-SVD}} \triangleq \mathbf{\Sigma}\mathbf{W}^T$ has the same size $(N_u \times N_v)$ as $\mathbf{V}$.

4. We separate the data contained in the transformed matrix $\mathbf{V}_{\text{U-SVD}}$ by reversing the operations performed in Step 2 so that the transformed data have the same format as the raw data and can be input to the standard activity recognition scheme without making any modifications.

Since $\mathbf{V}_{\text{U-SVD}} = \mathbf{U}^T\mathbf{V}$, each row of $\mathbf{V}_{\text{U-SVD}}$ is a linear combination of the rows $\mathbf{v}_i$ of $\mathbf{V}$, each of which is associated with a unique sensor unit. The matrix $\mathbf{U}$ that contains the linear combination coefficients is calculated by SVD in Step 3 such that the rows of the transformed data matrix $\mathbf{V}_{\text{U-SVD}}$ are the projections of $\mathbf{v}_i$ onto the principal axes in the N_u-dimensional space. When the rows $\mathbf{v}_i$ of $\mathbf{V}$ are re-ordered as a result of interchanging the sensor units, the projections onto the principal axes remain the same, so does the matrix $\mathbf{V}_{\text{U-SVD}}$. Therefore, the U-SVD transformation is invariant to the interchanging of the sensor units. U-SVD is analogous to the method proposed in Section 2.2 where the x, y, z axes of the tri-axial sensors are projected on their principal axes to achieve robustness to sensor unit orientations.

We apply the U-SVD transformation independently to each time segment. In this way, we allow the sensor units to be interchanged differently in each time segment. We need to apply U-SVD to both the training and test data to match them with each other for accurate classification. In this way, we allow the units to be interchanged differently in each time segment of both the training and the test data.

The activity recognition accuracy for the U-SVD approach is shown in Figure 5.18. U-SVD obtains a much higher accuracy than RIU when the units are randomly interchanged. Compared with the reference approach, allowing the units to be interchanged decreases the accuracy as expected; however, the reduction is relatively small: 4.1% for P-fold and 14.9% for L1O cross validation.

5.2.3 Interchangeable Sensor Units with Position Invariance within the Same Body Part

To allow interchangeable sensor units and at the same time achieve position invariance within the same body part, we apply one of the transformations $\boldsymbol{\omega}\mathbf{m}$p and $\boldsymbol{\omega}\mathbf{m}p\mathbf{q}$ followed by U-SVD. In applying U-SVD, we no longer have the original sensor sequences as they are transformed beforehand using the $\boldsymbol{\omega}\mathbf{m}$p (or $\boldsymbol{\omega}\mathbf{m}p\mathbf{q}$) approach. The new sequences are $\boldsymbol{\omega}$, $\mathbf{m}$, p (and possibly $\mathbf{q}$) with dimensions three, three, one (and four) and we treat them in the same way as we treated the data from three different sensor types when applying the U-SVD transformation.

The accuracy values for the approaches where the U-SVD transformation is applied together with $\boldsymbol{\omega}\mathbf{m}$p or $\boldsymbol{\omega}\mathbf{m}p\mathbf{q}$ are shown in Figure 5.18. We observe that both combinations achieve accuracies close to the stand-alone use of U-SVD for P-fold cross validation, whereas there is a noticeable drop in the accuracy for L1O. In particular, compared to U-SVD, the approach "$\boldsymbol{\omega}\mathbf{m}p\mathbf{q}$ + U-SVD" causes an accuracy reduction of only 6.7% and 9.3% for P-fold and L1O, respectively. Therefore, we may achieve position invariance within the same body part by also allowing interchangeable sensor units with a reasonable drop in the classification accuracy.

5.3 Classification Based on a Single Sensor Unit with or without Position Invariance within the Same Body Part

In this section, we consider the scenario named single-unit classification (SUC) where the training data are collected from multiple sensor units attached at different positions on different body parts and activity recognition is performed based on a single unit that is placed at one of these positions. The system is trained for all the sensor unit positions that are available in the dataset and does not use the information about at which of them the unit is placed in the test phase. This flexibility makes the system less obtrusive since the user may attach the unit at any preferred position and does not need to provide the position information by any means.

We consider three different approaches for this purpose:

- **SUC-I:** In this approach, we train the classifiers in a generalized way as follows (see Section 1.1.1.2): Let $\mathbf{f}_i^{(j)}$ be a column vector that contains the features extracted from unit i in the jth time segment, where $i \in \{1, \ldots, N_u\}$ and $j \in \{1, \ldots, N_s\}$ with N_s being the total number of segments in the training set. In the reference approach, we stack the features associated with all the sensor units as

$$\mathbf{f}^{(j)} = \left[\left(\mathbf{f}_1^{(j)} \right)^T, \left(\mathbf{f}_2^{(j)} \right)^T, \ldots, \left(\mathbf{f}_{N_u}^{(j)} \right)^T \right]^T \tag{5.16}$$

  and form the training set as $\mathscr{T}_{\text{reference}} = \left\{ \mathbf{f}^{(1)}, \mathbf{f}^{(2)}, \ldots, \mathbf{f}^{(N_s)} \right\}$ which contains N_s training vectors. However, in generalized training, we use the features extracted from each unit as a separate training instance and form the training set as

$$\mathscr{T}_{\text{SUC-I}} = \left\{ \mathbf{f}_1^{(1)}, \mathbf{f}_2^{(1)}, \ldots, \mathbf{f}_{N_u}^{(1)}, \quad \mathbf{f}_1^{(2)}, \mathbf{f}_2^{(2)}, \ldots, \mathbf{f}_{N_u}^{(2)}, \quad \ldots, \quad \mathbf{f}_1^{(N_s)}, \mathbf{f}_2^{(N_s)}, \ldots, \mathbf{f}_{N_u}^{(N_s)} \right\}. \tag{5.17}$$

  In this way, we have $N_u \times N_s$ training feature vectors, which is N_u times more than the reference approach and the vectors have N_u times smaller length compared to the reference case.

  In the test phase, we perform the classification based on a single sensor unit; hence, we have separate test feature vectors associated with each unit as in the training set. Using the generalized classifier, we classify the activity type separately for each test feature vector without using the information about which position it is associated with. Since the training set consists of feature vectors associated with all the positions that are available in the dataset, we expect the classifier to match one of these training feature vectors to the given test feature vector obtained from a single sensor unit position. This is not an easy task because of the variation of the data within the activity classes, especially for L1O.

- **SUC-II:** We perform activity recognition in two steps, as proposed in [59]: In the first step, we classify the sensor unit's position among the positions that are available in the dataset. We follow the same classification scheme as in activity recognition.

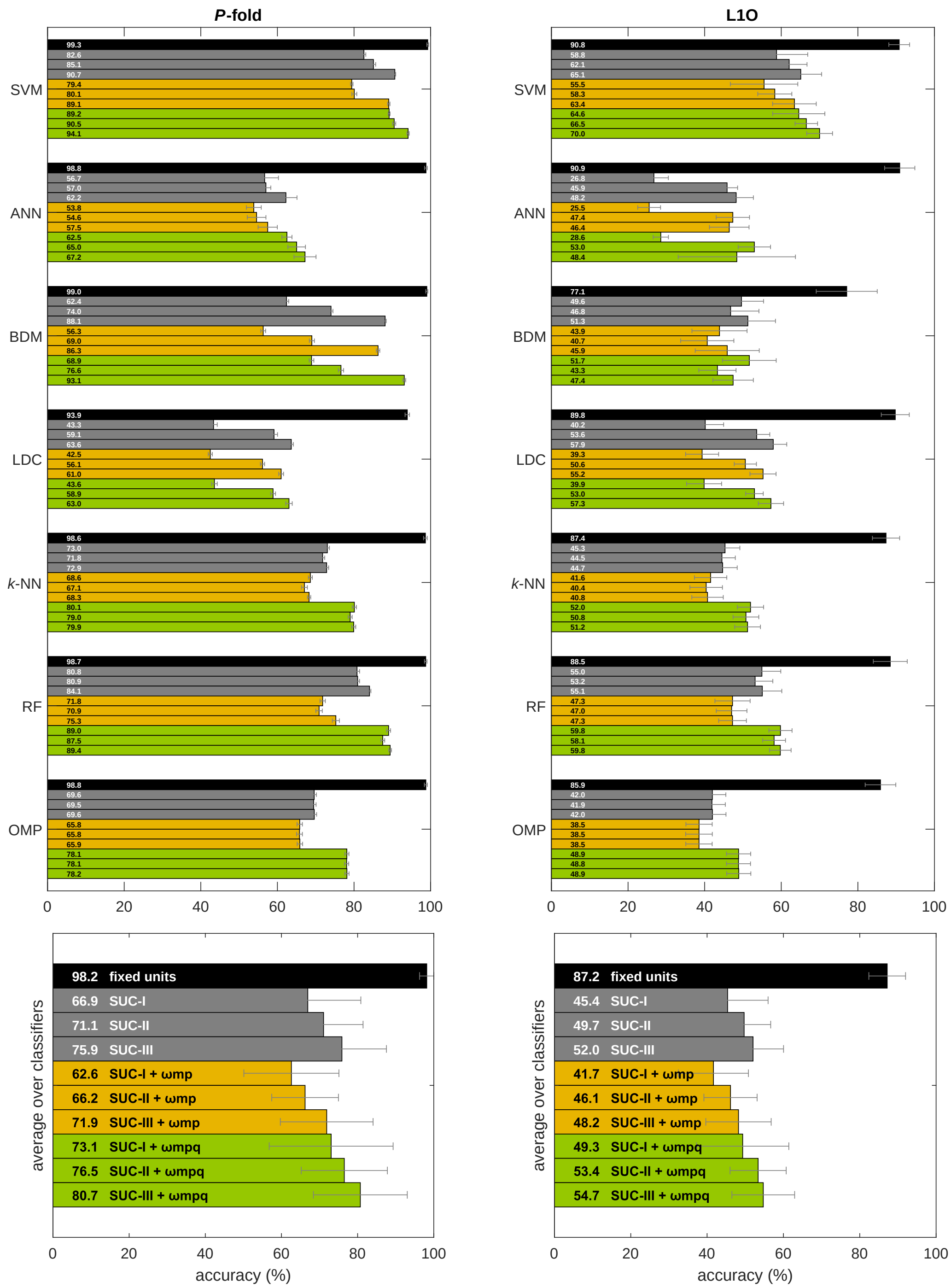
  For the second step, we train a different activity classifier that is specialized for each unit position. The training set for unit position i is formed as

  $$\mathscr{T}_{\text{SUC-II}, i} = \left\{ \mathbf{f}_i^{(1)}, \mathbf{f}_i^{(2)}, \ldots, \mathbf{f}_i^{(N_s)} \right\} \tag{5.18}$$

  where $i \in \{1, \ldots, N_u\}$. Based on the unit classification result obtained in the first step, we select the activity classifier trained for that specific unit and then recognize the activity.

- **SUC-III:** In this approach, we consider simultaneous position and activity classification [57]. For this purpose, we treat each sensor unit position associated with each activity as a different class so that the number of classes is $N_u \times N_a$ where N_a denotes the number of activities. We may associate the estimated classes with the unit positions and activities; hence, this method simultaneously classifies both of them. We present here only the activity classification accuracy results since classifying sensor unit positions is not the aim of this study.

The activity recognition accuracies for the SUC-I, SUC-II, and SUC-III approaches are comparatively provided in Figure 5.19. All three SUC approaches obtain accuracies that are considerably lower than the reference approach, as expected. The accuracy drop is smaller in P-fold than L1O cross validation. The SUC-III approach obtains the highest accuracy.



(a) (b)

Figure 5.19: Activity recognition accuracy for single-unit classification (SUC) employed on its own and together with the $\boldsymbol{\omega}\mathbf{m}$p or $\boldsymbol{\omega}\mathbf{m}$pq approaches for (a) *P*-fold and (b) L1O cross validation.

To achieve position invariance within the same body part, we apply each of the three SUC approaches on the data transformed using the methods $\boldsymbol{\omega}\mathbf{m}p$ and $\boldsymbol{\omega}\mathbf{m}p\mathbf{q}$. This scheme has only the following requirement:

> Only one sensor unit needs to be placed at the pre-determined orientation on one of the body parts on which the sensor units are placed during the training phase.

