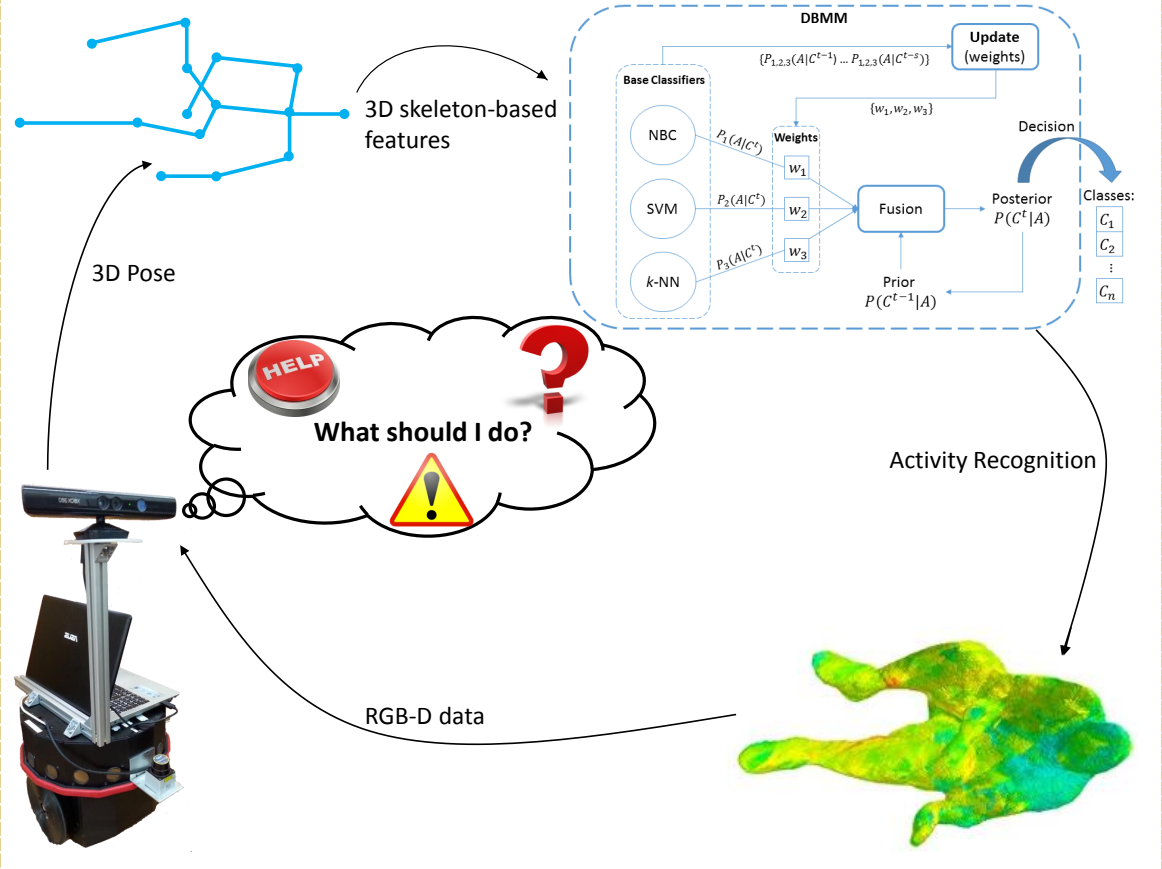


Mário Augusto da Costa Vieira

Recognition of Daily Activities and Risk Situations towards Robot-Assisted Living

September 2015

· U C ·

Universidade de Coimbra

Departamento de Engenharia Electrotécnica e de Computadores
Faculdade de Ciências e Tecnologia
Universidade de Coimbra

A Dissertation
for Graduate Study in MSc Program
Master of Science in Electrical and Computer Engineering

Recognition of Daily Activities and Risk Situations towards Robot-Assisted Living

Mário Augusto da Costa Vieira

Research Developed Under Supervision of
Prof. Doutor Urbano José Carreira Nunes and
Doutor Diego Resende Faria

Jury
Prof. Doutor Paulo José Monteiro Peixoto
Prof. Doutor Rui Paulo Pinto da Rocha
Prof. Doutor Urbano José Carreira Nunes

September 2015

Work developed in the Institute of Systems and Robotics of the University of Coimbra.

Learn from yesterday, live for today, hope for tomorrow. The important thing is not to stop questioning.
(Albert Einstein)

Acknowledgements

In this short text, I would like to thank the people without whom this journey would not have come to fruition.

