Reliable Real-Time Recognition of Motion Related Human Activities Using MEMS Inertial Sensors

Korbinian Frank[‡], María Josefa Vera Nadales[‡*], Patrick Robertson[‡] and Michael Angermann[‡]
[‡] *German Aerospace Center (DLR), Institute of Communications and Navigation, Oberpfaffenhofen, Germany*
[*] *Universidad de Málaga, Escuela Técnica Superior de Ingeniería de Telecomunicación, Málaga, Spain*

BIOGRAPHY

Korbinian Frank holds a Dipl.-Inform. degree in computer science from the Ludwig-Maximilians University in Munich and a Maîtrise d'Informatique degree from the University of Nice-Sophia Antipolis in France. Since 2006 he has been working as a PhD candidate at the Institute for Communications and Navigation at the German Aerospace Centre (DLR) in Oberpfaffenhofen, Germany. His current focus is lying on context reasoning, learning and prediction in ubiquitous computing environments using Bayesian techniques.

María J. Vera Nadales received a degree in Telecommunications Engineering from the University of Málaga, Spain, in 2010. Her Master Thesis was carried out at the Institute for Communications and Navigation at the German Aerospace Centre (DLR) in Oberpfaffenhofen, Germany. She is currently working as services engineer for the optimisation of 2G/3G mobile communication networks.

Patrick Robertson received a Ph.D. from the University of the Federal Armed Forces, Munich, in 1995. He is currently with DLR, where his research interests are navigation, sensor based context aware systems, signal processing, and novel systems and services in various mobile and ubiquitous computing contexts.

Michael Angermann holds a Dipl.-Ing. Degree in electrical engineering from the Technical University of Munich and a PhD (Dr.-Ing.) from the University of Ulm. Among his research interests are the application of situation awareness in mobile/pervasive/ubiquitous computing, the application of information theory and Bayesian methods to context awareness and SLAM.

ABSTRACT

Knowledge about the current motion related activity of a person is information that is required or useful for a number of applications. Technical advances in the past years have reduced prices for sensors capable of providing the necessary input, in particular MEMS based inertial measurement units (IMUs). In addition to a low price, unobtrusiveness is a requirement for a activity recognition system. We achieve this by mounting one IMU to the belt of the user. In this work we present the design of our recognition system, including the features computed from the raw accelerations and turn rates as well as four different classification algorithms. These are used in Bayesian techniques trained from a semi naturalistic, labeled data set. The best classifier recognizes the activities 'Sitting', 'Standing', 'Walking', 'Running', 'Jumping', 'Falling' and 'Lying' of any person with recognition recalls and precisions between 93 and 100% except for an only 80% recall rate for 'Falling' as that suffers from its very short duration.

1 INTRODUCTION

Knowledge about the current activity of a person, in particular motion related activities, is helpful in many domains:

Indoor Navigation for instance would benefit from the knowledge about the current activity. Bayesian Location estimation systems like the one in [1] for instance can use this information to select an appropriate movement model for the person to be navigated. GPS receivers can go into idle modes when the person is not moving, or they can change their tracking characteristics.

In indoor positioning and navigation, the current activity may also be used as an information source to limit the possible locations in combination with the integration of floor plans (see for instance [2]), just like walls act like constraints that aid localisation. An example for this is Microsoft's *Greenfield* experiment [3].

For first responders, security personnel or fire fighters, knowledge about their current or recent physical activity or status is very relevant. The controlling agency can react more quickly to unforeseen events and is alerted if personnel are endangered. In domains like Ambient Assisted Living knowledge of a person's physical activity can be used as early warning systems in the case, say, that they are showing signs of reduced activity or unhealthy or unusual activity patterns. In the future, Smart Phones and other devices might even adapt their appearance and interfaces not just as a function of time and location, but also in response to whether the user is walking or if she is sitting, for example. One can also consider lifestyle applications that, for instance, automatically adapt the music played by a portable

music player depending on time, location and activity.

In all these use cases, a set of requirements becomes obvious. The recognition of activities has to function in real time, without long learning phases during usage. The system must not depend on infrastructure settings as activity recognition via image processing would with fixed mounted cameras, and finally, the system must be easy to wear, lightweight, compact and unobtrusive.

The aim of this paper is to show the design and performance of robust and reliable activity recognition in real-time with a single IMU worn on the belt for the activities 'Sitting', 'Standing', 'Walking', 'Running', 'Jumping', 'Falling', and 'Lying'. These seven activities have been chosen to serve the above described use cases. These activities are the most general ones, and furthermore representative for all motion related activities, as they include such with a repetitive pattern, static activities, but also short-time activities. We also take into account activity transitions in general – they are less relevant for applications, but helpful to support the recognition system itself.

After an overview of the related work in section 2, we will explain in section 3 how we process the raw data to extract signal-level features that represent characteristics for human movements. These features can be used in various classification algorithms, in particular Bayesian algorithms, to infer the activity. The algorithms we used in our work are presented in section 4. They are tested and evaluated with real data, which is described in sections 5 and 6, before we end with a short conclusion and outlook to further work.

2 RELATED WORK

Most of the related work on activity recognition to date is focussed on inferring human motion and posture related activities [4, 5, 6, 7, 8, 9, 10, 11, 12, 13] rather than high level activities like in [14, 15]. Recognizing motion related activities, along with precise location, is fundamental to all higher-level context aware computing applications, since so much of what defines our current situation depends on, or can be inferred from, where we are and what we are physically doing.

There are a number of motion related activities studied in the literature [5]:

- Human motion activities like *walking, walking upstairs and downstairs, standing* or *sitting*.
- Sports activities such as *jogging, cycling, rowing, callisthenics* or *martial arts moves*.
- Gestures such as *open door* or *close door* mainly for industrial environments.

Several examples of the activities and sensors used in the related work are shown in Table 1. Good results have been reached, in particular when combining several sensors from different positions on the body, which makes such recognition systems however more obtrusive and less practically usable.

In general however results can hardly be compared as the data sets differ, are recorded under different conditions (see a discussion on the relevancy of these in Intille [16]) and evaluated with different mechanisms.

3 FEATURE SELECTION

To recognisee activities with an IMU, features computed from its raw data can be used. This is because a feature is a statistical parameter or characteristic of the signal that could be significant for at least one activity or transition. We have chosen the approach of using features as a stepping stone from the signal to the estimator (see Section 4) because they are a much more compact representation, and they also lend themselves to interpretation as to what aspects of the signals are relevant to our estimation problem.

3.1 SOURCES OF INFORMATION

Most related work in biomechanics (see e.g. [20]) and accelerometer based recognition of human motion related activities (see the overview in section 2) identified the *norm* of the acceleration as the main source of information. However, an IMU can provide acceleration and angular velocity regarding different reference frames as well as angular information describing the axes of every frame. The following sections explain the reference frames and the measured sensor information we used.

3.1.1 Global Frame

The global frame is in our case defined in reference to the Earth and its center: The Earth gravity field influences the measurements of the accelerometers (as a component of the specific force). Measurements made in this reference frame are important sources of information and are considered in our work. Concretely, relevant information on human motion is strongly reflected in the vertical axis.

The angular velocity in the global frame, however, is not studied as is appeared from our studies that the information of this signal over this frame is not strongly related to human motion.

3.1.2 Approximation of the Body Frame

The IMU we used (an Xsens MTx-28A53G25) provides the measurements in the sensor frame (SF) and the necessary attitude information in order to rotate them to the global frame. However, acceleration and angular velocity relative to the human body seem to be the most relevant information (and not relative to an earth-fixed reference frame or the sensor frame as the sensor can be placed on the body in any orientation and position). The three axes of the body frame are defined to intersect at the sensor location (see Figure 1 (a)), the z axis is directed towards the head, while the other axes (x and y) form the plane orthogonal to this vertical axis. In order to obtain the rotation between the sensor frame from the vertical axis in the body frame, the 3D orientation computed internally in the sensor package can be used (see Figure 1 (b)).