Our dataset captures the movements of the right and left lower arm, right and left upper leg, and torso (see Section 3.4 and the second column of Table 2.1); thus, it is sufficient for the user to place the unit on one of these body parts at the pre-determined orientation.

Referring to the activity recognition accuracies given in Figure 5.19, we observe that applying the transformation $\boldsymbol{\omega}\mathbf{m}p$ decreases the accuracy for the SUC approaches. This is expected because $\boldsymbol{\omega}\mathbf{m}p$ allows more flexibility in the positioning of the sensor units. On the other hand, the $\boldsymbol{\omega}\mathbf{m}p\mathbf{q}$ approach surprisingly improves the accuracy when combined with the SUC approaches even with respect to using each SUC method on its own.

5.4 Run-Time Analysis

The run times of the data transformation techniques are provided in Table 5.1 as the average values per 5-s time segment. The processing was performed on 64-bit MATLAB® R2018b running on a laptop computer whose specifications are provided in Section 2.5. Among the position-invariant transformations, the proposed $\boldsymbol{\omega}\mathbf{m}p$ approach is computationally more efficient than the existing approach $\boldsymbol{\omega}\mathbf{m}\tilde{\mathbf{a}}$, whereas the second proposed approach $\boldsymbol{\omega}\mathbf{m}p\mathbf{q}$ takes the longest to execute. The U-SVD transformation that is proposed for the interchangeability of the units runs faster when it is applied together with $\boldsymbol{\omega}\mathbf{m}p$ and slower when it is applied together with $\boldsymbol{\omega}\mathbf{m}p\mathbf{q}$ because of the varying dimension of time-domain data. All of the run times in the table are much shorter than the duration of the time segments, thus, can be executed in near real time.

Table 5.1: Average run times of the transformation techniques per 5-s time segment.

data transformation technique	**run time per 5-s time segment (ms)**
$\boldsymbol{\omega}\mathbf{m}$p	2.57
$\boldsymbol{\omega}\mathbf{m}p\mathbf{q}$	11.19
$\boldsymbol{\omega}\mathbf{m}\tilde{\mathbf{a}}$	6.43
U-SVD	18.00
$\boldsymbol{\omega}\mathbf{m}$p + U-SVD	12.85
$\boldsymbol{\omega}\mathbf{m}p\mathbf{q}$ + U-SVD	38.79

Table 5.2 shows the run times of the classifiers in terms of their averages and standard deviations over the following transformation techniques: fixed units, $\boldsymbol{\omega}\mathbf{m}$p, $\boldsymbol{\omega}\mathbf{m}p\mathbf{q}$, $\boldsymbol{\omega}\mathbf{m}\tilde{\mathbf{a}}$, U-SVD, $\boldsymbol{\omega}\mathbf{m}$p + U-SVD, $\boldsymbol{\omega}\mathbf{m}p\mathbf{q}$ + U-SVD, SUC-I, SUC-II, SUC-III, $\boldsymbol{\omega}\mathbf{m}$p + SUC-I, $\boldsymbol{\omega}\mathbf{m}$p + SUC-II, SUC-III + $\boldsymbol{\omega}\mathbf{m}$p, $\boldsymbol{\omega}\mathbf{m}p\mathbf{q}$ + SUC-I, $\boldsymbol{\omega}\mathbf{m}p\mathbf{q}$ + SUC-II, and $\boldsymbol{\omega}\mathbf{m}p\mathbf{q}$ + SUC-III. Table 5.2(a) and (b) contain the total run time (including the training phase, classification of all test feature vectors, and programming overheads) and the training time, respectively, both provided in seconds for an average cross-validation iteration. We observe that, in terms of the total run time, k-NN is the fastest and OMP is the slowest among the seven classifiers. These two classifiers do not have an execution in the training phase other than the storage of the training feature vectors, whereas the RF classifier takes the longest to train. Table 5.2(c) contains the average classification time in milliseconds per single test feature vector associated with a 5-s time segment. Although all of the classifiers can label a test feature vector in a duration much shorter than the associated time segment, the ANN and LDC classifiers perform this operation almost instantly, whereas the OMP classifier is more than two orders of magnitude slower than the others.

Table 5.2: (a) Total run time (including training and classification of all test feature vectors) and (b) training time in an average L1O iteration. (c) Average classification time of a single test feature vector. The run times are shown as the average values plus/minus one standard deviation over the following transformation techniques: fixed units, $\boldsymbol{\omega}\mathbf{m}$p, $\boldsymbol{\omega}\mathbf{m}p\mathbf{q}$, $\boldsymbol{\omega}\mathbf{m}\tilde{\mathbf{a}}$, U-SVD, $\boldsymbol{\omega}\mathbf{m}$p + U-SVD, $\boldsymbol{\omega}\mathbf{m}p\mathbf{q}$ + U-SVD, SUC-I, SUC-II, SUC-III, $\boldsymbol{\omega}\mathbf{m}$p + SUC-I, $\boldsymbol{\omega}\mathbf{m}$p + SUC-II, $\boldsymbol{\omega}\mathbf{m}$p + SUC-III, $\boldsymbol{\omega}\mathbf{m}p\mathbf{q}$ + SUC-I, $\boldsymbol{\omega}\mathbf{m}p\mathbf{q}$ + SUC-II, and $\boldsymbol{\omega}\mathbf{m}p\mathbf{q}$ + SUC-III.

	(a)	**(b)**	**(c)**
classifier	**total run time (s)**	**training time (s)**	**classification time (ms)**
SVM	8.77 ± 2.70	7.25 ± 2.33	0.33 ± 0.11
ANN	5.43 ± 2.10	5.42 ± 2.10	0.01 ± 0.00
BDM	1.41 ± 0.11	0.01 ± 0.00	1.37 ± 0.11
LDC	1.59 ± 0.40	0.28 ± 0.01	0.03 ± 0.00
k-NN	0.13 ± 0.02	–	0.12 ± 0.02
RF	23.20 ± 4.07	20.33 ± 3.63	0.80 ± 0.07
OMP	200.74 ± 279.72	–	194.78 ± 271.13

5.5 Concluding Remarks

This chapter has focused on the positioning of wearable sensor units. We have proposed a number of methods that allow the subjects to wear each sensor unit at different positions within a pre-determined body part or across different body parts. We have also developed techniques to recognize activities by using a single sensor unit that is placed at an arbitrary position, based on training data acquired from multiple units. We have comparatively evaluated these approaches using a publicly available dataset containing daily and sports activities which are much more complex and larger in number than those in existing studies. We have employed seven state-of-the-art classifiers and two cross-validation techniques to demonstrate the robustness of our methodology. We have observed a trade-off between the flexibility in sensor unit placement and the classification accuracy.

Chapter 6

Simultaneous Invariance to Sensor Unit Position and Orientation

In this chapter, we simultaneously achieve position and orientation invariance by applying the position-invariant techniques that are proposed in Chapter 5 and the orientation-invariant approaches proposed in Chapter 3. We employ the same dataset (dataset A), activity recognition methodology, and cross-validation techniques as in Chapter 5.

6.1 Simultaneous Position and Orientation Invariance within the Same Body Part

In this section, we analyze the effects of differently placed sensor units within the same body part and propose a method to simultaneously achieve position and orientation invariance.

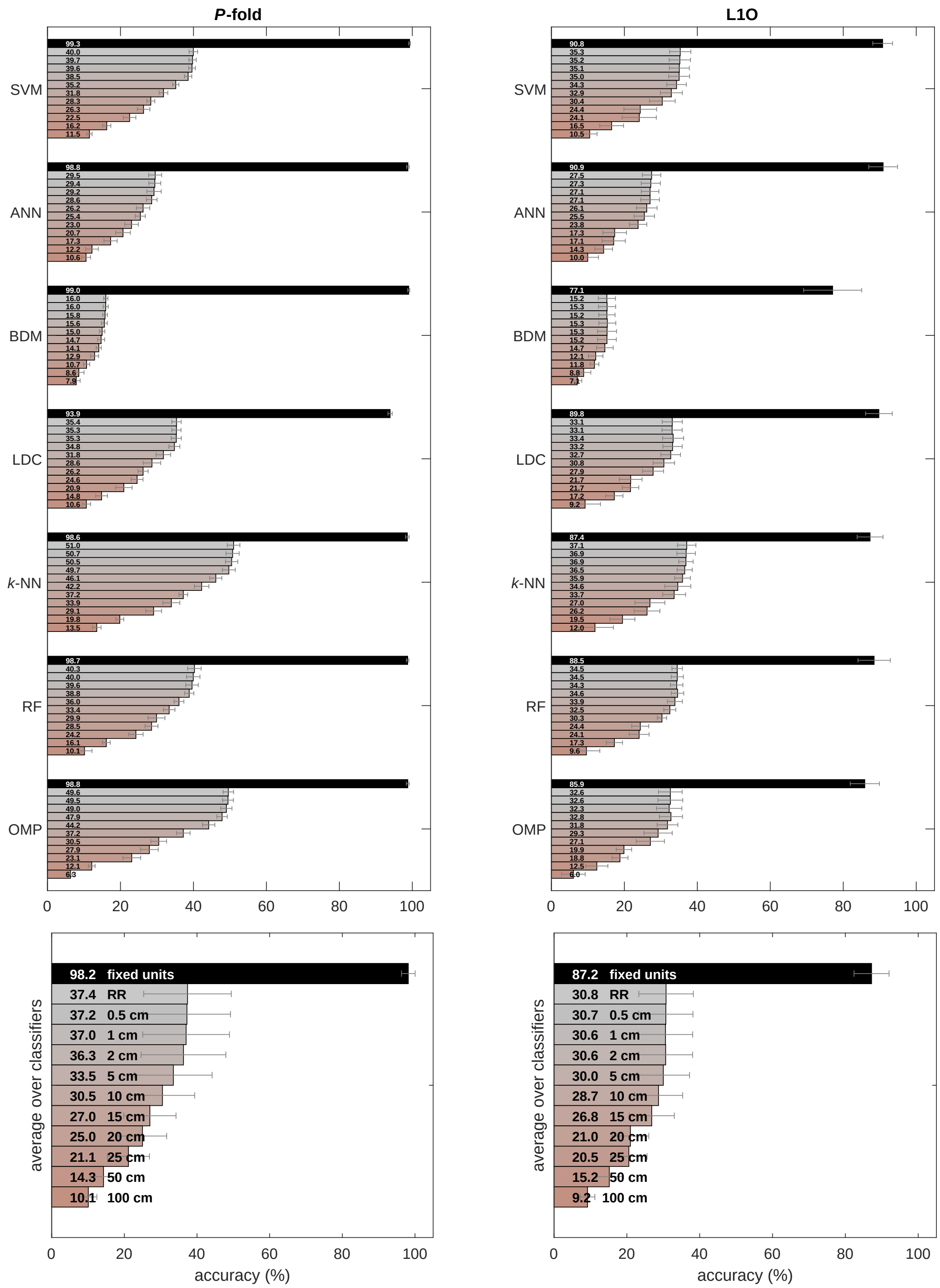
6.1.1 Impact of Sensor Unit Positioning within the Same Body Part on the Activity Recognition Accuracy

To simulate randomly positioned and oriented sensors within the same body part, we first displace the sensor units using the RD-conc, RD-trun, and RD-uni simulation models that are explained in Section 5.1.1. Then, we randomly rotate (RR) the sensor data as described in Section 3.4.1.1. These transformations simulate the case where each sensor unit is placed at a random position and orientation within a disk that is coincident with the surface where the unit makes contact with the body. The classifiers do not learn the effects of the transformations in the training phase because we apply both of the RD and RR transformations to each time segment in the test data only, which demonstrates a real-world scenario.

The activity recognition accuracy is shown for the fixed and randomly rotated sensor units as well as for both randomly rotated and displaced units using the RD-conc, RD-trun, and RD-uni approaches in Figures 6.1–6.3, respectively. Randomly rotating the units decreases the accuracy by more than 55% compared to the fixed units. When the units are also displaced, the accuracy decreases further. The drop in the accuracy increases with the displacement distance, as expected. The degradation in the accuracy caused by RD-uni is more apparent than RD-conc and RD-trun.