First and foremost, I thank my parents and sister for always being there for me. I thank their unconditional and unwavering support throughout every day of my life, and especially to my parents for the sacrifice and effort they have made to provide me with everything necessary to get this far.

I would like to thank my advisors, Professor Urbano Nunes and Dr. Diego Faria, for their guidance, support and encouragement that made this work become a pleasant journey.

I thank to my laboratory colleagues that in one way or another supported me in this work, in particular to Jorge Perdigão, who helped me since the first day, whenever he saw me in trouble.

I would also like to thank ISR for hosting me, providing the necessary resources, conditions and personnel that allowed me to accomplish all the goals I worked for. This work has been supported by Fundação para a Ciência e Tecnologia (FCT), COMPETE and QREN programs, under the project "AMS-HMI12 - Assisted Mobility Supported by shared control and advanced Human Machine Interfaces" with reference RECI/EEI-AUT/0181/2012.

To all my friends, I thank them for their friendship and support. To those that are with me since forever and to those that have appeared throughout my academic life, I thank you for sharing with me great memories and unforgettable moments.

A special thanks to my girlfriend, Daniela, who was closer to me and my work. I thank her for sharing my happiness and success in the best of times and for her patience and encouraging attitude in not so good times.

Thank you all.

Abstract

Assisted living systems can greatly help disabled or elderly people with their daily tasks, helping them to maintain a safe, healthy and independent life. Therefore, it is essential that a personal robot is endowed with cognitive skills in order to understand what surrounds it and decide the best action to take in accordance with the situation. Recognition of activities in real-time is essential to understand the behaviour of the person being assisted and to quickly detect any risk situation. It is important not only to detect these risk behaviours, but also reacting as soon as possible, assisting the person effectively, avoiding as much damage as possible.

In this research work, an integrated artificial cognitive system was developed for a mobile robot, which all methods were implemented under the Robot Operating System (ROS). To this end, a mobile robot equipped with a Red-Green-Blue and Depth sensor (RGB-D) and a laser range finder was used. By using the RGB-D sensor is possible to detect and track the human skeleton and extract relevant spatio-temporal features in order to characterize daily activities, including risk situations. A classification module has been implemented based on a probabilistic ensemble of classifiers as well as a decision-making module for the robot to react given a recognized activity.

The entire system was tested both offline and online, i.e. either with data previously acquired (datasets) and also running on-the-fly using a mobile robot. The results attained for activity recognition in terms of accuracy, precision and recall were 93.41%, 93.61% and 92.25% for assessment on our dataset and 90.55%, 90.84% and 90.55% for testing in real time application of robot-assisted living. The activity recognition framework with the proposed skeleton-based features was also evaluated using a public state-of-the-art dataset, UTKinect Action Dataset [37], achieving a good performance compared to other state-of-the-art approaches.

Experiments have shown that the developed system has the potential to be used in robot-assisted living.

Keywords: Activity Recognition, Assisted Living, Human Skeleton, Kinect, ROS.

Resumo

Os sistemas de vida assistida podem ajudar fortemente pessoas deficientes ou idosas com as suas tarefas diárias, ajudando-as a manter uma vida segura, saudável e independente. Para isso, é indispensável que um robô pessoal seja dotado de capacidades cognitivas, de modo a perceber o que o rodeia e a decidir a melhor ação a tomar de acordo com a situação. O reconhecimento de atividades em tempo real é essencial para compreender o comportamento da pessoa a ser assistida e detetar o mais prontamente possível quaisquer situações de risco. É importante não só detetar esses comportamentos de risco, mas também agir o mais brevemente possível, assistindo a pessoa de forma eficaz, evitando o máximo de danos possível.

Neste trabalho, desenvolveu-se um sistema artificial cognitivo integrado num robô móvel onde todos os métodos foram implementados no *Robot Operating System* (ROS). Para tal, foi usado um robô móvel equipado com um sensor Red-Blue-Green and Depth (RGB-D) e um *laser range finder*. Usando um sensor RGB-D, foi possível detetar e seguir o esqueleto humano e extrair características espácio-temporais relevantes de modo a caracterizar atividades diárias, incluindo situações de risco. Foi implementado um módulo de classificação que tem por base uma fusão probabilística de classificadores e um módulo de tomada de decisão para que o robô reaja de acordo com a atividade detetada.