Ref.	Activities	Sensors	Sensor Placement	Data set	Recognition Accuracy
[4]	Human motion	Accelerometer X-Y, GPS	Pocket	10 people (L)	85% - 90%
[6]	Human motion	Accelerometer 3D	Chest & thigh	5 people (S)	89.3%
[9]	Human motion + Activities of Daily Life (ADL)	Accelerometer 3D, compass, ambient light, force sensor	Right thigh, necklace, right & left wrist	13 people (S)	90.61%
[17]	Human motion, ADL	Accelerometer	Dominant wrist	7 people (L)	92.86% +/- 5.91%
[18]	5 activities to repair a bike	Accelerometer	Torso, left & right sleeve, left upper & lower arm & left hand	3 people (L)	82.7%
[10]	Human motion	Accelerometer	Wrist & thigh	1 person (N)	86% - 93%
[19]	3 Kung Fu movements	Accelerometer	2 at wrist	1 person (L)	96.67%
[11]	Human motion	Accelerometer 2D	Left upper leg	6 people (L)	42% - 96%
[12]	Human motion	Accelerometer	2 at hip	1 person (L)	83% - 90%
[13]	Human motion	Accelerometer	2 at thigh	8 people (L)	92.85% - 95.91%

Table 1. Examples of work on activity recognition. ADLs are usual high level activities such as bathing, dressing or opening a drawer. The data set column indicates the conditions of the data collected: under laboratory (L), semi-naturalistic (S) or naturalistic conditions (N).

(a) The ideal body frame will have its origin in the center of mass of the human body in *standing* position or any other location that does not vary over time. However, the sensor location will define the origin of the body frame and location depends on where the user puts the sensor.

(b) Final approximation to the body frame: The heading of the human body cannot be estimated unless magnetometers are used. Relevant information, however, is only included in the norm, and signal components in the vertical axis and the horizontal plane of the body frame.

Figure 1. Defining the body frame at the sensor location.

3.1.3 Information Signals

Taking into account global (GF) and body (BF) frames, the following signals are defined in this work:

- $|a|$, the norm of the acceleration a and $|\omega|$, the norm of the angular velocity ω defined as

$$|a| = \sqrt{a_x^2 + a_y^2 + a_z^2}$$
$$|\omega| = \sqrt{\omega_x^2 + \omega_y^2 + \omega_z^2}, \quad (1)$$

  where a_i and ω_i, $i \in \{\text{x,y,z}\}$ is the acceleration and angular velocity respectively at the i'th-axis.

- $|a_{horiz_{BF}}|$, horizontal acceleration in the body frame and $|\omega_{horiz_{BF}}|$, angular velocity in the horizontal plane

of the body frame,

$$|a_{horiz_{BF}}| = \sqrt{a_{x_{BF}}^2 + a_{y_{BF}}^2}$$
$$|\omega_{horiz_{BF}}| = \sqrt{\omega_{x_{BF}}^2 + \omega_{y_{BF}}^2}, \quad (2)$$

where $a_{i_{BF}}$ and $\omega_{i_{BF}}$, $i \in \{\text{x,y}\}$ is the measured acceleration and angular velocity respectively at the *i*'th-axis of the body frame.

- $a_{vert_{BF}}$, vertical acceleration in the body frame and $\omega_{vert_{BF}}$, angular velocity in the vertical axis in the body frame,

$$a_{vert_{BF}} = a_{z_{BF}}$$
$$\omega_{vert_{BF}} = \omega_{z_{BF}}, \quad (3)$$

where $a_{z_{BF}}$ and $\omega_{z_{BF}}$ are the acceleration and angular velocity at the z-axis of the body frame.

- $a_{vert_{GF}}$, vertical acceleration in the global frame,

$$a_{vert_{GF}} = a_{z_{GF}}, \quad (4)$$

where $a_{z_{GF}}$ is the acceleration measured along the z-axis of the global frame.

3.2 FEATURE REDUCTION AND SELECTION

Our objective was to select features which represent the main physical phenomena and signals for every activity. Every feature should add new information to the system. In order to select the most relevant ones, features have been computed and plotted in combination with other relevant features in a way that clusters became observable. Also the observation of signals over time (an example is shown in Figure 2) helped to recognize interdependencies between features and activities.

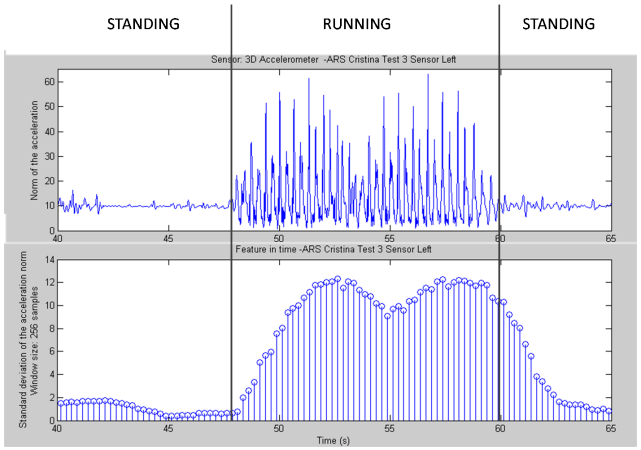


Figure 2. Example of the evolution of some features over time for the sequence *standing, running* and *standing*. $|a|$ is plotted for this sequence. Below, standard deviation of $|a|$ over a running window of 256 samples, $\sigma_{|a|}$.

After extensive human evaluation of the signals and features and their relevance to activities, we chose 19 features (shown in Table 2) which we believe to be significant and which are used for the classification algorithms.

Feature No.	Definition	Window size
1	$MAX_{\lvert a_{horiz_{BF}}\rvert}$	128
2	$\overline{\lvert a_{horiz_{BF}}\rvert}$	128
3	$\sigma_{\lvert a_{horiz_{BF}}\rvert}$	128
4	$MAX_{a_{vert_{BF}}}$	128
5	$\overline{a_{vert_{BF}}}$	128
6	$\sigma_{a_{vert_{BF}}}$	128
7	$RMS_{a_{vert_{BF}}}$	128
8	$IQR_{\lvert \omega_{horiz_{BF}}\rvert}$	128
9	$\overline{a_{vert_{GF}}}$	32
10	$\overline{\lvert a\rvert}$	32
11	$\overline{\lvert a\rvert}$	512
12	$\sigma_{\lvert a\rvert}$	256
13	$IQR_{\lvert a\rvert}$	128
14	$MFC_{\lvert a\rvert}$	128
15	$\hat{E}(\lvert a\rvert_{LPF\ <2.85\ Hz})$	128
16	$\hat{E}(\lvert a\rvert_{BPF\ 1.6-4.5\ Hz})$	64
17	$\hat{E}(\lvert a\rvert_{BPF\ 1.6-4.5\ Hz})$	512
18	$\rho_{a_{vert_{BF}},\lvert a\rvert}$	128
19	$att_{\lvert a_{horiz_{BF}}\rvert,a_{vert_{BF}}}$	64

Table 2. The set of features used for activity recognition. Most of the features are extracted from the body frame and $|a|$.

Features 10 – 17 are computed from the norm of the acceleration, the signal shown in (1). Mean values over different window lengths are relevant for short-term and longer, repetitive activities. The standard deviation $\sigma_{|a|}$ helps distinguishing between static and dynamic activities, while the interquartile range $IQR_{|a|}$ is relevant for the distinction between *jumping* and *falling*. The interquartile range is the difference between the 25th and the 75th percentile (where the 50th percentile is the median). The main frequency component $MFC_{|a|}$ computed by a Fast Fourier Transform, can identify walking and running and is in particular used to distinguish *falling* and *jumping* from *running*. Features 15 – 17 represent the energy of the norm of the acceleration in some particular frequency bands. While the low pass filter in feature 15 is helpful in distinguishing between *walking* and *jumping* or *running*; the band pass filters in features 16 and 17 can help us distinguish between *running* and *jumping*.

Features 1 – 3 and 8 use the horizontal acceleration in the body frame, see (2). The maximum horizontal acceleration $MAX_{|a_{horiz_{BF}}|}$ helps distinguishing between static and dynamic activities and has particularly high values for *falling*. The mean value distinguishes the static activities reliably and the standard deviation helps to distinguish the activities *jumping*, *falling* and *running*. The horizontal angular velocity in the body range is used in the interquartile ranges $IQR_{\omega_{horiz_{BF}}}$. High values are only reached for *falling*.