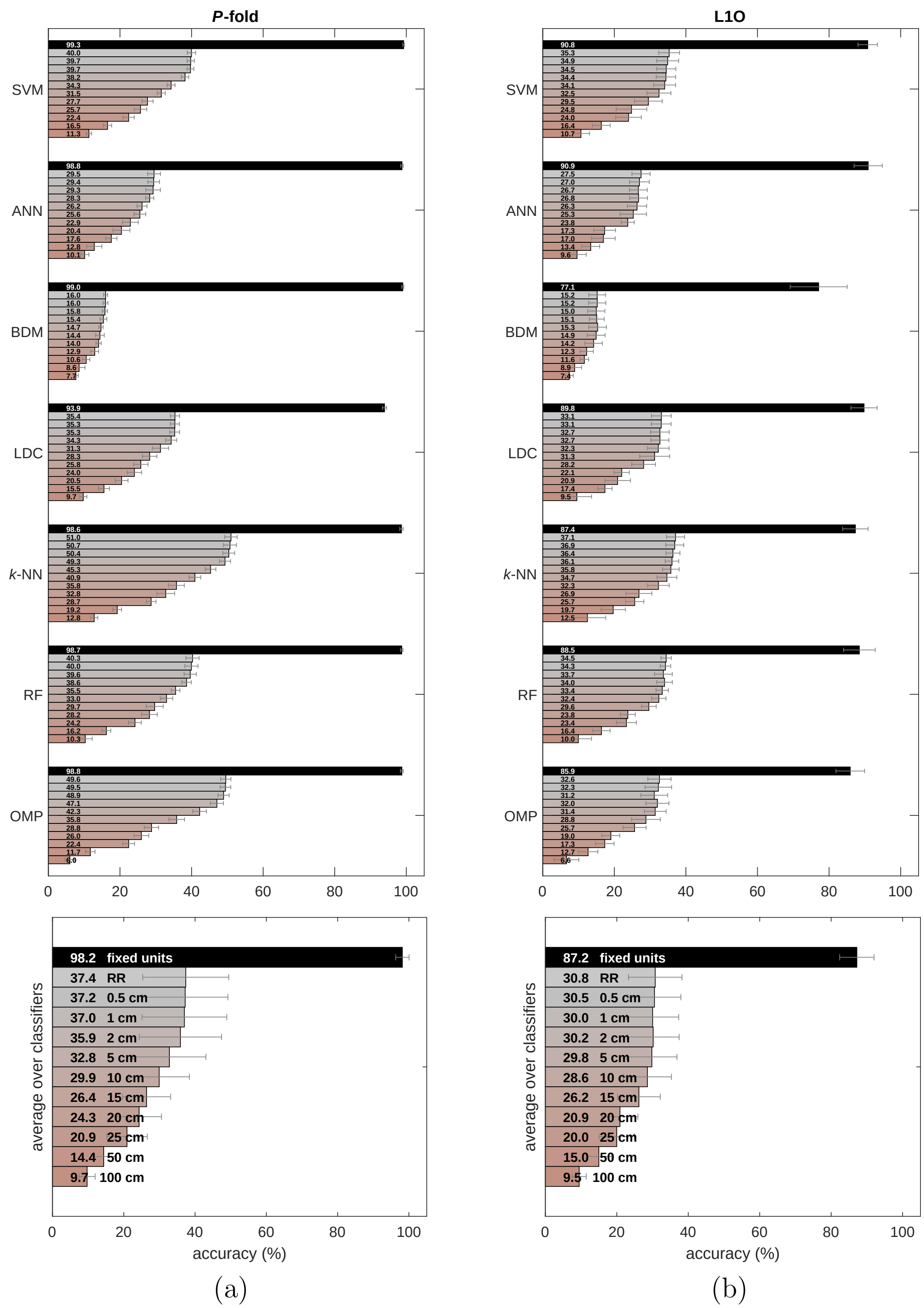
6.1.2 Proposed Method for Position and Orientation Invariance within the Same Body Part

To allow orientation invariance in addition to position invariance within the same body part, we replace the sensor sequences and the extracted position-invariant features that are used in Section 5.1.2 with their orientation-invariant counterparts. For this purpose, we first estimate sensor unit orientation using the method proposed in Section 4.2. Based on the estimated orientation, we represent the position-invariant sensor sequences $\boldsymbol{\omega}$ and $\mathbf{m}$ as well as the position-invariant quantity p in the Earth frame, denoting them with the superscript E as in



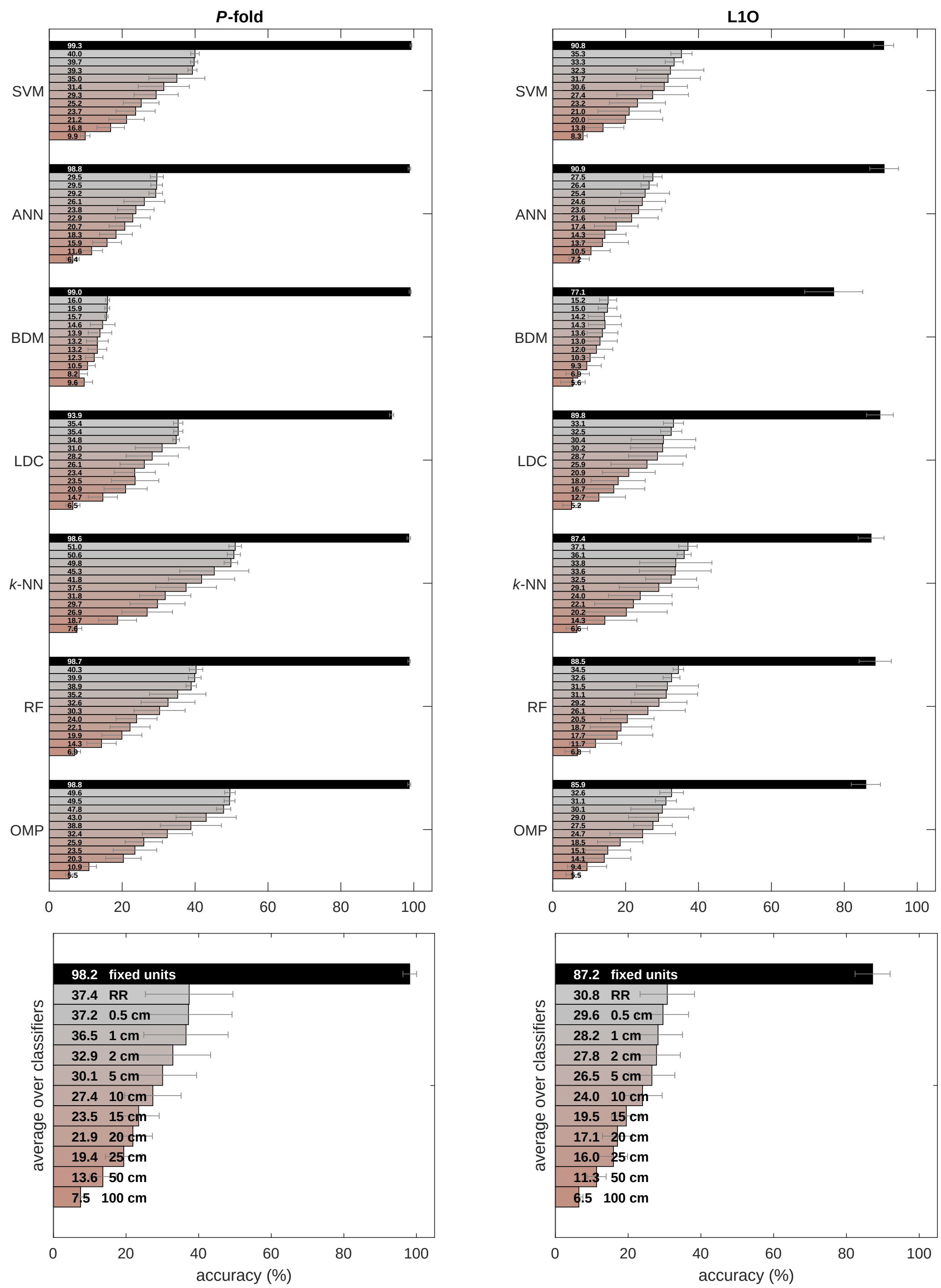
(a) (b)

Figure 6.1: Activity recognition accuracy for fixed and randomly rotated (RR) units as well as both randomly rotated and displaced units with the RD-conc approach for (a) P-fold and (b) L1O cross validation. The lengths of the bars indicate the accuracy values. The thin sticks represent plus/minus one standard deviation over the cross-validation iterations and over the classifiers at the top and bottom parts of the figure, respectively.



(a) (b)

Figure 6.2: Activity recognition accuracy for fixed and randomly rotated (RR) units as well as both randomly rotated and displaced units with the RD-trun approach for (a) P-fold and (b) L1O cross validation.



(a) (b)

Figure 6.3: Activity recognition accuracy for fixed and randomly rotated (RR) units as well as both randomly rotated and displaced units with the RD-uni approach for (a) P-fold and (b) L1O cross validation.

Chapter 3. We use the differential sensor quaternion $\mathbf{q}^{\text{diff}}$ (described in Section 3.3) instead of the orientation quaternion $\mathbf{q}$. Since the quantities $\boldsymbol{\omega}^E$, $\mathbf{m}^E$, p^E, and $\mathbf{q}^{\text{diff}}$ do not depend on the orientation at which the units are worn on the body, we ensure that the approaches that employ the combinations of these quantities are invariant to the sensor unit orientation.

The activity recognition results for the approaches $(\boldsymbol{\omega}\mathbf{m})^E$ and $(\boldsymbol{\omega}\mathbf{m}\mathrm{p})^E\,\mathbf{q}^{\text{diff}}$, both of which are position and orientation invariant, are provided in Figure 6.4. Compared to the case where the units are correctly placed, the $(\boldsymbol{\omega}\mathbf{m}\mathrm{p})^E\,\mathbf{q}^{\text{diff}}$ approach achieves only 3.2% and 6.2% lower accuracy values for P-fold and L1O cross-validation techniques, respectively, whereas the degradation caused by the $(\boldsymbol{\omega}\mathbf{m})^E$ approach is significantly higher. Comparing Figure 6.4 with Figures 6.1–6.3 reveals that both of the $(\boldsymbol{\omega}\mathbf{m})^E$ and $(\boldsymbol{\omega}\mathbf{m}\mathrm{p})^E\,\mathbf{q}^{\text{diff}}$ approaches obtain an accuracy much higher than RR that is employed on its own or together with RD.