Todo o sistema foi testado tanto offline como online, isto é, tanto com dados previamente adquiridos e guardados como também executando em tempo real, usando um robô móvel. Os resultados obtidos para reconhecimento de atividades em termos de *accuracy*, *precision* e *recall* foram 93.41%, 93.61% e 92.25% para os testes offline e 90.55%, 90.84% e 90.55% para os testes na aplicação de tempo real para vida assistida por robôs. O módulo de reconhecimento de atividades com as características propostas baseadas no esqueleto foi também avaliado usando um dataset público do estado da arte, *UTKinect Action Dataset* [37], alcançando um bom desempenho comparado com outras abordagens do estado da arte.

As experiências realizadas mostram que o sistema desenvolvido tem potencial para ser usado em

aplicações de vida assistida por robôs.

Palavras-chave: Reconhecimento de Atividades, Vida Assistida, Esqueleto Humano, Kinect, ROS.

Contents

List of Figures

List of Tables

List of Abbreviations

2D, 3D	Two Dimensional, Three Dimensional
ANN	Artificial Neural Network
BMM	Bayesian Mixture Model
DBMM	Dynamic Bayesian Mixture Model
DMM	Depth Motions Maps
DTW	Dynamic Time Warping
FCT	Fundação para a Ciência e Tecnologia
fps	frames per second
FTP	Fourier Temporal Pyramid
GMM	Gaussian Mixture Model
Hector	Heterogeneous Cooperating Teams of Robots
HMM	Hidden Markov Models
HOG	Histogram of Oriented Gradients
HRI	Human-Robot Interaction
IR	Infrared
ISR	Institute of Systems and Robotics
***k*-NN**	*k*-Nearest Neighbours
LOOCV	Leave-one-out cross-validation

NBC	Naive Bayes Classifier
NiTE	Natural Interaction Middleware
RGB	Red, Green, Blue
RGB-D	Red, Green, Blue and Depth
ROS	Robot Operating System
SIFT	Scale-invariant Feature Transform
SLAM	Simultaneous Localization and Mapping
SSVM	Structural Support Vector Machine
SVM	Support Vector Machine

Chapter 1

Introduction

Accordingly to the United Nations [23], world population ageing is increasing every year in nearly all the countries of the world, and will keep increasing in future decades. This phenomenon results from increased average life expectancy and declining fertility. The global share of people aged 60 years or over increased from 9.2% in 1990 to 11.7% in 2013 and will continue to grow as a proportion of the world population, reaching 21.1% by 2050. 40% of these older people live alone or with their spouse only. Even the older population is ageing, with a global share of older people aged 80 years or over of 14% in 2013 and a projected share of 19% in 2050.

The increasing ageing population will result in some challenges for society and the health care system: increase in diseases, such as the Alzheimer's disease or Parkinson's disease; increase in health care costs; shortage of caregivers; dependency due to the increase of diseases. Some older people will be unable to live independently. Because of these and other issues, researchers in the robotics domain have been trying to develop technologies that allow the introduction of robots in our daily life. Assistive robots should be able to blend in with humans, being aware of the surrounding environment and interacting in a friendly and secure way. This kind of robots can act as caregivers for the elderly and disabled by helping them in their daily activities, overcoming the necessity of home nurses or family caregivers.

This dissertation describes a research work on " Recognition of Daily Activities and Risk Situations towards Robot-Assisted Living". This work aims studying classification methods and implement a classification framework capable of recognizing human activities. In order to test the system in real scenarios, a mobile robot was used to assemble an integrated system capable of assist humans.

1.1 Motivation

Recognizing human activities has been a challenging issue for researchers in the last few years. Human behaviour is an important issue in indoor environments, for several applications. In the context of security and surveillance systems, it is important to detect abnormal or suspicious behaviours. These systems can assist security personnel to detect such behaviours in crowded environments. By exploring recent advances in human pose detection using RGB-D sensors, the features are usually computed extracting the human body silhouette and 3D skeleton from depth images. After features extraction, a classification method is trained and adopted to recognize a set of activities.

In this work, we focus our attention on the domain of human-centered robotics, therefore using human activity recognition as a mean to support people. Mobile robots endowed with cognitive skills are able to help and support humans in an indoor environment, providing increased availability, awareness and access, as compared to static systems. For that, the robot must be able to understand human behaviours, distinguishing human daily routine from potential risk situations. In this context, a robot that can recognize human activities will be useful for assisted care, such as human-robot or child-robot interaction and also monitoring elderly and disabled people regarding strange or unusual behaviours.