The attitude of the sensor, feature 19, takes into ac-

count the signals shown in (2) and (3). The attitude

$$att_{|a_{horiz_{BF}}|,a_{vert_{BF}}} = (\Delta\overline{|a_{horiz_{BF}}|})^2 + (\Delta\overline{a_{vert_{BF}}})^2$$

gives us information about the attitude difference between the current activity and the known sensor attitude during *standing*.

The vertical acceleration in the body frame, the signal defined in (3), is also used for features 4 – 7 and 18. The maximum value $MAX_{a_{vert_{BF}}}$ helps to distinguish between *jumping, falling* and *walking*. The mean value distinguishes *standing, sitting* and *lying*, while the standard deviation helps discriminating between all dynamic activities. The root mean square $RMS_{a_{vert_{BF}}}$ is a good discriminator for the static activities.

The correlation coefficient $\rho_{a_{vert_{BF}},|a|}$ (i.e. between the accelerations defined in (1) and (3)) is used as feature 18. *Walking*, *Running* and *Jumping* have very high values here, while other activities do not lead to consistent patterns.

Features 9 finally uses the vertical acceleration in the global frame, signal (4). Its mean value is used to detect the free fall phases during *jumping* and *falling*.

3.3 FEATURE QUANTIZATION

The feature quantization (or value discretization)) process tries to identify meaningful value ranges of these features. To identify these value ranges, histograms and plots in 2D of the features have been used, together with an inspection of the pertinent activity discrimination.

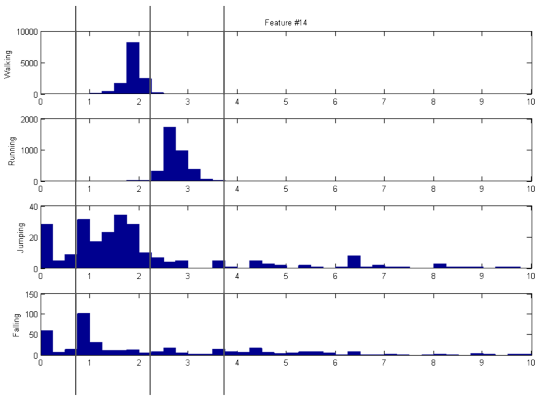
Figure 3 shows an example for a quantization. The quantized feature is the main frequency component of $|a|$. The activities for which this feature is meaningful and appropriate are *walking, running, jumping* and *falling*. The histograms and the two dimensional plot of the feature suggest four intervals of these states, the numerical values of which are determined with the help of the graphical representation.

4 CLASSIFICATION ALGORITHMS

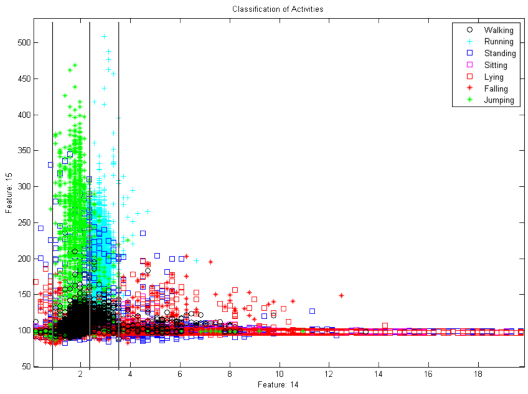
The activity recognition system has to decide which of the seven physical activities have effectively caused the measured values of the 19 features. This is a general classification problem that can be dealt with by a large range of algorithms, such as logics, *k-nearest Neighbor* approaches, *Support Vector Machines* (SVMs), *Artificial Neural networks* (ANNs) [21], *Decision Trees* or *Bayesian Techniques* [22].

Our work applies and compares different Bayesian estimation techniques as they suffer least from overfitting, high storage and processing requirements, intolerance to noise or outliers. The following characteristics make them the most appropriate approach:

1. Discrete Bayesian networks require little storage space as only the conditional probability tables (CPTs) of every node have to be stored.
2. Bayesian networks have proven successful for many applications.
3. They may need a relatively small dataset [21].



(a)



(b)

Figure 3. Feature quantization example. The feature is the main frequency component of $|a|$. It mainly is relevant for *walking, running, jumping* and *falling*. Four states can be identified with the help of the histograms (a) and the plots of a pair of features (b).

4. Their visual representation is easy to interpret and hence the link to physical world is not lost.
5. They are suitable for real-time applications as the speed of classification is high, under the constraint that there is evidence in all nodes.
6. They are very tolerant to noise as they use the probability distribution of the data.
7. Under some conditions, and these are met here, inference is a very simple computational task.

Bayesian techniques span again a number of concrete techniques. There are static and dynamic Bayesian Networks (BNs), with a number of inference algorithms for each kind. The structure and the parameters (probabilities) of a BN can be designed manually or learnt from an existing dataset with dedicated algorithms. A structural simplification of BNs is the *Naïve Bayes* approach, assuming that all observed features are independent of each other [23].

In our work, we compare a Naive Bayes approach with parameters learnt from a data set, with a true BN whose parameters and structure have been learnt from the same data

set. Based on these basic BNs, we also investigate the utility of *dynamic* BNs in comparison to static ones by employing a *Hidden Markov Model* that underlies our discrete activity transitions and evaluating it with a *grid based* filter. Learning was performed with the K2 algorithm of Cooper and Herskovits [24] using a Log score function to rate possible BN structures proposed by a Greedy Hill Climber.

The following sections will briefly outline the theory behind BNs and the different classification methods in more detail.

4.1 BAYESIAN NETWORKS

A *Bayesian Network (BN)* [25] is a probabilistic model consisting of a Triplet (V, E, p), with a set of *Random Variables (RVs)* $V = \{X_1, X_2, \ldots, X_n\}$, a set of dependencies $E = \{(X_i, X_j)|i \neq j, i, j \in V\}$ between these RVs and a *joint probability distribution (JPD)* $p(V) = P(X_1, X_2, \ldots, X_n)$. P is the product of the *Conditional Probability Distribution (CPD)* of every RV $p(X_i)\forall X_i \in V$. A BN must not contain directed cycles. This model subsumes a great variety of other stochastic methods, such as Hidden Markov Models or stochastic dynamic systems [26]. It allows for inference of knowledge being able to deal with missing or uncertain data (as for erroneous sensors or uncertain data links) and can be built or modified either by machine learning algorithms or by human expertise.

Random variables represent sets of events. Thereby they can be continuous or discrete, which has consequences on CPDs. In the case of a continuous value range $\mathbb{R}$, the CPD is a function $CPD(X_i) : \mathbb{R} \rightarrow [0, 1]$, in the case of a discrete value range, $\mathbb{R}$ consists of a finite number of states, that are assigned a probability depending on the state of the nodes which X_i depends on.

A particular interpretation of BNs are *Causal Networks*, where dependencies are interpreted as causal influence. This model makes understanding of such a network very intuitive, in particular with a graphical representation of the BN. A BN can be drawn as a *directed acyclic graph (DAG)* like the one in Fig. 4. These graphs take advantage of the fact that with its explicit dependencies, a BN exploits the conditional independence to represent a JPD more compactly. Every RV represents a node or vertex in the graph, every dependency (X_i, X_j) a directed edge from node X_i to node X_j. This representation imposes the understanding of the set $pa(X_j) = \{X_i|\forall i \in V \wedge (X_i, X_j) \in E\}$ as the *parents of* X_j. The definition of *children of* X_j $ch(X_j)$ follows similarly. Dependencies and therefore the set of parents of all nodes help to represent the JPD more compactly:

$$p(V) = p(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{n} p(X_i|pa(X_i)). \quad (5)$$

With the structure (RVs and their dependencies encoded in the network structure) and the CPDs, these networks contain the already known information about a specific domain represented by the BN. They are a knowledge representation and maintenance format. To incorporate current observations about the domain to allow inference, these can be introduced as *evidence* into the corresponding RV. The observation that RV $X_j = x_{j,1}$ sets $p(X_j = x_{j,1}) = 1$ and $p(X_j = x_{j,z}) = 0\forall z \neq 1$. In the case of discrete RVs, this can be interpreted as "switching" the probability tables of children nodes to the observed columns.