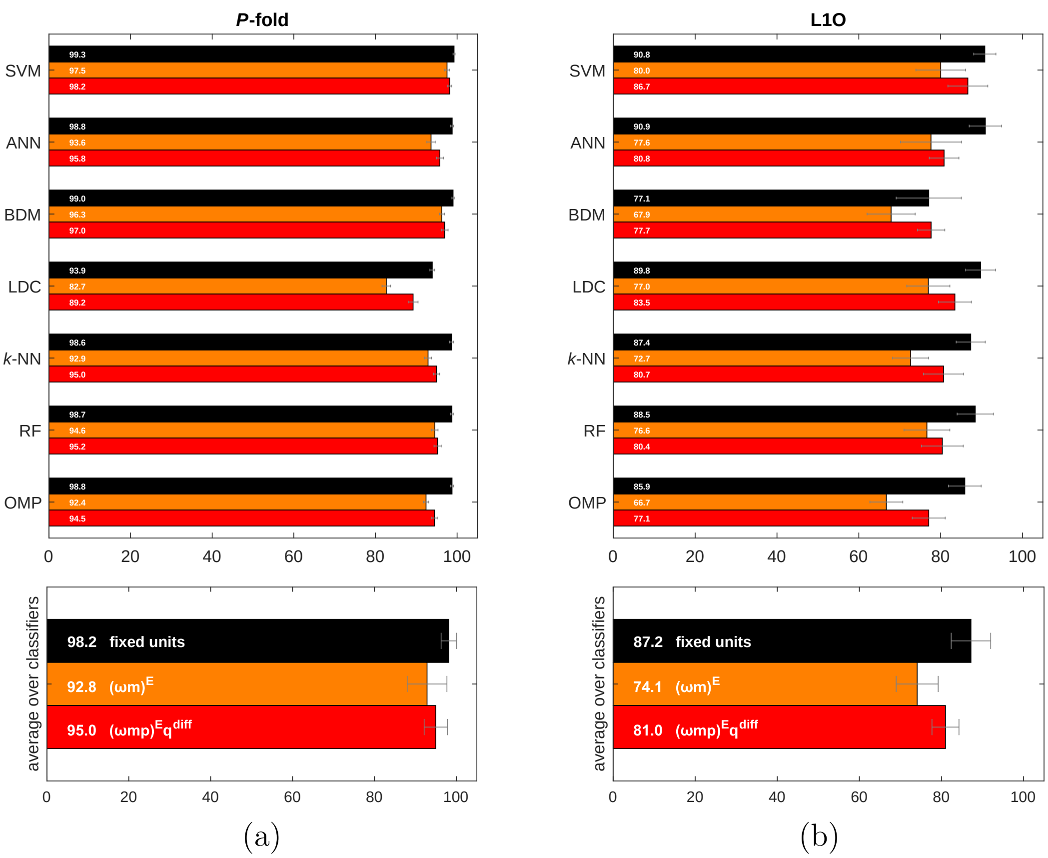


Figure 6.4: Activity recognition accuracy for the $(\boldsymbol{\omega}\mathbf{m})^E$ and $(\boldsymbol{\omega}\mathbf{m}\mathrm{p})^E\,\mathbf{q}^{\text{diff}}$ approaches for (a) P-fold and (b) L1O cross validation.

6.2 Position and Orientation Invariance within the Same Body Part with Interchangeable Sensor Units

We have developed the U-SVD transformation to allow the sensor units to be interchanged with each other, as explained in Section 5.2.2. We have also combined it with the approaches $\boldsymbol{\omega}\mathbf{m}$ and $\boldsymbol{\omega}\mathbf{m}\mathrm{p}\mathbf{q}$ to additionally achieve position invariance within the same body part in Section 5.2.3. In this section, we allow the units to be placed at any orientation as well. For this purpose, we apply the simultaneously position- and orientation-invariant approach $(\boldsymbol{\omega}\mathbf{m}\mathrm{p})^E\,\mathbf{q}^{\mathrm{diff}}$ together with the U-SVD transformation for interchangeable units. In utilizing these transformations, we first calculate $(\boldsymbol{\omega}\mathbf{m}\mathrm{p})^E\,\mathbf{q}^{\mathrm{diff}}$ and then apply the U-SVD by taking the sensor type dimensions as three, three, one (and four) in the first step of U-SVD (see Section 5.2.2).

Referring to the activity recognition accuracies that are provided in Figure 6.5, the proposed $(\boldsymbol{\omega}\mathbf{m}\mathrm{p})^E\,\mathbf{q}^{\mathrm{diff}}$ + U-SVD approach obtains an acceptable accuracy for P-fold cross validation, which is about 40% higher than RIU (see Section 5.2.1) and 17% lower than fixed units. The proposed approach brings an improvement to the accuracy compared to RIU for L1O cross validation as well, although it is less effective in L1O than P-fold. We also observe in Figure 6.5 that allowing the units to be interchanged using U-SVD degrades the accuracy more than the position invariance within the same body part achieved by $(\boldsymbol{\omega}\mathbf{m}\mathrm{p})^E\,\mathbf{q}^{\mathrm{diff}}$, although the difference is small for P-fold. Allowing both of the flexibilities by using $(\boldsymbol{\omega}\mathbf{m}\mathrm{p})^E\,\mathbf{q}^{\mathrm{diff}}$ + U-SVD further degrades the accuracy, as expected, because the only requirement for the user is to place exactly one sensor unit at any position and orientation on each of the body parts on which the sensor units are placed in the dataset.

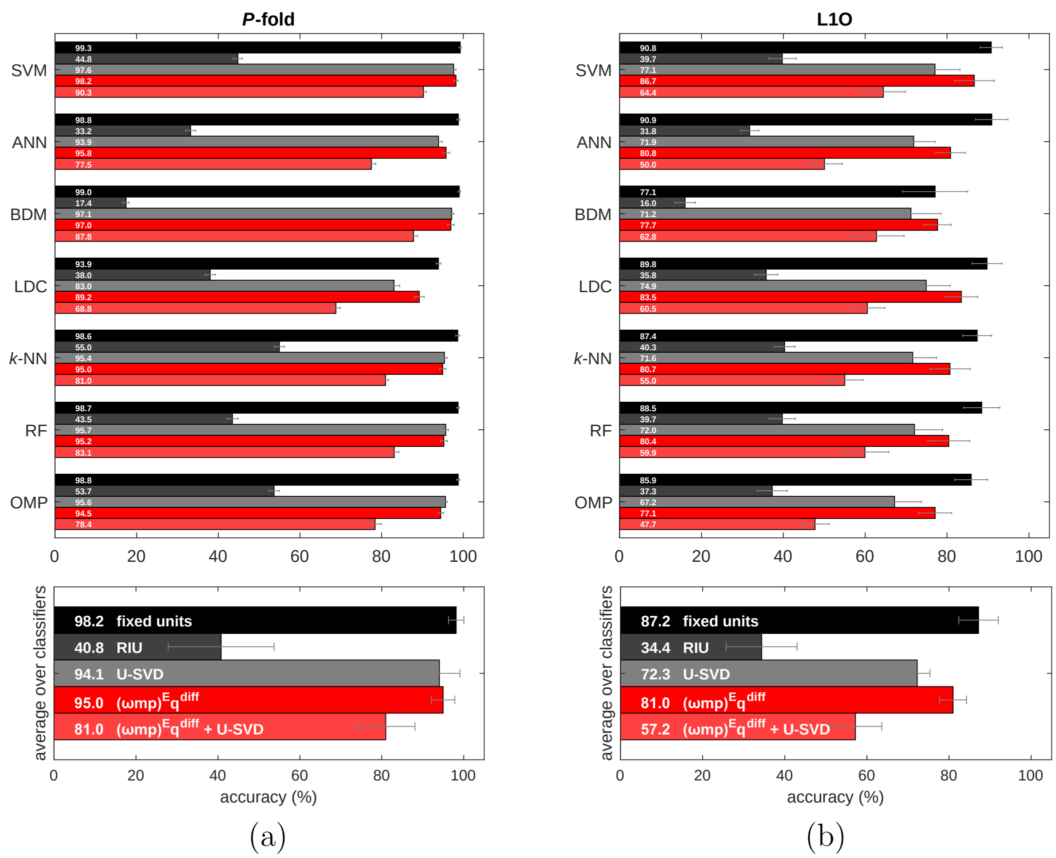


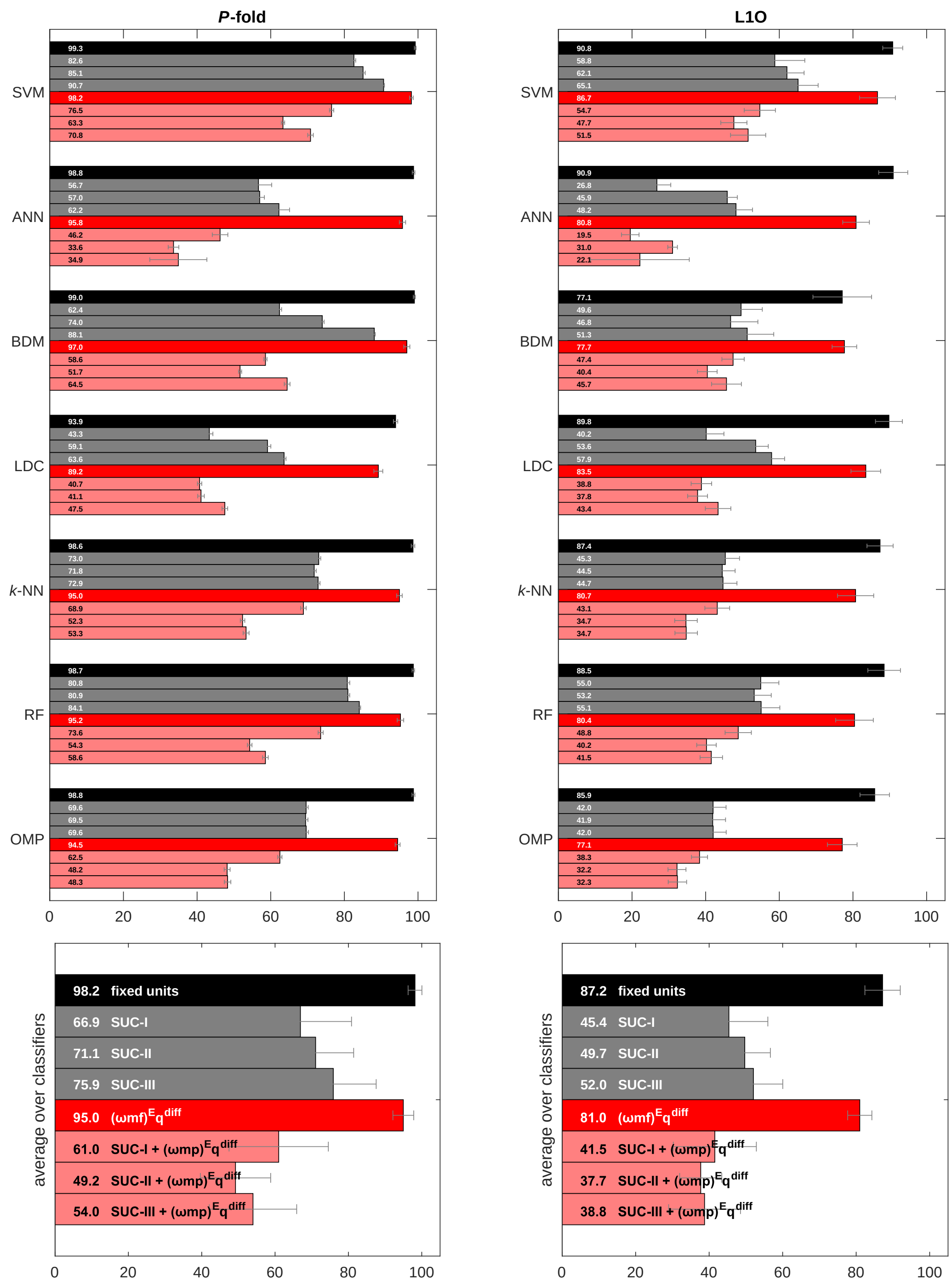
Figure 6.5: Activity recognition accuracy randomly interchanged sensor units (RIU) as well as the proposed U-SVD and $(\boldsymbol{\omega}\mathbf{m}\mathrm{p})^{E}\,\mathbf{q}^{\mathrm{diff}}$ approaches that are employed on their own and simultaneously for (a) P-fold and (b) L1O cross validation.

6.3 Position and Orientation Invariance within the Same Body Part with Single-Unit Classification

We combine the SUC methods that are explained in Section 5.3 with the simultaneously position- and orientation-invariant transformation $(\boldsymbol{\omega}\mathbf{m}\mathrm{p})^{E}\,\mathbf{q}^{\mathrm{diff}}$ that is described in Section 5.1.2 to classify the activities based on a single sensor unit that is placed at any position and orientation on one of the body parts included in the dataset. According to the accuracies provided in Figure 6.6, applying the transformation $(\boldsymbol{\omega}\mathbf{m}\mathrm{p})^{E}\,\mathbf{q}^{\mathrm{diff}}$ in addition to the SUC approaches significantly degrades the accuracy compared to fixed sensor units. However, the accuracy values are still well above random decision making, which has an average accuracy of $1/19 = 5.3\%$ for 19 classes. The SUC-I approach performs better than SUC-II and SUC-III when applied together with $(\boldsymbol{\omega}\mathbf{m}\mathrm{p})^{E}\,\mathbf{q}^{\mathrm{diff}}$ whereas SUC-III is the most accurate among the three when they are employed on their own without considering position or orientation invariance.