1.2 Objectives

The primary goal of this dissertation is to implement a framework in ROS capable of recognizing human activities, using a mobile robot with an onboard RGB-D sensor. In addition, different modules are integrated with the classification framework in order to have a robot capable of autonomously navigate in an indoor environment to recognize human activities.

Thus, the main objectives are:

- Modelling discriminative skeleton-based features for activity recognition.
- Developing a module for human activity recognition in ROS
- Integrating the activity recognition module and ROS navigation packages towards monitoring the human activities.
- Developing a reaction module towards assisting a human.
- Evaluating the integrated system in real-time tests in real scenarios.

1.3 Implementations and main contributions

The following scientific question is addressed in this dissertation: "*How can an artificial system be endowed with cognitive skills in order to recognize human daily activities to monitor and assist humans?*"

In this dissertation, we tried to answer this question by understanding how humans perform some activities and what are the consequences of these activities. The focus of this work is on developing an artificial cognitive system to be executed by a mobile robot, capable of recognizing daily activities and risk situations as well as reacting accordingly to the activity being performed.

A dataset of human daily activities and risk situations was collected and relevant features were extracted from this dataset to properly characterize the activities. Several classification methods were studied and a probabilistic ensemble of classifiers proposed in [10] was implemented. The approach was tested and validated offline, using the dataset collected by us and a state of the art dataset, and on-the-fly, using a mobile robot. The results show a significant improvement on the classification performance using the adopted approach. A reaction module was developed so the robot make a decision after detecting the activity being performed. An artificial cognitive system was developed integrating ROS navigation packages, the classification framework and the reaction module.

The implementations and main contributions of the presented work are the following:

Activity Recognition Framework (Chapter 3):

- Creation of a dataset of human daily activities and risk or unusual behaviours.
- Proposed spatio-temporal skeleton-based features.

Artificial Cognitive System Implemented in ROS, as shown in figure 1.1 (Chapter 4):

- Navigation Module: A simple node for random navigation was implemented, with SLAM and collision avoidance.
- Classification Module: Implementation of the Dynamic Bayesian Mixture Model (DBMM) [9] [10] in ROS environment.
- Reaction Module: A module to endow the robot with the ability of deciding what to do after an activity be recognized.
- Combination of the different ROS modules for a robot-assisted living application.

Experimental Results (Chapter 5):

- Comparison of different single classifiers and DBMM.
- Offline validation of DBMM using leave-one-out cross validation on ”unseen” person.
- Online validation of the integrated artificial cognitive system.

In chapter 6 conclusions are drawn and guidelines for future work are provided.