An important concept for BNs is *d-separation* with the "d" standing for dependence. It helps to reduce the network to only relevant portions for given *observations* and a specific *target RV* whose state is queried. If two variables are d-separated relative to a set of variables Z, then they are independent conditional on Z in all probability distributions of its BN. Roughly speaking, two variables X and Y are independent conditional on Z if knowledge about X gives you no extra information about Y once you have knowledge of Z [27].

More precisely: a *path* is a sequence of consecutive edges including one or more nodes. A path is called blocked or *d-separated* if a node on the path blocks the dependency. This is the case if the path p and the set of observed nodes Z are in a constellation in which

- *"p contains a chain* $i \rightarrow m \rightarrow j$ *or a fork* $i \leftarrow m \rightarrow j$ *such that the middle node* m *is in* Z*, or"*
- *"p contains an inverted fork (or collider)* $i \rightarrow m \leftarrow j$ *such that the middle node* m *is not in* Z *and such that no descendant of* m *is in* Z*."*

The d-separation criterion can be summarised by: *"a node is conditionally independent of its non-descendants, given its parents"* or *"a node is conditionally independent of all other nodes in the network, given its parents, children, and children's parents – that is, given its* Markov blanket*"* [28]. This means that the Markov blanket of a node is the only knowledge needed to predict the behavior of that node [25]. The values of the parents and children of a node evidently give information about that node. However, its children's parents also have to be included, because they can be used to "explain away´´ the node in question. For the node `Activity` the Markov Blanket is shown shaded for the exemplary BN shown in Fig. 4.

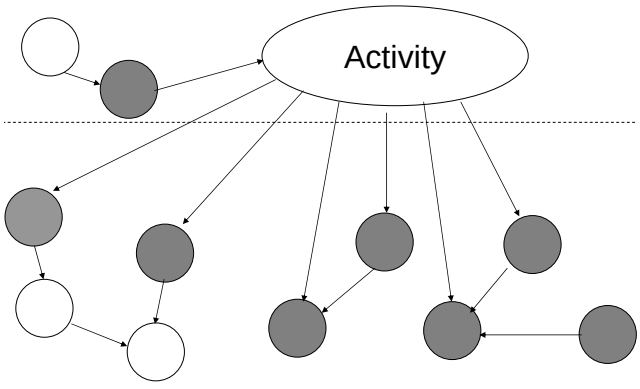


Figure 4. The Markov Blanket of the node $Activity$ in a simplified example BN is shown in gray. It contains the node's parents, children and the parents of the children without the node itself.

4.2 INFERENCE IN STATIC BAYESIAN NETWORKS

There are many different approaches for efficient inference in BNs, a good overview is given bei Guo in [29]. One of the most famous algorithms for exact inference that can be used for real-time inference is Lauritzen and Spiegelhalter's clique-tree propagation algorithm called *Probability Propagation in Trees of Clusters (PPTC)*. It takes advantage of proven fast inference algorithms in tree-like structured BNs [30], but transformation in this structure still takes exponential time in the number of nodes of the BN. The general case of inference in BN is proven to be NP hard.

The problem of exact inference can be simplified if the required probability $p(X|e)$ needs to be computed of a single RV X_{target} whose Markov blanket carries evidence in all its nodes. As this is the fortunate case in the problem presented in this paper, all values of features are known at the same time, because we are calculating the features from the IMU signals. The classification problem is now simplified to infer

$$arg_i max[p(X_{target,i}|e_{MB})], \tag{6}$$

where e_{MB} is the evidence of all RVs in the Markov blanket. Using the Bayesian theorem,

$$p(X_{target}|e_{MB}) = \frac{p(X_{target}, e_{MB})}{p(e_{MB})},, \tag{7}$$

where $p(e_{MB})$ is constant for all the states of the target node, so the problem is simplified to compute

$$\begin{aligned} &arg_i max[p(X_{target,i}, e_{MB})] = \\ &= arg_i max(p(act^i, f_1, f_2...f_M)) \,, \end{aligned} \tag{8}$$

where act^i is the RV for `Activity` (*jumping, falling, walking,...*), and the features f_i $(1 \leq i \leq 19)$ in our case represent evidence e_{MB}.

$p(act^i, f_1, f_2...f_M,)$ can be calculated multiplying the correspondent value of the CPTs as denoted in Equation 5.

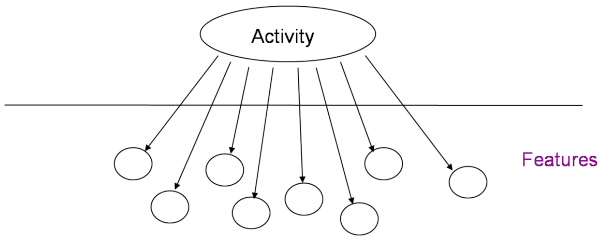


Figure 5. Naïve Bayes approach for activity recognition. The activity the user is performing is the cause of the observation of the features.

Classification with Naïve Bayes BNs, like the one shown in Figure 5, can make use of the independence assumption when calculating $arg_i max(p(act^i|f_1, f_2...f_M))$. As $p(f_1, f_2...f_M)$ is constant for all the activities and if we assume that all features are independent conditioned on `Activity`,

$$\begin{aligned} &arg_i max(p(act^i|f_1, f_2...f_M)) = \\ &= arg_i max(p(f_1, f_2...f_M|act^i)p(act^i)) \\ &= arg_i max(\prod_{j=1}^{M} p(f_j|act^i) \cdot p(act^i)). \end{aligned} \tag{9}$$

Hence classification has to infer the activity that maximizes $\prod_{j=1}^{M} p(f_j|act^i) \cdot p(act^i)$, where $p(f_j|act^i)$ is learnt from the data set and $p(act^i)$ is the prior probability, in our case set manually to values given in Table 3.

As all features carry evidence, $p(act_i, f_1, f_2...f_M)$ can be calculated immediately and the most likely activity which maximizes this probability can be identified directly.

4.3 DYNAMIC BAYESIAN NETWORKS

In most cases, the last activity a person has performed influences their current activity. For instance, if somebody is currently *lying*, the most probable activity he or she will be performing immediately afterwards is *getting up* or still *lying*, but usually not *falling* and certainly not *running*. This knowledge can provide valuable input for activity recognition, since it constrains the estimated sequence of activities to one which is likely. We can say that estimating the activity at discrete time instance t is aided by the inference of previous activities.

Figure 6 shows a Hidden Markov Model (HMM) that models this process. A Hidden Markov Model of first order is the simplest case of a Dynamic Bayesian Network, which has one discrete hidden node and one observed node per time slot. Feature values are observed (and therefore called O) during the current user's activity. The only influence from a time slot t to a time slot $t+1$ is the current activity the user is performing. Periodic evidence by features disclose the probabilities of every activity through Bayesian network inference, but these probabilities are also modified depending on the probabilities of the last activities performed. One can think of each time slice of Figure 6 to comprise a BN such as that shown in Figure 4 or Figure 5, with an additional arrow from the previous activity pointing to the current activity (O_t contains all observed features at that time instance t).

This first-order HMM for the activity recognition can be characterized by

$$\lambda \sim (A, B, \pi), \tag{10}$$

where:

- $A = \{a_{ij}|a_{ij} = p(act^j_{t+1}|act^i_t), 1 \leq i, j \leq N\}$ is the state transition probability distribution or *transition model*. A will be represented as a matrix and is given in Table 4.

- N is the number of states of the hidden variable which correspond to the different activities. The individual states at time t are $act_t = act^1_t, act^2_t, ..., act^N_t$. Activities are considered as hidden because they cannot be observed directly.

Sitting	Standing	Walking	Running	Jumping	Falling	Lying	Up & Down
0.195	0.2435	0.409	0.001	0.001	0.0005	0.14	0.01

Table 3. Prior probabilities of the node *Activity* assumed in this work.