6.4 Run-Time Analysis

The run times of the simultaneously position- and orientation-invariant techniques for an average 5-s time segment are provided in Table 6.1. Specifications of the device on which the processing was performed are provided in Section 5.4. In all of the approaches, the acquired data and the calculated quantities are represented in the Earth frame. This representation requires the estimation of sensor unit orientations, which takes most of the run time (see Section 4.4). The approach $(\boldsymbol{\omega}\mathbf{m}\mathrm{p})^{E}\,\mathbf{q}^{\mathrm{diff}}$ + U-SVD has a longer run time than the other two because of the calculation of the SVD transformation. Nevertheless, all of the run times are much shorter than the time segment duration (5 s) and can be executed in near real time.



(a) (b)

Figure 6.6: Activity recognition accuracy for single-unit classification (SUC) employed on its own and together with the $(\boldsymbol{\omega}\mathbf{m}\mathrm{p})^{E}\,\mathbf{q}^{\mathrm{diff}}$ approach for (a) P-fold and (b) L1O cross validation.

Table 6.1: Average run times of the transformation techniques per 5-s time segment.

data transformation technique	**run time per 5-s time segment (ms)**
$(\boldsymbol{\omega}\mathbf{m})^E$	26.53
$(\boldsymbol{\omega}\mathbf{m}\mathrm{p})^E\,\mathbf{q}^{\mathrm{diff}}$	32.86
$(\boldsymbol{\omega}\mathbf{m}\mathrm{p})^E\,\mathbf{q}^{\mathrm{diff}}$ + U-SVD	54.62

The run times of the classifiers are provided in Table 6.2. Part (a) of the table shows the total run time for an average cross-validation iteration including the training phase and classification of all the test feature vectors. The k-NN classifier has the shortest total run time among the seven classifiers whereas OMP has the longest. The training times of the classifiers in an average cross-validation iteration are provided in Table 6.2(b). In the training phase, the k-NN and OMP classifiers only store the training feature vectors and have no training time in practice. The RF classifier is the slowest in terms of training time. Table 6.2(c) contains the average classification time for a test feature vector. The ANN, LDC, and k-NN classifiers are the fastest, identifying the activity in no longer than than 0.1 ms. The OMP classifier has the longest run time because it executes an iterative algorithm independently for each test feature vector, but its run time is still much shorter than the segment duration, allowing a near real-time implementation.

Table 6.2: (a) Total run time (including training and classification of all test feature vectors) and (b) training time in an average L1O iteration. (c) Average classification time of a single test feature vector. The run times are shown as the average values plus/minus one standard deviation over the following transformation techniques: fixed units, $(\boldsymbol{\omega}\mathbf{m})^E$, $(\boldsymbol{\omega}\mathbf{m}\mathrm{p})^E\,\mathbf{q}^{\mathrm{diff}}$, $(\boldsymbol{\omega}\mathbf{m}\mathrm{p})^E\,\mathbf{q}^{\mathrm{diff}}$ + U-SVD, SUC-I + $(\boldsymbol{\omega}\mathbf{m}\mathrm{p})^E\,\mathbf{q}^{\mathrm{diff}}$, SUC-II + $(\boldsymbol{\omega}\mathbf{m}\mathrm{p})^E\,\mathbf{q}^{\mathrm{diff}}$, SUC-III + $(\boldsymbol{\omega}\mathbf{m}\mathrm{p})^E\,\mathbf{q}^{\mathrm{diff}}$.

	(a)	**(b)**	**(c)**
classifier	**total run time (s)**	**training time (s)**	**classification time (ms)**
SVM	10.65 ± 2.83	8.83 ± 2.65	0.39 ± 0.10
ANN	4.90 ± 1.06	4.89 ± 1.05	0.01 ± 0.00
BDM	1.31 ± 0.04	0.01 ± 0.00	1.27 ± 0.04
LDC	1.75 ± 0.11	0.26 ± 0.03	0.02 ± 0.00
k-NN	0.11 ± 0.01	–	0.10 ± 0.01
RF	26.57 ± 3.22	23.34 ± 2.98	0.83 ± 0.08
OMP	89.26 ± 7.77	–	87.25 ± 7.78

6.5 Concluding Remarks

In this chapter, we have concentrated on simultaneous position and orientation invariance of wearable motion sensor units in the context of human activity recognition. To improve the robustness of the activity recognition system, we have proposed to utilize the techniques that we have developed in Chapters 3 and 5. This scheme allows the users to place the wearable sensor units at any position and orientation on their body, provided that the sensor configuration used to acquire the dataset includes the body parts on which the units are worn. The subjects may either place the units without the need of identifying them or place only one sensor unit at any position and orientation on a body part from which data are collected.

Chapter 7

Summary and Conclusions

We have proposed techniques to achieve robustness to the placement of wearable motion sensor units where none of the approaches in the literature provide a generic framework that achieves significant robustness to their placement throughout the body. To this aim, we have developed three types of transformation and classification methods:

- orientation-invariant techniques that transform the data such that they are not affected from the orientation at which the units are placed on the body (Chapters 2 and 3),
- position-invariant techniques that achieve robustness to the positioning and the interchanging of the units on the body (Chapter 5), and
- simultaneously position- and orientation-invariant techniques that allow both of the above flexibilities (Chapter 6).

We have also proposed a novel technique for estimating the orientations of the sensor units to improve the accuracy of the orientation-invariant techniques that are based on orientation estimation.

We have employed publicly available datasets to assess the performance of these techniques for repeatability. These datasets had been recorded independently of the techniques proposed in this thesis so that there was no possibility to fine tune the types of activities, sensor types and configurations, and experimental procedures to unfairly improve the effectiveness of the proposed methods. We intentionally have not exploited the types of activities in the datasets and the specific sensor positions in developing the transformation techniques because we have aimed to keep the proposed techniques applicable to different scenarios. We have used the standard activity recognition scheme including multiple state-of-the-art classifiers and cross-validation techniques as well as datasets recorded by other researchers (when applicable) to demonstrate the robustness of our methodology.

Unlike most of the existing studies, we have compared all of the proposed methods with the reference approach where the sensor units are correctly placed as well as with the existing approaches in the literature. Hence, we could provide the reduction in the accuracy caused by the robustness to the placement of sensor units and compare different techniques that are proposed for the same purpose. We have also compared the proposed methods with the worst-case scenarios by simulating randomly rotated and/or displaced sensor units and have presented the improvement obtained by these methods.

The proposed techniques are applicable to short time segments of recorded sensor data independently, which enables them to be used in different applications, including near real-time ones, since there is no long-term dependency on the past data. This property also restricts the impact of a shift or sudden change in the positions and/or orientations of the sensor units to the time segment during which the change occurs. The proposed transformations can be applied in the pre-processing stage of existing wearable systems without much effort, making them invariant to sensor position and/or orientation. The use of these transformations does not require restrictive assumptions about the activity types and the experimental setup. Most of the transformations do not make any assumptions about sensor types as well, enabling them to be employed in various wearable sensing applications.

In general, we have observed a trade-off between the activity recognition accuracy and the flexibility that is allowed in sensor unit placement, as expected. While the orientation-invariance property causes a negligible reduction in the accuracy in most cases, some of the position-invariant methods considerably decrease the accuracy, especially for L1O cross validation where the variability between the training and test data is high. The simultaneously position- and orientation-invariant approaches achieve the lowest accuracy at the expense of allowing the sensor units to be placed on the body almost arbitrarily. The accuracy might be improved by selecting a subset of the activity types according to the application, or acquiring training data from the specific subject for personalized training.

This study is a proof-of-concept for a comparative analysis of the accuracies and run times of the proposed and existing methods as well as state-of-the-art classifiers. Therefore, we have implemented them as well as the remaining parts of the activity recognition framework on a laptop computer rather than on a mobile platform.

Given that the data transformation techniques and most of the classifiers have been implemented in MATLAB in this study, it is possible to further improve the efficiency of the algorithms by programming them in other languages such as C++, by implementing them on an FPGA platform, or by embedding the algorithms in wearable hardware. As such, our methodology can be handled by the limited resources of wearable systems such as computing processor, battery and storage capacity, and wireless transmission capability. Alternatively, transmitting the data acquired from wearable devices wirelessly to a cloud server would allow performing the activity recognition in the cloud [64, 125]. Despite the latency issues that will arise in this case, this approach would provide additional flexibility and enable the applications of wearables to further benefit from the proposed methodology and the advantages of cloud computing.

As future work, one may consider investigating additional robust features invariant to the placement of the sensor units such as differential quaternions represented in the sensor frame. Differential quaternions with respect to the Earth frame may be extracted over a wider time window rather than over only two consecutive time samples (Section 3.3), which may improve robustness against high-frequency noise.

The number of activity types may be reduced for simultaneous position and orientation invariance as this might improve the accuracy and used in more specific applications. Besides activity recognition and monitoring, the proposed approaches can be exploited in other applications of wearable sensing such as gesture recognition, posture and gait analysis, fall detection and classification, sports science, virtual reality, pedestrian dead reckoning, and automated evaluation of physical therapy exercises. For instance, the study reported in [36] assumes that the motion sensors used for gait-based personal authentication have fixed orientations. In [16], physical therapy exercises are detected and evaluated based on template signals by using time-domain sequences acquired from wearable sensors. Making fall detection and classification algorithms invariant to sensor position and orientation would be another valuable contribution. The proposed techniques can be employed in such applications to allow flexibility in the placement of motion sensor units. Energy harvesting techniques based on MEMS technology can be used in order to extend the battery lives of wireless sensor units [126].

Appendix A

Sensor Unit Orientation Estimation Using Gauss-Newton Algorithm

The OEM in [107] combines orientation estimates based on two sources of information. The first, *dynamic* estimate is obtained simply by integrating the gyroscope angular rate measurements. This estimate is accurate in the short term but drifts in the long term. The second, *static* estimate relies on the direction of the gravity vector measured by the accelerometer and the magnetic field of the Earth detected by the magnetometer in the long term. For the long-term estimation, the Gauss-Newton method [107] is used to solve a minimization problem where the cost function decreases as the acquired acceleration vector is aligned with the gravity vector and as the acquired magnetic field vector is aligned with the magnetic North of the Earth. The short- and long-term estimates are combined through weighted averaging [107].

In the orientation estimation algorithm, we relate the sensor and the Earth frames by a quaternion $\hat{\mathbf{q}}_n = (q_1,\, q_2,\, q_3,\, q_4)^T$ corresponding to the rotation matrix $\hat{\mathbf{R}}_E^{S_n} = \left(\hat{\mathbf{R}}_{S_n}^E\right)^{-1}$ for all n as follows [119]:

$$\hat{\mathbf{R}}_E^{S_n} = \begin{bmatrix} q_1^2+q_2^2-q_3^2-q_4^2 & 2\left(q_2q_3-q_1q_4\right) & 2\left(q_1q_2+q_2q_4\right) \\ 2\left(q_2q_3+q_1q_4\right) & q_1^2-q_2^2+q_3^2-q_4^2 & 2\left(q_3q_4-q_1q_2\right) \\ 2\left(q_2q_4-q_1q_3\right) & 2\left(q_1q_2+q_3q_4\right) & q_1^2-q_2^2-q_3^2+q_4^2 \end{bmatrix} \tag{A.1}$$

The short- and long-term orientation estimates are denoted by $\hat{\mathbf{q}}_{n,\,\mathrm{ST}}$ and $\hat{\mathbf{q}}_{n,\,\mathrm{LT}}$ and the overall estimate is denoted by $\hat{\mathbf{q}}_n$.

The short-term estimate of the sensor quaternion $\hat{\mathbf{q}}_{n,\,\mathrm{ST}}$ at time sample n based on the overall estimate $\hat{\mathbf{q}}_{n-1}$ at the previous time sample is given by:

$$\hat{\mathbf{q}}_{n,\,\mathrm{ST}} = \hat{\mathbf{q}}_{n-1} + \Delta t \left(\frac{1}{2}\hat{\mathbf{q}}_{n-1} \otimes \boldsymbol{\omega}'^S[n]\right) \tag{A.2}$$

where $\boldsymbol{\omega}'^S[n] = \left(0,\ \omega_x^S[n],\ \omega_y^S[n],\ \omega_z^S[n]\right)^T$ is an augmented vector consisting of zero and the angular rate vector acquired by the gyroscope at time sample n [107] and Δt is the sampling interval. Note that the equation involves feedback because $\hat{\mathbf{q}}_{n,\,\mathrm{ST}}$ is calculated based on $\hat{\mathbf{q}}_{n-1}$.

For the long-term estimation, let $\mathbf{a}^S[n]$ and $\mathbf{m}^S[n]$ be the acceleration and the magnetic field vectors, respectively, represented in the sensor frame and normalized by their magnitudes. To align $\mathbf{a}^S[n]$ with the z_E axis of the Earth frame, we represent it in the Earth frame as $\mathbf{a}^E[n] = \mathbf{q}_n \otimes \mathbf{a}^S[n] \otimes \mathbf{q}_n^*$, and solve the following minimization problem [107]:

$$\begin{aligned} \hat{\mathbf{q}}_{n,\,\text{LT-1}} &= \arg\min_{\mathbf{q}_n}\ \mathfrak{f}_1\left(\mathbf{q}_n,\ \mathbf{a}^S[n]\right) \\ &\quad \text{where} \quad \mathfrak{f}_1\left(\mathbf{q}_n,\ \mathbf{a}^S[n]\right) = \left\|(0,\ 0,\ 1)^T - \mathbf{q}_n \otimes \mathbf{a}^S[n] \otimes \mathbf{q}_n^*\right\| \end{aligned} \tag{A.3}$$

where $\otimes$ denotes the quaternion product operator.

We represent the magnetic field vector $\mathbf{m}^S[n]$ as $\mathbf{m}^E[n] = \mathbf{q}_n \otimes \mathbf{m}^S[n] \otimes \mathbf{q}_n^*$ in the Earth frame and allow it to have only a vertical component along the z_E direction and a horizontal component along the x_E direction. Hence, we align $\mathbf{m}^E[n]$ with the magnetic reference vector defined as $\mathbf{m}_0[n] \triangleq \left(\sqrt{(m_x^E[n])^2 + (m_y^E[n])^2},\ 0,\ m_z^E[n]\right)^T$ in the Earth frame by solving the following minimization problem [107]:

$$\hat{\mathbf{q}}_{n,\,\text{LT-2}} = \arg\min_{\mathbf{q}_n} \ \mathfrak{f}_2\big(\mathbf{q}_n,\ \mathbf{m}^S[n]\big)$$
$$\text{where} \quad \mathfrak{f}_2\big(\mathbf{q}_n,\ \mathbf{m}^S[n]\big) = \big\|\mathbf{m}_0[n] - \mathbf{q}_n \otimes \mathbf{m}^S[n] \otimes \mathbf{q}_n^*\big\| \tag{A.4}$$

To simultaneously align the acceleration and magnetic field vectors, we combine the minimization problems defined in Equations (A.3) and (A.4) into one and solve the following joint minimization problem:

$$\hat{\mathbf{q}}_{n,\,\text{LT}} = \arg\min_{\mathbf{q}_n} \ \mathfrak{f}\left(\mathbf{q}_n,\ \mathbf{a}^S[n],\ \mathbf{m}^S[n]\right) \tag{A.5}$$

where the combined objective function is

$$\mathfrak{f}\left(\mathbf{q}_n,\ \mathbf{a}^S[n],\ \mathbf{m}^S[n]\right) = \mathfrak{f}_1^2\big(\mathbf{q}_n,\ \mathbf{a}^S[n]\big) + \mathfrak{f}_2^2\big(\mathbf{q}_n,\ \mathbf{m}^S[n]\big) \tag{A.6}$$

We use the Gauss-Newton method to solve the problem defined in Equation (A.5) iteratively [107]. The quaternion at iteration $i+1$ can be calculated based on the estimate at the ith iteration as follows:

$$\mathbf{q}_{n,\,\text{LT}}^{(i+1)} = \mathbf{q}_{n,\,\text{LT}}^{(i)} - \left(\mathbf{J}^T\mathbf{J}\right)^{-1}\mathbf{J}^T\,\mathfrak{f}\left(\mathbf{q}_{n,\,\text{LT}}^{(i)},\ \mathbf{a}^S[n],\ \mathbf{m}^S[n]\right) \tag{A.7}$$

where $\mathbf{J}$ is the 6×4 Jacobian matrix of $\mathfrak{f}$ with respect to the elements of $\mathbf{q}_n^{(i)}$. This matrix is provided in closed form in [107].

Finally, the short- and long-term estimates are merged by using weighted averaging [107]:

$$\hat{\mathbf{q}}_n = \mathcal{K}\hat{\mathbf{q}}_{n,\,\text{ST}} + (1-\mathcal{K})\hat{\mathbf{q}}_{n,\,\text{LT}} \tag{A.8}$$

where the parameter $\mathcal{K}$ is selected as 0.98 as in [107]. The estimated quaternion $\hat{\mathbf{q}}_n$ represents the rotation matrix $\hat{\mathbf{R}}_E^{S_n}$ compactly, where we drop the hat notation ($\hat{\ }$) in the body of the text for simplicity.

Bibliography

[1] S. J. Preece, Y. J. Goulermas, L. P. Kenney, D. Howard, K. Meijer, and R. Crompton, “Activity identification using body-mounted sensors—A review of classification techniques,” *Physiological Measurement*, vol. 30, no. 4, pp. R1–R33, April 2009.

[2] Ö. Yürür, C. H. Liu, and W. Moreno, “A survey of context-aware middleware designs for human activity recognition,” *IEEE Communications Magazine*, vol. 52, no. 6, pp. 24–31, June 2014.

[3] M. Shoaib, S. Bosch, Ö. D. İncel, H. Scholten, and P. J. M. Havinga, “A survey of online activity recognition using mobile phones,” *Sensors*, vol. 15, no. 1, pp. 2059–2085, January 2015.

[4] B. Logan, J. Healey, M. Philipose, E. M. Tapia, and S. Intille, “A long-term evaluation of sensing modalities for activity recognition,” *Proceedings of the 9th International Conference on Ubiquitous Computing (UbiComp 2007), Lecture Notes in Computer Science*, J. Krumm et al. (eds.), vol. 4717, pp. 483–500, Springer: Berlin, Heidelberg, Germany, 2007.

[5] P. Turaga, R. Chellapa, V. S., Subrahmanian, and O. Udrea, “Machine recognition of human activities: A survey,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 11, pp. 1473–1488, November 2008.

[6] T. R. Bennett, N. Gans, and R. Jafari, “Data-driven synchronization for Internet-of-Things systems,” *ACM Transactions on Embedded Computing Systems*, vol. 16, no. 3, article no: 63, July 2017.

[7] C. Chen, R. Jafari, and N. Kehtarnavaz, "A real-time human action recognition system using depth and inertial sensor fusion," *IEEE Sensors Journal*, vol. 16, no. 3, pp. 773-781, February 2016.

[8] A. Mileo, D. Merico, S. Pinardi, and R. Bisiani, "A logical approach to home healthcare with intelligent sensor-network support," *The Computer Journal*, vol. 53, no. 8, pp. 1257–1276, October 2010.

[9] M. Ros, M. P. Cuéllar, M. Delgado, A. Vila, "Online recognition of human activities and adaptation to habit changes by means of learning automata and fuzzy temporal windows," *Information Sciences*, vol. 220, pp. 86–101, January 2013.

[10] R. Poppe, "A survey on vision-based human action recognition," *Image and Vision Computing*, vol. 28, no. 6, pp. 976-990, June 2010.

[11] T. B. Moeslund, A. Hilton, and V Krüger, "A survey of advances in vision-based human motion capture and analysis," *International Journal of Ambient Computing and Intelligence*, vol. 104, no. 2–3, pp. 90–126, November–December 2006.

[12] D. Karaoğlan and A. Levi, "A survey on the development of security mechanisms for body area networks," *The Computer Journal*, vol. 57, no. 10, pp. 1484–1512, October 2014.

[13] A. Yurtman and B. Barshan, "Human activity recognition using tag-based radio frequency localization," *Applied Artificial Intelligence*, vol. 30, no. 2, pp. 153–179, March 2016.

[14] A. Yurtman and B. Barshan, "Human activity recognition using tag-based localization (Etiket-tabanlı konumlama ile insan aktivitelerinin tanınması)," *Proceedings of the IEEE 20th Conference on Signal Processing, Communications, and Applications*, in Turkish, 18–20 April 2012, Fethiye, Muğla, Turkey.

[15] C. Chen, R. Jafari, and N. Kehtarnavaz, "UTD-MHAD: A multimodal dataset for human action recognition utilizing a depth camera and a wearable

inertial sensor," *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pp. 168–172, 27–30 September 2015, Québec City, QC, Canada.

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

[17] B. Barshan and A. Yurtman, "Investigating inter-subject and inter-activity variations in activity recognition using wearable motion sensors," *The Computer Journal*, vol. 59, no. 9, pp. 1345–1362, September 2016.

[18] A. Yurtman and B. Barshan, "Activity recognition invariant to sensor orientation with wearable motion sensors," *Sensors*, vol. 17, no. 8, article no: 1838, August 2017.

[19] A. Yurtman, B. Barshan, and B. Fidan, "Activity recognition invariant to wearable sensor unit orientation using differential rotational transformations represented by quaternions," *Sensors*, vol. 18, no. 8, article no: 2725, August 2018.

[20] A. Yurtman and B. Barshan, "Detection and evaluation of physical therapy exercises by dynamic time warping using wearable motion sensor units," *Information Sciences and Systems 2013—Proceedings of the 28th International Symposium on Computer and Information Sciences*, 28–29 October 2013, Paris, France, *Lecture Notes in Electrical Engineering*, vol. 264, pp. 305–314, E. Gelenbe and R. Lent (eds.), Springer International Publishing, Switzerland, 2013.

[21] A. Yurtman and B. Barshan, "Recognizing activities of daily living regardless of wearable device orientation," *Fifth International Symposium on Engineering, Artificial Intelligence, and Applications*, Book of Abstracts, 1–3 November 2017, Kyrenia (Girne), Turkish Republic of Northern Cyprus.

[22] A. Yurtman and B. Barshan, "Classifying daily activities regardless of wearable motion sensor orientation," *Proceedings of the Eleventh International*

Conference on Advances in Computer-Human Interactions (ACHI), 25–29 March 2018, Rome, Italy.

[23] A. Yurtman and B. Barshan, "Investigation of personal variations in activity recognition using miniature inertial sensors and magnetometers (Minyatür eylemsizlik duyucuları ve manyetometrelerle aktivite tanımada kişiler arası farklılıkların incelenmesi)," *Proceedings of the IEEE 20th Conference on Signal Processing, Communications, and Applications*, in Turkish, 18–20 April 2012, Fethiye, Muğla, Turkey.

[24] A. Yurtman and B. Barshan, "Detection and evaluation of physical therapy exercises from wearable motion sensor signals by dynamic time warping (Fizik tedavi egzersizlerinin giyilebilir hareket algılayıcıları işaretlerinden dinamik zaman bükmesiyle sezimi ve değerlendirilmesi)," *Proceedings of the IEEE 20th Conference on Signal Processing, Communications, and Applications*, in Turkish, 23–25 April 2014, Trabzon, Turkey.

[25] A. Yurtman, *Detection and Classification of Human Activities Using Wearable Sensors*, M.Sc. Thesis, Department of Electrical and Electronics Engineering, Bilkent University, Ankara, Turkey, September 2012.

[26] K. Kunze and P. Lukowicz, "Sensor placement variations in wearable activity recognition," *IEEE Pervasive Computing*, vol. 13, no. 4, pp. 32–41, October–December 2014.

[27] J. L. Rodrigues, N. Gonçalves, S. Costa, and F. Soares, "Stereotyped movement recognition in children with ASD," *Sensors and Actuators A: Physical*, vol. 202, pp. 162–169, November 2013.

[28] M. H. ur Rehman, C. S. Liew, Y. W. Teh, J. Shuja, and B. Daghighi, "Mining personal data using smartphones and wearable devices: A survey," *Sensors*, vol. 15, no. 2, pp. 4430–4469, February 2015.

[29] L. Chen, J. Hoey, C. D. Nugent, D. J. Cook, and Z. Yu, "Sensor-based activity recognition," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 6, pp. 790–808, November 2012.