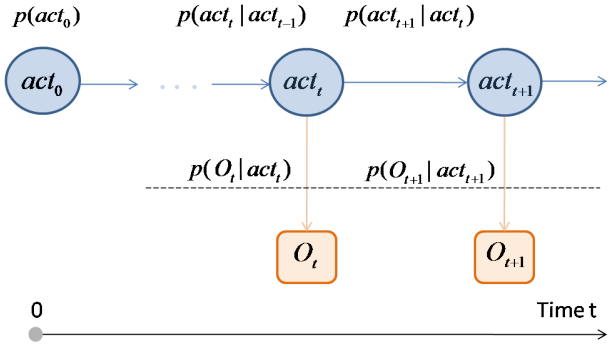


Figure 6. Hidden Markov Model of activity transitions in which $p(act_t|act_{t-1})$ were manually configured by expert knowledge and $p(O_t|act_t^i, \lambda)$ are given by the underlying Bayesian network.

- $B = \{b_j|b_j = p(O_t|act_t^j), 1 \leq j \leq N\}$ is the observation symbol probability distribution in state j, defined by the measurement model in the underlying Bayesian network.

- O are the *observation symbols* which represent the observable physical or calculated output [31]. An observation symbol for a single point in time is in our case given by a vector with values for all features $f_1, f_2, ..., f_M$ computed from the raw sensor data. The vector of features is common for all the hidden states and can be denoted $O_t = (f_{1,t}, f_{2,t}, ..., f_{M,t})$, where $f_{i,t}, 1 \leq i \leq M$ is the value of the feature i at time t.

- $\pi = \{\pi_i\}$ is the initial state distribution, where $\pi_i = p(act_0^i)$ is the prior probability, $1 \leq i \leq N$ (see Table 3; alternatively we can initialize inference with a known, defined activity such as standing).

4.4 INFERENCE IN DYNAMIC BAYESIAN NETWORKS WITH A GRID BASED FILTER

Inference in a HMM such as that in Figure 6 is the estimation of the most probable hidden state at time t given the past and current observations $O_{1:t} = O_1, O_2, ..., O_t$ as well as the model λ, $arg_i max(p(act_t^i|O_1, O_2, ..., O_t, \lambda))$.

As in our system the hidden state space has a finite number of states (i.e. activities), grid based methods can be applied providing an optimal estimation of the posterior probability density function $p(act_t|O_{1:t}, \lambda)$. Like the general optimal recursive Bayesian filter, the grid based filter consists of the prediction and update steps. The corresponding equations adapted to the HMM for activity recognition are the following:

- Prediction:

$$p(act_t|O_{1:t-1}, \lambda) = \sum_{i=1}^{N} w_{t|t-1}^i \delta(act_t - act_t^i), \quad (11)$$

- Update:

$$p(act_t|O_{1:t}, \lambda) = \sum_{i=1}^{N} w_{t|t}^i \delta(act_t - act_t^i), \quad (12)$$

where

$$\begin{aligned} w_{t|t-1}^i &\triangleq \sum_{j=1}^{N} w_{t-1|t-1}^j p(act_t^i|act_{t-1}^j) \\ w_{t|t}^i &\triangleq \frac{w_{t|t-1}^i p(O_t|act_t^i, \lambda)}{\sum_{j=1}^{N} w_{t|t-1}^j p(O_t|act_t^j, \lambda)}. \end{aligned} \quad (13)$$

Once the posterior probability is estimated, the most probable activity is given by the state with the maximum probability.

5 ACQUISITION OF TEST DATA

A total of 16 people, 6 females and 10 males aged between 23 and 50 years, of different height, weight and constitution participated in the acquisition of the test data set. They were all asked to follow a schedule of which activities to perform and in which order, to allow us to cover all activities. Test candidates were asked to execute them in their personal style without a strict choreography. They even were encouraged to perform the same activities differently and to sometimes perform these activities in such as way that a human observer could just about identify them accurately.

Data were recorded in indoor and outdoor environment under semi-naturalistic conditions. The human observer was carrying a laptop computer to which the sensor was mounted. This person was responsible for the labeling with a dedicated graphical application. The sensor was placed on the belt of the test candidate either on the right or the left part of the body. The data set comprises all different sensor positions. In order to check orientation performance of the sensor, test candidates performed their activities also with different headings.

The final data set contains over 4 hours and 30 minutes of activity data. It is online and freely accessible under `http://www.kn-s.dlr.de/activity/`. Table 5 shows the exact amount of recorded data per activity.

6 EVALUATION

This section first compares (in subsection 6.1) the four different classifiers described in Section 4 using recorded IMU raw data of two subjects. These recordings had not been included in the training data used to learn the BN.

act_t act_{t+1}	Sitting	Standing	Walking	Running	Jumping	Falling	Lying	Up & Down
Sitting	0.990902	0	0	0	0.000045	0.000045	0	0.009008
Standing	0	0.566024	0.424518	0	0.004717	0.000024	0	0.004717
Walking	0.033217	0.332171	0.498256	0.11626	0.003322	0.000166	0	0.016609
Running	0	0	0.39984	0.439824	0.079968	0.0004	0	0.079968
Jumping	0	0.36842	0.157894	0.263157	0.157894	0.000003	0	0.052631
Falling	0.4	0	0	0	0	0.1	0.5	0
Lying	0	0	0	0	0	0	0.818182	0.181818
Up & Down	0.666445	0.302929	0	0	0	0.000333	0	0.030293

Table 4. State transition probability matrix A. Each cell defines the transition probability $p(act_{t+1}|act_t)$.

Activity	Duration (minutes)
Standing	107
Sitting	55
Lying	25
Walking	70
Running	15
Jumping	7
Falling	2

Table 5. Constitution of the data set per activities.

To evaluate the performance of the classifiers, we use *precision* and *recall* measures, see [32].Subsequently, in subsection 6.2 we evaluate the implemented HW and SW system with our requirement that activities have to be recognized in real time.

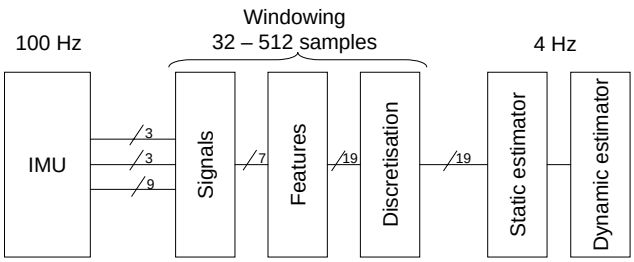


Figure 7. Information flow of the raw data to the finally recognised activity

The complete system implementing the activity recognition applies the information flow explained in Figure 7. The acceleration and turn rate data in 3D and the rotation matrix are the input to the recognition algorithm that first computes the basic signals. Using those signals, the features are computed. These (and those signals which are used directly as features without any further computation) are input to the static classifiers estimating the current activity. In the case of dynamic inference, this estimate is then used in the dynamic classifier, determining the most probable current activity.

The structure of the learnt Bayesian network used in the classifiers can be seen in Figure 8. It was trained with the complete data set that was collected as described in Section 5 with evidence samples and ground-truth labeling provided at 4 Hz.

Classifying at 4 Hz proved to be sensible as a trade off between accuracy and resource consumption. As the minimum duration of one activity is around one second, four classifications and therefore four feature computations per second are sufficient to not miss significant phases of any activity. Moreover, additional investigations have shown that classification with higher frequencies does not change precision or recall significantly.