[30] A. Bulling, U. Blanke, and B. Schiele, "A tutorial on human activity recognition using body-worn inertial sensors," *ACM Computing Surveys*, vol. 46, no. 3, article no: 33, January 2014.

[31] O. Banos, M. A. Toth, M. Damas, H. Pomares, and I. Rojas, "Dealing with the effects of sensor displacement in wearable activity recognition," *Sensors*, vol. 14, no. 6, pp. 9995–10023, June 2014.

[32] A. T. Özdemir and B. Barshan, "Detecting falls with wearable sensors using machine learning techniques," *Sensors*, vol. 14, no. 6, pp. 10691–10708, June 2014.

[33] K. Förster, P. Brem, D. Roggen, and G. Tröster, "Evolving discriminative features robust to sensor displacement for activity recognition in body area sensor networks," *Proceedings of the 5th International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, pp. 43–48, 7–10 December 2009, Melbourne, VIC, Australia.

[34] O. Tunçel, K. Altun, and B. Barshan, "Classifying human leg motions with uniaxial piezoelectric gyroscopes," *Sensors*, vol. 9, no. 11, pp. 8508–8546, October 2009.

[35] B. Ayrulu-Erdem and B. Barshan, "Leg motion classification with artificial neural networks using wavelet-based features of gyroscope signals," *Sensors*, vol. 11, no. 2, pp. 1721–1743, January 2011.

[36] T. T. Ngo, Y. Makihara, H. Nagahara, Y. Mukaigawa, and Y. Yagi, "The largest inertial sensor-based gait database and performance evaluation of gait-based personal authentication," *Pattern Recognition*, vol. 47, no 1, January 2014.

[37] K. Kunze and P. Lukowicz, "Dealing with sensor displacement in motion-based onbody activity recognition systems," *Proceedings of the 10th International Conference on Ubiquitous Computing*, pp. 20–29, 21–24 September 2008, Seoul, South Korea.

[38] M. Jiang, H. Shang, Z. Wang, H. Li, and Y. Wang, "A method to deal with installation errors of wearable accelerometers for human activity recognition," *Physiological Measurement*, vol. 32, no. 3, pp. 347–358, February 2011.

[39] T. Hur, J. Bang, D. Kim, O. Banos, and S. Lee, "Smartphone location-independent physical activity recognition based on transportation natural vibration analysis," *Sensors*, vol. 17, no. 4, article no: 931, April 2017.

[40] D. Ravì, C. Wong, B. Lo, and G.-Z. Yang, "A deep learning approach to on-node sensor data analytics for mobile or wearable devices," *IEEE Journal of Biomedical and Health Informatics*, vol. 21, no. 1, pp. 56–64, December 2016.

[41] S. Dernbach, B. Das, N. C. Krishnan, B. L. Thomas, and D. J. Cook, "Simple and complex activity recognition through smart phones," *Proceedings of the 8th International Conference on Intelligent Environments*, pp. 214–221, 26–29 June 2012, Guanajuato, Mexico.

[42] Y. Wang, J. Lin, M. Annavaram, Q. A. Jacobson, J. Hong, B. Krishnamachari, and N. Sadeh, "A framework of energy efficient mobile sensing for automatic user state recognition," *Proceedings of the 7th International Conference on Mobile Systems, Applications, and Services*, pp. 179–192, 22–25 June 2009, Kraków, Poland.

[43] A. M. Khan, M. H. Siddiqi, and S.-W. Lee, "Exploratory data analysis of acceleration signals to select light-weight and accurate features for real-time activity recognition on smartphones," *Sensors*, vol. 13, no. 10, pp. 13099–13122, September 2013.

[44] A. M. Khan, A. Tufail, A. M. Khattak, and T. H. Laine, "Activity recognition on smartphones via sensor-fusion and KDA-based SVMs," *International Journal of Distributed Sensor Networks*, vol. 10, no. 5, article no: 503291, May 2014.

[45] A. M. Khan, Y. K. Lee, and S. Y. Lee, "Accelerometer's position free human activity recognition using a hierarchical recognition model," *Proceedings of*

the 12th IEEE International Conference on e-Health Networking, Applications and Services, pp. 296–301, 1–3 July 2010, Lyon, France.

[46] J. Lester, T. Choudhury, and G. Borriello, "A practical approach to recognizing physical activities," *Proceedings of the Fourth International Conference on Pervasive Computing*, 7–10 May 2006, Dublin, Ireland, *Lecture Notes in Computer Science*, vol. 3968, no. 1, pp. 1–16, Springer, Berlin, Heidelberg, Germany, 2006.

[47] L. Sun, D. Zhang, B. Li, B. Guo, and S. Li, "Activity recognition on an accelerometer embedded mobile phone with varying positions and orientations," *Proceedings of the 7th International Conference on Ubiquitous Intelligence and Computing*, 26–29 October 2010, Xi'an, China, *Lecture Notes in Computer Science*, vol. 6406, no. 1, pp. 548–562, Springer, Berlin, Heidelberg, Germany, 2010.

[48] S. Reddy, M. Mun, J. Burke, D. Estrin, M. Hansen, and M. Srivastava, "Using mobile phones to determine transportation modes," *ACM Transactions on Sensor Networks*, vol. 6, no. 2, article no: 13, February 2010.

[49] A. Anjum and M. U. Ilyas, "Activity recognition using smartphone sensors," *Proeedings of the 10th IEEE Consumer Communications and Networking Conference*, pp. 914–919, 11–14 January 2013, Las Vegas, NV, U.S.A.

[50] P. Siirtola and J. Röning, "Ready-to-use activity recognition for smartphones," *IEEE Symposium on Computational Intelligence and Data Mining*, pp. 59–64, 16–19 April 2013, Singapore, Singapore.

[51] J. Doppler, G. Holl, A. Ferscha, M. Franz, C. Klein, M. dos S. Rocha, and A. Zeidler, "Variability in foot-worn sensor placement for activity recognition," *Proceedings of the International Symposium on Wearable Computers*, pp. 143–144, 4–7 September 2009, Linz, Austria.

[52] A. Henpraserttae, S. Thiemjarus, and S. Marukatat, "Accurate activity recognition using a mobile phone regardless of device orientation and location," *Proceedings of the International Conference on Body Sensor Networks*, pp. 41–46, 23–25 May 2011, Dallas, TX, U.S.A.

[53] S. Thiemjarus, A. Henpraserttae, and S. Marukatat, "A study on instance-based learning with reduced training prototypes for device-context-independent activity recognition on a mobile phone," *Proceedings of the IEEE International Conference on Body Sensor Networks*, pp. 1–6, 6–9 May 2013, Cambridge, MA, U.S.A.

[54] R. Chavarriaga, H. Bayati, and J. del R. Millán, "Unsupervised adaptation for acceleration-based activity recognition: robustness to sensor displacement and rotation," *Personal and Ubiquitous Computing*, vol. 17, no. 3, pp. 479–490, June 2011.

[55] K. Förster, D. Roggen, and G. Tröster, "Unsupervised classifier self-calibration through repeated context occurrences: is there robustness against sensor displacement to gain?" *Proceedings of the International Symposium on Wearable Computers*, pp. 77–84, 4–7 September 2009, Linz, Austria.

[56] K. Kunze, P. Lukowicz, H. Junker, and G. Tröster, "Where am I: recognizing on-body positions of wearable sensors," *International Symposium on Location- and Context-Awareness*, 12–13 May 2005, Oberpfaffenhofen, Germany, *Lecture Notes in Computer Science*, vol. 3479, no. 1, pp. 264–275, Springer, Berlin, Heidelberg, Germany, 2005.

[57] W. Xu, M. Zhang, A. A. Sawchuk and M. Sarrafzadeh, "Robust human activity and sensor location corecognition via sparse signal representation," *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 11, pp. 3169–3176, November 2012.

[58] H. Lu, J. Yang, Z. Liu, N. D. Lane, T. Choudhury, and A. T. Campbell, "The Jigsaw continuous sensing engine for mobile phone applications," *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, pp. 71–84, 3–5 November 2010, Zürich, Switzerland.

[59] H. Martín, A. M. Bernardos, J. Iglesias, and J. R. Casar, "Activity logging using lightweight classification techniques in mobile devices," *Personal and Ubiquitous Computing*, vol. 17, no. 4, pp. 675–695, April 2013.

[60] T. Sztyler and H. Stuckenschmidt, "On-body localization of wearable devices: an investigation of position-aware activity recognition," *Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pp. 1–9, 14–19 March 2016, Sydney, NSW, Australia.

[61] Y. Zhong and Y. Deng, "Sensor orientation invariant mobile gait biometrics," *Proceedings of the IEEE International Joint Conference on Biometrics*, pp. 1–8, 29 September–2 October 2014, Clearwater, FL, U.S.A.

[62] A. Yurtman and B. Barshan, "Inter- and intra-subject variations in activity recognition using inertial sensors and magnetometers," *The 5th International Conference on Cognitive Systems*, Collection of Posters, p. 8, 22–23 February 2012, Vienna, Austria.

[63] S. Bhattacharya, P. Nurmi, N. Hammerla, and T. Plötz, "Using unlabeled data in a sparse-coding framework for human activity recognition," *Pervasive and Mobile Computing*, vol. 15, no. 1, pp. 242–262, December 2014.

[64] M. Shoaib, S. Bosch, Ö. D. İncel, H. Scholten, and P. J. M. Havinga, "Fusion of smartphone motion sensors for physical activity recognition," *Sensors*, vol. 14, no. 6, pp. 10146–10176, June 2014.

[65] M. Janidarmian, A. R. Fekr, K. Radecka, and Z. Zilic, "A comprehensive analysis on wearable acceleration sensors in human activity recognition," *Sensors*, vol. 17, no. 3, article no: 529, March 2017.

[66] P. Siirtola and J. Röning, "Recognizing human activities user-independently on smartphones based on accelerometer data," *International Journal of Interactive Multimedia and Artificial Intelligence: Special Issue on Distributed Computing and Artificial Intelligence*, vol. 1, no. 5, pp. 38–45, June 2012.

[67] Y. Yang, "Toward physical activity diary: motion recognition using simple acceleration features with mobile phones," *Proceedings of the 1st International Workshop on Interactive Multimedia for Consumer Electronics*, pp. 1–10, 23 October 2009, Beijing, China.

[68] N. Wang, S. J. Redmond, E. Ambikairajah, B. G. Celler, and N. H. Lovell, "Can triaxial accelerometry accurately recognize inclined walking terrains?" *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 10, pp. 2506–2516, October 2010.

[69] Y. E. Ustev, O. D. Incel, and C. Ersoy, "User, device and orientation independent human activity recognition on mobile phones: challenges and a proposal," *Proceedings of the ACM Conference on Pervasive and Ubiquitous Computing*, pp. 1427–1436, 8–12 September 2013, Zürich, Switzerland.

[70] J. Morales, D. Akopian, and S. Agaian, "Human activity recognition by smartphones regardless of device orientation," *Proceedings of SPIE-IS&T Electronic Imaging: Photonics: Mobile Devices and Multimedia: Enabling Technologies, Algorithms, and Applications*, R. Creutzburg and D. Akopian (eds.), SPIE vol. 9030, pp. 90300I-1–90300I-12, 18 February 2014, San Francisco, CA, U.S.A.

[71] J. B. Yang, M. N. Nguyen, P. P. San, X. L. Li, and S. Krishnaswamy, "Deep convolutional neural networks on multichannel time series for human activity recognition," *Proceedings of the 24th International Conference on Artificial Intelligence (IJCAI'15)*, pp. 3995–4001, 25–31 July 2015, Buenos Aires, Argentina.

[72] M. A. Alsheikh, A. Selim, D. Niyato, L. Doyle, S. Lin, and H.-P. Tan, "Deep activity recognition models with triaxial accelerometers," *Proceedings of the Workshop at the Thirtieth AAAI Conference on Artificial Intelligence: Artificial Intelligence Applied to Assistive Technologies and Smart Environments*, 12–17 February 2016, Phoenix, AZ, U.S.A.