6.1 STATIC VERSUS DYNAMIC INFERENCE

This section will compare the four approaches. Therefore we use the labeled data of two colleagues, Emil and Sinja,

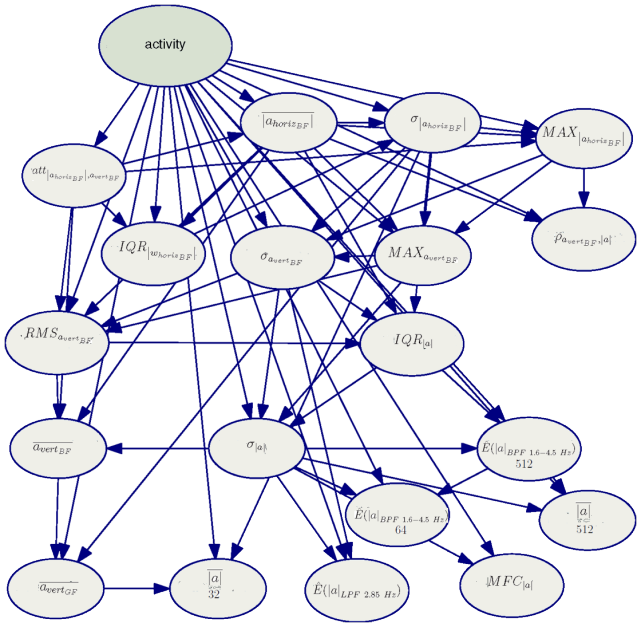
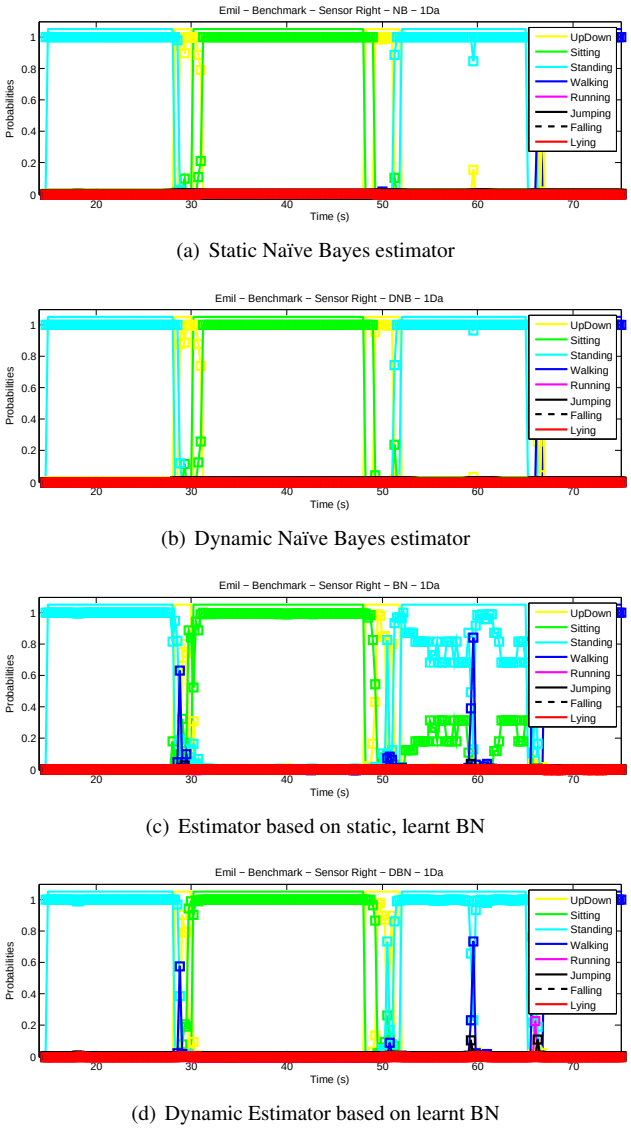


Figure 8. Structure of the learnt Bayesian network. All features are inside the Markov Blanket of the node *activity*.



(a) Static Naïve Bayes estimator

(b) Dynamic Naïve Bayes estimator

(c) Estimator based on static, learnt BN

(d) Dynamic Estimator based on learnt BN

Figure 9. Inference results for an example of the sequence *standing, sitting* and *standing*. The thin, colored line at the top of these figures depicts the ground truth, colors identify the current activity. Below the ground truth the estimated probabilities of every activity are plotted (squares).

and compare the labels with the output of the four approaches for each evaluation, i.e. four times per second.

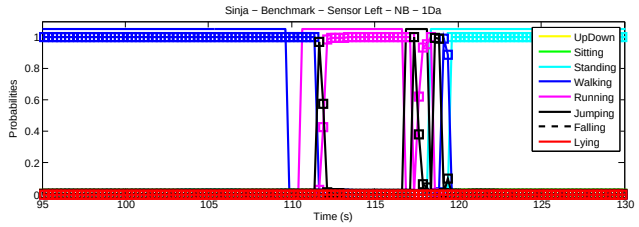
The evaluation of the activities *standing, sitting* and *standing* of Emil are shown in Figure 9. As the distinction between *standing* and *sitting* is based only on the attitude of the sensor – which depends again on the particular way the subject is sitting – the dynamic estimation improves the result significantly in this case.

The evaluation of an example of the sequence *walking, running, jumping* and *standing* is shown in Figure 10. On the one hand, the distinction of *running* and *jumping* is improved by the approach based on the unrestricted Bayesian network. On the other hand, *walking* and *running* are not confused by Naïve Bayes, but the unrestricted Bayesian network approach provides good results as well as soon as the dynamic information is included.

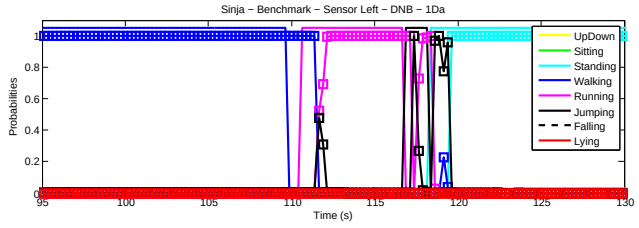
An exemplary sequence *walking, falling* and *lying* is shown in Figure 11. All four approaches work well in general. It is important to point out, however, that the duration of falling (at 166 seconds) is constantly over-estimated, especially by the Naïve Bayes estimators, where falling is estimated to extend to the 168s time frame.

Figure 12 shows promising classification results in terms of the recall and precision criteria. It illustrates clearly the advantages of the approaches with the full, learnt BN as opposed to just the Naïve Bayes approaches. Although the system is able to recognize every activity at some point of its duration, most activities are misclassified at their beginning, which affects the results in terms of precision and recall. Particularly visible are these effect for short-time activities. It is caused by the sliding window containing data samples that resulted during the previous activity. In activities like *falling*, which last for about 3 to 5 evidence samples (roughly one second), our effective recognition delay of about 2 samples (see below) decreases the precision and recall of the system significantly. In all figures 9, 10, and 11 we can identify this recognition delay. Another factor possibly degrading the results originates from the manual labeling of the test data, which includes human error, particularly in terms of accurately labeling the transitions between activities.

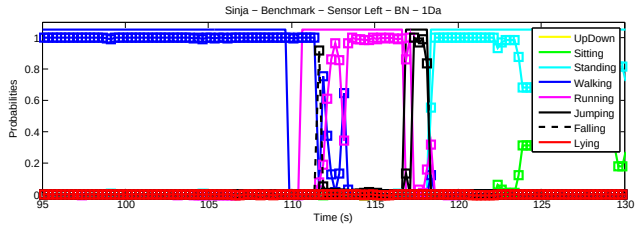
To quantify the recognition delay of the system, remember that most of the features of the final set are defined for a window length of 128 samples (or 1.28 seconds). It is sensible to postulate that at least the 50% of the window should be associated with the current activity in order to achieve an accurate inference result. 50% of the samples



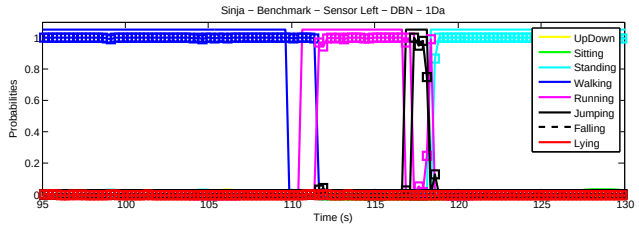
(a) Static Naïve Bayes estimator



(b) Dynamic Naïve Bayes estimator

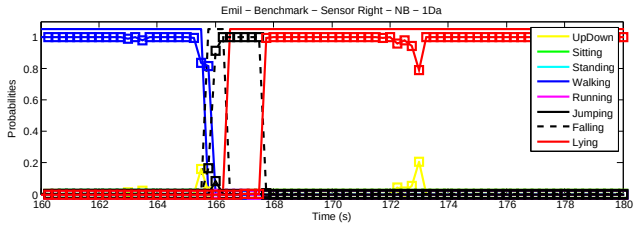


(c) Estimator based on static, learnt BN

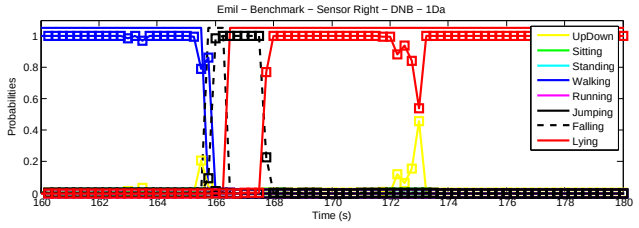


(d) Dynamic Estimator based on learnt BN

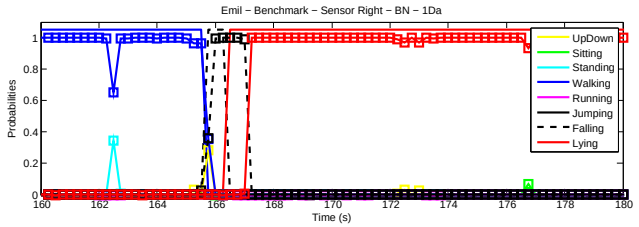
Figure 10. Inference results of the sequence *walking, running, jumping* and *standing*. The thin, colored line at the top of these figures depicts the ground truth, colors identify the current activity. Below the ground truth the estimated probabilities of every activity are plotted (squares).



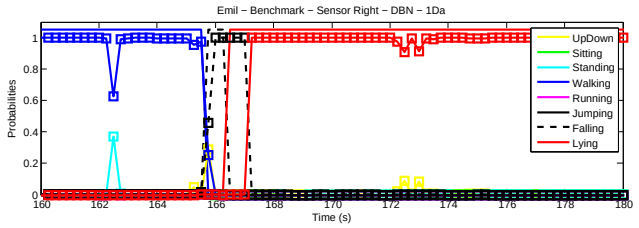
(a) Static Naïve Bayes estimator



(b) Dynamic Naïve Bayes estimator



(c) Estimator based on static, learnt BN



(d) Dynamic Estimator based on learnt BN

Figure 11. Detail of the inference results of *walking, falling* and *lying*. The thin, colored line at the top of these figures depicts the ground truth, colors identify the current activity. Below the ground truth the estimated probabilities of every activity are plotted (squares).

of a window of 128 samples implies a recognition delay of at least 64 samples or 0.64 seconds (at a sample frequency of 100 Hz). If evidence is computed every 0.25 seconds, the recognition delay can be approximated to be two evidence samples. Taking this into account, recall and precision for every activity are shown in Table 6 for the static and the dynamic approach with the learnt BN.

Table 6 shows that the dynamic approach leads to better recall rates. The dynamic approach also improves precision (specially for *falling*) in most of the activities except for *jumping* and *sitting*. Taking into account the recognition delay, all activities achieve recall rates higher than 93%. The system's accuracy for transitions (up and down) is not evaluated here, as they were not a target for our approach, but were only used as additional states to improve the dynamic inference.

6.2 COMPUTATION TIME

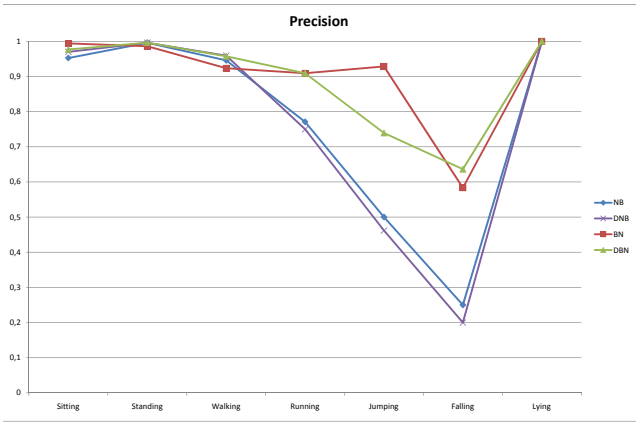
To measure the execution time, feature computation and inference were repeated 780 times. The evaluation platform was a PC with Intel Core 2 Duo microprocessor, E8400, at 3.00 GHz with 2 GB RAM running Windows XP.

The results for all classifiers which were implemented in Java are shown in Table 7 together with the length of feature computation, in order to compare the complexity of these processes. In this table, the 25th percentile, the 50th percentile, the 75th percentile, the mean, the minimum and the maximum of the execution times obtained are given. The feature computation is not time-consuming. The dif-

Static unrestricted Bayesian network recognition algorithm							
	Sitting	Standing	Walking	Running	Jumping	Falling	Lying
Recall	0.99	0.96	1	0.69	0.66	1	0.99
Precision	0.99	0.98	0.94	1	1	0.57	1
Dynamic unrestricted Bayesian network recognition algorithm							
	Sitting	Standing	Walking	Running	Jumping	Falling	Lying
Recall	1	0.98	1	0.93	0.93	1	0.98
Precision	0.97	1	0.98	1	0.93	0.8	1

Table 6. Precision and recall for static and dynamic inference in the learnt Bayesian network considering the recognition delay of 0.5 seconds.

(a)



(b)

Figure 12. Recall and precision for every activity and every recognition algorithm. The approaches using the learnt BN structure outperform the Naïve Bayes classifiers.

ference between the inference of Naïve Bayes and the unrestricted Bayesian network considering static and dynamic approaches is notable. Inference based on the Naïve Bayes network takes from 0.3 to 0.4 ms. In contrast, using the learnt Bayesian network, inference takes from 7 to 8 ms, due to the complexity of this network. The complexity of a BN is determined by the number of its nodes and the size of their corresponding CPTs. The memory size of the learnt Bayesian network and the Naïve Bayes network shown in Table 8 demonstrate the difference of complexity of both networks and explain the significant inference differences.

Inference based on the Grid-based filter for the dynamic approach takes around 1 up to 2 ms more than the static approach. Dealing with the HMM increases the inference time, but the main computational cost in terms of execution time comes from the Bayesian network used.

BN	Memory size
Naïve Bayes network	33.3 KB
Learnt Bayesian network	3.62 MB

Table 8. Memory size of the Bayesian networks after the training process (unrestricted Bayesian network) and for the assumption of Naïve Bayes.

This evaluation shows that activities can be recognized in real time. Using the grid based filter for classification which has proven best in section 6.1, the recognition time amounts in total to approximately 10 ms, which allows for classification with 100 Hz, the maximum rate of our IMU. Given that we seen that classification with 4 Hz already yields our excellent results, activity recognition in real time is even realizable on processors with less resources or running as a background process.

7 CONCLUSION AND OUTLOOK

In this paper we have demonstrated how to design a complete activity recognition system based on Bayesian techniques using acceleration and turn rate data from an IMU worn at the belt. The presented approach is unobtrusive, reliable, shows high precision and recall and can be evaluated in real time.

Our present set-up assumes a placement of the sensor array on the belt and hence once can assume that sensor orientation does not change over time. In terms of other sensor placements, we suggest that algorithms might be implemented that estimate the current orientation of the device with respect to the human body to account for shifts in orientation. A scenario where a mobile phone is carried in the pocket could be a very useful one. It is conceivable that we calibrate for different orientations during periods of walking or running where we can assume that the person is in an upright pose.

In the future we want to fuse this activity information with pedestrian positioning systems for indoor and outdoor environments to improve the actual positioning accuracy.