[73] D. Ravi, C. Wong, B. Lo, and G.Yang, "A deep learning approach to on-node sensor data analytics for mobile or wearable devices," *IEEE Journal of Biomedical and Health Informatics*, vol. 21, no. 1, pp. 56–64, January 2017.

[74] S. Thiemjarus, "A device-orientation independent method for activity recognition," *Proceedings of the International Conference on Body Sensor Networks*, pp. 19–23, 7–9 June 2010, Biopolis, Singapore.

[75] J. Wu and R. Jafari, "Orientation independent activity/gesture recognition using wearable motion sensors," *IEEE Internet of Things Journal*, available online: 13 July 2018, DOI:10.1109/JIOT.2018.2856119.

[76] T. K. Moon and W. C. Stirling, *Mathematical Methods and Algorithms for Signal Processing*, Chapter 7, pp. 369, Prentice Hall: New Jersey, NJ, U.S.A., 2000.

[77] L. De Lathauwer, B. De Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM Journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1253–1278, March–May 2000.

[78] G. Seçer and B. Barshan, "Improvements in deterministic error modeling and calibration of inertial sensors and magnetometers," *Sensors and Actuators A: Physical*, vol. 247, pp. 522–538, August 2016.

[79] B. Barshan and G. Seçer, "Improved deterministic measurement model for consumer-grade accelerometers," *Electronics Letters*, vol. 52, no. 7, pp. 529–531, April 2016.

[80] K. Altun and B. Barshan, "Daily and Sports Activities Dataset," UCI Machine Learning Repository, Univ. California, Irvine, School of Information and Computer Sciences, 2013 (http://archive.ics.uci.edu/ml/datasets/Daily+and+Sports+Activities).

[81] K. Altun and B. Barshan, "Daily and Sports Activities Dataset," IEEE Data Port, February 2019 (DOI: http://dx.doi.org/10.21227/at1v-6f84).

[82] W. Ugulino, D. Cardador, K. Vega, E. Velloso, R. Milidiú, and H. Fuks, "Wearable Computing: Classification of Body Postures and Movements (PUC-Rio) Data Set," *UCI Machine Learning Repository*, University of California, Irvine, School of Information and Computer Sciences, Irvine, CA, U.S.A., 2013. Available from: `https://archive.ics.uci.edu/ml/datasets/Wearable+Computing%3A+Classification+of+Body+Postures+and+Movements+(PUC-Rio)`.

[83] J. L. Reyes-Oritz, D. Anguita, A. Ghio, L. Oneto, and X. Parra, "Human Activity Recognition Using Smartphones Data Set," *UCI Machine Learning Repository*, University of California, Irvine, School of Information and Computer Sciences, Irvine, CA, U.S.A., 2012. Available from: `https://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones`.

[84] M. Zhang and A. A. Sawchuk, "USC-HAD: A daily activity dataset for ubiquitous activity recognition using wearable sensors," *Proceedings of the ACM International Conference on Ubiquitous Computing Workshop on Situation, Activity, and Goal Awareness*, pp. 1036–1043, 5–8 September 2012, Pittsburgh, PA, U.S.A.

[85] P. Casale, O. Pujol, and P. Redeva, "Human activity recognition from accelerometer data using a wearable device," *Proceedings of the Iberian Conference on Pattern Recognition and Image Analysis*, 8–10 June 2011, Gran Canaria, Spain, *Pattern Recognition and Image Analysis, Lecture Notes in Computer Science*, vol. 6669, pp. 289–296, Springer: Berlin/Heidelberg, Germany, 2011.

[86] K. Altun, B. Barshan, and O. Tunçel, "Comparative study on classifying human activities with miniature inertial and magnetic sensors," *Pattern Recognition*, vol. 43, no. 10, pp. 3605–3620, October 2010.

[87] B. Barshan and M. C. Yüksek, "Recognizing daily and sports activities in two open source machine learning environments using body-worn sensor units," *The Computer Journal*, vol. 57, no. 11, pp. 1649–1667, November 2014.

[88] Xsens Technologies B.V., Enschede, The Netherlands, *MTi, MTx, and XM-B User Manual and Technical Documentation*, 2019, `http://www.xsens.com`.

[89] W. Ugulino, D. Cardador, K. Vega, E. Velloso, R. Milidiú, and H. Fuks, "Wearable computing: Accelerometers' data classification of body postures and movements," *Proceedings of the 21st Brazilian Symposium on Artificial Intelligence: Advances in Artificial Intelligence*, 20–25 October 2012, Curitiba, Brasil, *Lecture Notes in Computer Science/Lecture Notes in Artificial Intelligence*, vol. 7589, pp. 52–61, Springer: Berlin/Heidelberg, Germany, 2010.

[90] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Oritz, "A public domain dataset for human activity recognition using smartphones," *Proceedings of the 21th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, pp. 437–442, 24–26 April 2013, Bruges, Belgium.

[91] P. Casale, O. Pujol, and P. Redeva, "Personalization and user verification in wearable systems using biometric walking patterns," *Personal and Ubiquitous Computing*, vol. 16, pp. 563–580, June 2012.

[92] K. Altun and B. Barshan, "Human activity recognition using inertial/magnetic sensor units," *Proceedings of the International Workshop on Human Behaviour Understanding*, 22 August 2010, Istanbul, Turkey, *Behavior Understanding: First International Workshop, Lecture Notes in Computer Science*, vol. 6219, pp. 38–51, Springer, Berlin, Heidelberg, Germany, 2010.

[93] M. Rulsch, J. Busse, M. Struck, and C. Weigand, "Method for daily-life movement classification of elderly people," *Biomedical Engineering*, vol. 57, no. 1, pp. 1071–1074, September 2012.

[94] M. Á. Álvarez de la Concepción, L. M. Soria Morillo, J. A. Álvarez García, and L. González-Abril, "Mobile activity recognition and fall detection system for elderly people using Ameva algorithm," *Pervasive and Mobile Computing*, vol. 34, pp. 3–13, January 2017.

[95] 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.

[96] A. Webb, *Statistical Pattern Recognition*, John Wiley & Sons, New York, NY, U.S.A., 2002.

[97] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, John Wiley & Sons, New York, NY, U.S.A., 2000.

[98] S. S. Keerthi and C.-J. Lin, "Asymptotic behaviors of support vector machines with Gaussian kernel," *Neural Computation*, vol. 15, no. 7, pp. 1667–1689, July 2003.

[99] K.-B. Duan and S. S. Keerthi, "Which is the best multiclass SVM method? An empirical study," *Proceedings of the 6th International Workshop*, 13–15 June 2005, Seaside, CA, U.S.A., *Multiple Classifier Systems, Lecture Notes in Computer Science*, vol. 3541, pp. 278–285, Springer, Berlin, Heidelberg, Germany, 2005.

[100] C. C. Chang and C. J. Lin, "LIBSVM: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, article no: 27, April 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`.

[101] C. W. Hsu, C. C. Chang, and C. J. Lin, "A practical guide to support vector classification," Technical Report, Department of Computer Science, National Taiwan University, Taipei, Taiwan, 2003.

[102] J. M. Zurada, *Introduction to Artificial Neural Networks*, vol. 8, St. Paul: West Publishing Company, Dakota Country, MN, U.S.A., 1992.

[103] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd ed., Prentice Hall, New Jersey, NJ, U.S.A., 1998.

[104] L. Wang, L. Cheng, and G. Zhao, *Machine Learning for Human Motion Analysis: Theory and Practice*, IGI Global: Hershey, PA, U.S.A., 2000.

[105] J. K. Aggarwal and M. S. Ryoo, "Human activity analysis: A review," *ACM Computing Surveys*, vol. 43, no. 3, article no: 16, April 2011.

[106] A. Henpraserttae, S. Thiemjarus, and S. Marukatat, "Accurate activity recognition using a mobile phone regardless of device orientation and location," *Proceedings of the International Conference on Body Sensor Networks*, pp. 41–46, 23–25 May 2011, Dallas, TX, U.S.A.

[107] D. Comotti, "Orientation estimation based on Gauss-Newton method and implementation of a quaternion complementary filter," Technical Report,

2011, Department of Computer Science and Engineering, University of Bergamo, Italy.

[108] Y. Tian, H. Wei, and J. Tan, "An adaptive-gain complementary filter for real-time human motion tracking with MARG sensors in free-living environments," *IEEE Transactions on Neural Systems Rehabilitation Engineering*, vol. 21, no. 2, pp. 254–264, March 2013.

[109] M. D. Shuster and S. D. Oh, "Three-axis attitude determination from vector observations," *Journal of Guidance, Control, and Dynamics*, vol. 4, no. 1, pp. 70–77, January 1981.

[110] X. Yun, E. R. Bachmann, and R. B. McGhee, "A simplified quaternion-based algorithm for orientation estimation from Earth gravity and magnetic field measurements," *IEEE Transactions on Instrumentation and Measurement*, no. 3, pp. 638–650, March 2008.

[111] S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan, "Estimation of IMU and MARG orientation using a gradient descent algorithm," *Proceedings of the IEEE International Conference on Rehabilitation Robotics*, Zürich, Switzerland, 29 June–1 July 2011.

[112] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *Journal of the Society for Industrial and Applied Mathematics*, vol. 11, no. 2, pp. 431–441, June 1963.

[113] M. Ghobadi, P. Singla, and E. T. Esfahani, "Robust attitude estimation from uncertain observations of inertial sensors using covariance inflated multiplicative extended Kalman filter," *IEEE Transactions on Instrumentation and Measurement*, vol. 67, no. 1, pp. 209–217, January 2018.

[114] H. Admed and M. Tahir, "Improving the accuracy of human body orientation estimation with wearable IMU sensors," *IEEE Transactions on Instrumentation and Measurement*, vol. 66, no. 3, pp. 535–542, March 2017.

[115] R. G. Valenti, I. Dryanovski, and J. Xiao, "A linear Kalman filter for MARG orientation estimation using the algebraic quaternion algorithm,"

IEEE Transactions on Instrumentation and Measurement, vol. 65, no. 2, pp. 467–481, February 2016.

[116] G. Cai, B. M. Chen, and T. H. Lee, "Coordinate Systems and Transformations," *Unmanned Rotorcraft Systems*, Chapter 2, pp. 23–34, Springer: London, U.K., 2011.

[117] M. W. Spong, S. Hutchinson, and M. Vidyasagar, "Rotational Transformations," *Robot Modeling and Control*, Section 2.3, pp. 44–48, John Wiley & Sons: New York, NY, U.S.A., 2006.

[118] C.-T. Chen, "Similarity Transformation," *Linear System Theory and Design*, Section 3.4, pp. 53–55, Oxford University Press: New York, NY, U.S.A., 1999.

[119] J. Diebel, *Representing Attitude: Euler Angles, Unit Quaternions, and Rotation Vectors*, Technical Report, 2006, `http://www.swarthmore.edu/NatSci/mzucker1/papers/diebel2006attitude.pdf`.

[120] I. H. Witten, E. Frank, M. A. Hall, C. J. Pal, *Data Mining: Practical Machine Learning Tools and Techniques*, Cambridge, MA, U.S.A., 4th ed., 2016.

[121] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition," *Proceedings of the 27th Asilomar Conference on Signals, Systems and Computers*, pp. 40–44, 1–3 November 1993, Pacific Grove, CA, U.S.A.

[122] Sensor Fusion and Tracking Toolbox, MATLAB R2018b, The MathWorks, Inc., Natick, MA, U.S.A.

[123] M. Pedley, M. Stanley, and Z. Baranski, "Freescale Sensor Fusion Kalman Filter," Open Source Sensor Fusion, GitHub Repository, October 2014. Available from: `https://github.com/memsindustrygroup/Open-Source-Sensor-Fusion`.

[124] J. L. Meriam and L. G. Kraige, *Engineering Mechanics: Dynamics*, John Wiley & Sons, New York, NY, U.S.A., 2012.

[125] R. Rawassizadeh, T. J. Pierson, R. Peterson, D. Kotz, "NoCloud: Exploring network disconnection through on-device data analysis," *IEEE Pervasive Computing*, no. 17, no. 1, pp. 64–74, January 2018.

[126] D. Briand, E. Yeatman, and S. Roundy, *Micro Energy Harvesting*, John Wiley & Sons, New York, NY, U.S.A., 2015.