Operation	Q_1 (ms)	Q_2 (ms)	Q_3 (ms)	μ (ms)	Min.	Max.
Feature computation	1.43	1.45	1.47	1.5	1.4	4.1
Static Naïve Bayes estimator	0.31	0.319	0.32	0.34	0.29	2.17
Dynamic Naïve Bayes estimator	0.33	0.34	0.35	0.36	0.3	3.26
Estimator based on static, learnt BN	5.6	7.2	8.3	7.2	3.9	27.7
Dynamic Estimator for the learnt BN	6	7.7	9	7.7	4.1	18

Table 7. Execution times of feature computation and inference process from 780 runs on an Intel Core 2 Duo microprocessor E8400,at 3.00 GHz with 2 GB RAM. The 25th percentile, the 50th percentile, the 75th percentile, the mean, the minimum and the maximum of the execution times for the feature computation and inference process based on Naïve Bayes and the learnt Bayesian network show, that inference with one estimator usually stay below 10 ms.

Therefore additional activities may have to be included. In particular 'climbing stairs' could be useful in 3D positioning scenarios.

Moreover, the physical activity information can be used together with other information to infer higher level information, e.g. that a person is giving a presentation, cooking, attending a meeting or in a dangerous situation.

ACKNOWLEDGMENTS

The research leading to these results has received funding from the European Community's Seventh Framework Programme [FP7/2007-2013] under grant agreement no. 215098 of the *Persist* (PERsonal Self-Improving SmarT spaces) Collaborative Project.

We also want to thank all persons helping to record the data set.

REFERENCES

[1] K. Wendlandt, M. Khider, M. Angermann, and P. Robertson, "Continuous location and direction estimation with multiple sensors using particle filtering," in *MFI 2006*, ser. Library of Congress: 2006930785, IEEE, Ed. IEEE Verlag, September 2006. [Online]. Available: http://elib.dlr.de/44240/

[2] Widyawan, M. Klepal, and S. Beauregard, "A backtracking particle filter for fusing building plans with PDR displacement estimates," in *Proceedings of the 5th Workshop on Positioning, Navigation and Communication, 2008, WPNC08*, Hannover, Germany, Mar. 2008.

[3] A. Bernheim Brush, A. K. Karlson, J. Scott, R. Sarin, A. Jacobs, B. Bond, O. Murillo, G. Hunt, M. Sinclair, K. Hammil, and S. Levi, "User experiences with activity-based navigation on mobile devices," in *Proceedings of Mobile HCI 2010*. ACM, Sep. 2010.

[4] C. Randell and H. Muller, "Context awareness by analyzing accelerometer data," in *ISWC '00: Proceedings of the Fourth IEEE International Symposium on Wearable Computers*. Washington, DC, USA: IEEE Computer Society, 2000, p. 175.

[5] D. T. G. Huynh, "Human Activity Recognition with Wearable Sensors," Ph.D. dissertation, Technische Universität Darmstadt, August 2008.

[6] K. Aminian, P. Robert, E. Buchser, B. Rutschmann, D. Hayoz, and M. Depairon, "Physical activity monitoring based on accelerometry: validation and comparison with video observation," *Medical & Biological Engineering & Computing*, vol. 37, no. 3, pp. 304–308, 1999.

[7] M. Makikawa and H. Iizumi, "Development of an ambulatory physical activity memory device and its application for the categorization of actions in daily life," in *Proceedings of 8th conference on medical informatics*. MEDINFO, 1995, pp. 747–750.

[8] S. Kurata, M. Makikawa, M. Kawato, H. Kobayashi, A. Takahashi, and R. Tokue, "Ambulatory physical activity monitoring system," *Proceedings of MEDINFO 98*, pp. 277–281, 1998.

[9] S. Pirttikangas, K. Fujinami, and T. Nakajima, "Feature selection and activity recognition from wearable sensors," in *Ubiquitous Computing Systems*. Springer-Verlag, 2006, pp. 516–527.

[10] M. Uiterwaal, E. Glerum, H. Busser, and R. van Lummel, "Ambulatory monitoring of physical activity in working situations, a validation study," *Journal of Medical Engineering & Technology*, vol. 22, no. 4, pp. 168–172, July/August 1998.

[11] K. V. Laerhoven and O. Cakmakci, "What shall we teach our pants?" in *ISWC '00: Proceedings of the 4th IEEE International Symposium on Wearable Computers*. Washington, DC, USA: IEEE Computer Society, 2000, p. 77.

[12] J. Mantyjarvi, J. Himberg, and T. Seppanen, "Recognizing human motion with multiple acceleration sensors," in *Systems, Man, and Cybernetics, 2001 IEEE International Conference on*, vol. 2, 2001, pp. 747–752.

[13] S.-W. Lee and K. Mase, "Activity and location recognition using wearable sensors," *Pervasive Computing, IEEE*, vol. 1, no. 3, pp. 24–32, 2002. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1037719

[14] T. Stiefmeier, C. Lombriser, D. Roggen, H. Junker, G. Ogris, and G. Tröster, "Event-based activity tracking in work environments," in *Proceedings of the 3 rd IFAWC*. VDE Verlag, 2006, pp. 91–100.

[15] M. Kranz, K. Frank, D. H. Galceran, and E. Weber, "Open vehicular data interfaces for in-car context inference," in *Proceedings of AutomotiveUI 2009*, A. Schmidt and A. Dey, Eds. ACM, September 2009, pp. 57–62. [Online]. Available: http://elib.dlr.de/59737/

[16] S. S. Intille, L. Bao, E. M. Tapia, and J. Rondoni, "Acquiring in situ training data for context-aware ubiquitous computing applications," in *in Proceedings of CHI 2004 Connect: Conference on Human Factors in Computing Systems*. ACM Press, 2004, pp. 1–8.

[17] J.-Y. Yang, Y.-P. Chen, G.-Y. Lee, S.-N. Liou, and J.-S. Wang, "Activity recognition using one triaxial accelerometer: A neuro-fuzzy classifier with feature reduction," in *Entertainment Computing - ICEC 2007*. International Federation for Information Processing, 2007, pp. 395–400.

[18] T. Stiefmeier, "Real-time spotting of human activities in industrial environments," Ph.D. dissertation, TU Darmstadt, 2008.

[19] G. S. Chambers, S. Venkatesh, G. A. W. West, and H. H. Bui, "Hierarchical recognition of intentional human gestures for sports video annotation," *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, vol. 2, pp. 1082–1085, 2002.

[20] D. A. Winter, *Biomechanics and Motor Control of Human Movement*. Wiley, August 2004. [Online]. Available: http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/047144989X

[21] S. B. Kotsiantis, "Supervised machine learning: A review of classification techniques," *Informatica*, vol. 31, pp. 249–268, 2007.

[22] J. Pearl, *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2000.

[23] G. I. Webb, J. R. Boughton, and Z. Wang, "Not so naive bayes: Aggregating one-dependence estimators," *Machine Learning*, vol. 58, pp. 5–24, 2005, 10.1007/s10994-005-4258-6. [Online]. Available: http://dx.doi.org/10.1007/s10994-005-4258-6

[24] G. F. Cooper and E. Herskovits, "A bayesian method for the induction of probabilistic networks from data," *Machine Learning*, vol. 09, no. 4, pp. 309–347, October 1992. [Online]. Available: http://www.ingentaconnect.com/content/klu/mach/1992/00000009/00000004/00422779

[25] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Francisco: Morgan Kaufmann, 1988.

[26] K. P. Baclawski, "Bayesian network development," in *International Workshop on Software Methodologies, Tools and Techniques, pages 18-48. Keynote address.*, Sept 2004.

[27] E. Charniak, "Bayesian networks without tears," *AI Magazine*, vol. 12, no. 4, pp. 50–63, 1991.

[28] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Prentice Hall, 1995.

[29] H. Guo and W. Hsu, "A survey of algorithms for real-time bayesian network inference," in *the joint AAAI-02/KDD-02/UAI-02 workshop on Real-Time Decision Support and Diagnosis Systems*, 2002. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.11.5182

[30] K. Frank, M. Röckl, M. J. Vera Nadales, P. Robertson, and T. Pfeifer, "Comparison of exact static and dynamic bayesian context inference methods for activity recognition," in *Proceedings of Workshop for Managing Ubiquitous Communications and Services 2010*, 2010, pp. 41–47.

[31] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[32] J. M. Francis, F. Kubala, R. Schwartz, and R. Weischedel, "Performance measures for information extraction," in *In Proceedings of DARPA Broadcast News Workshop*, 1999, pp. 249–252.