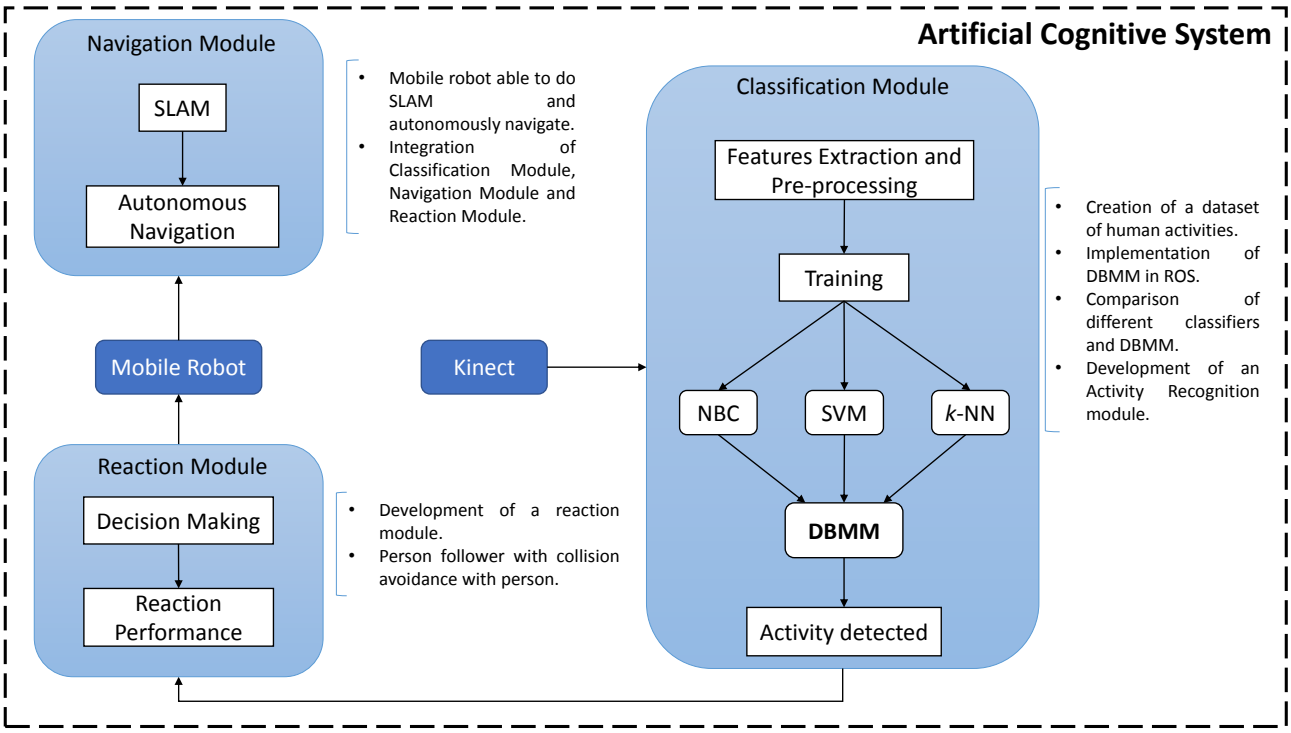


Figure 1.1: Implementations and main contributions.

Chapter 2

Background and State of the Art

This chapter describes the fundamental methodologies required to understand the work presented in this dissertation. It reviews important background theory required to develop the presented work and covers related state of the art topics, such as People Detection, Features Extraction, Activity Detection and Classification Algorithms and Activity Recognition Applications.

2.1 Background

2.1.1 Classification Methods

2.1.1.1 Naive Bayes

Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes' theorem assuming the features are independent from each other.

Bayes' theorem was first proposed by Thomas Bayes and describes the probability of an event, based on conditions that might be related to the event. Bayes' theorem can be stated mathematically as following:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}, \tag{2.1}$$

where $P(A)$ and $P(B)$ are the probabilities a priori of A and B; $P(A|B)$ and $P(B|A)$ are conditional probabilities, A given that B is true and B given that A is true, respectively. Using Bayesian probability terminology, equation (2.1) can be written as

$$posterior = \frac{likelihood \times prior}{evidence} \tag{2.2}$$

Given a class variable C and a dependent feature vector A_1 through A_n, Bayes' theorem states the following relationship:

$$P(C|A_1,...,A_n) = \frac{P(C)P(A_1,\ldots,A_n|C)}{P(A_1,\ldots,A_n)} \tag{2.3}$$

Using the naive independence assumption that

$$P(A_i|C,A_1,\ldots,A_{i-1},A_{i+1},\ldots,A_n) = P(A_i|C), \tag{2.4}$$

for all i, this relationship is simplified to

$$P(C \mid A_1,\ldots,A_n) = \frac{P(C)\prod_{i=1}^{n} P(A_i \mid C)}{P(A_1,\ldots,A_n)} \tag{2.5}$$

Thus, the independent feature model, that is, the naive Bayes probability model is obtained. The Naive Bayes Classifier (NBC) combines this model with a decision rule, usually using the maximum a posteriori (MAP) estimation [14] to estimate $P(C|A)$ and $P(A_i|C)$. The corresponding classifier, is the function that assigns a class label $\hat{y} = C$ as follows:

$$\hat{y} = \underset{C}{\operatorname{argmax}}\, P(C)\prod_{i=1}^{n} P(A_i|C). \tag{2.6}$$

2.1.1.2 Support Vector Machine

Support Vector Machines (SVM) are supervised learning models widely used for classification. The first SVM algorithm was proposed by Vladimir N. Vapnik and Alexey Ya. Chervonenkis in 1963, but only in 1995 the current version (soft margin) was published by Corinna Cortes and Vapnik [6].

A linear SVM tries to separate two different classes using a straight line (Figure 2.1). That straight line is determined, selecting two hyperplanes so the gap between the two classes is as wide as possible.

In order to do that, a vector $\bar{w}$ is defined as the normal vector to the hyperplane. However its magnitude is unknown and some steps are necessary in order to find the right $\bar{w}$. We can say, without loss of generality that for an unknown sample $\bar{u}$, if

$$\bar{w}\cdot\bar{u} + b \geq 0, \text{then the sample belongs to the first class} \tag{2.7}$$

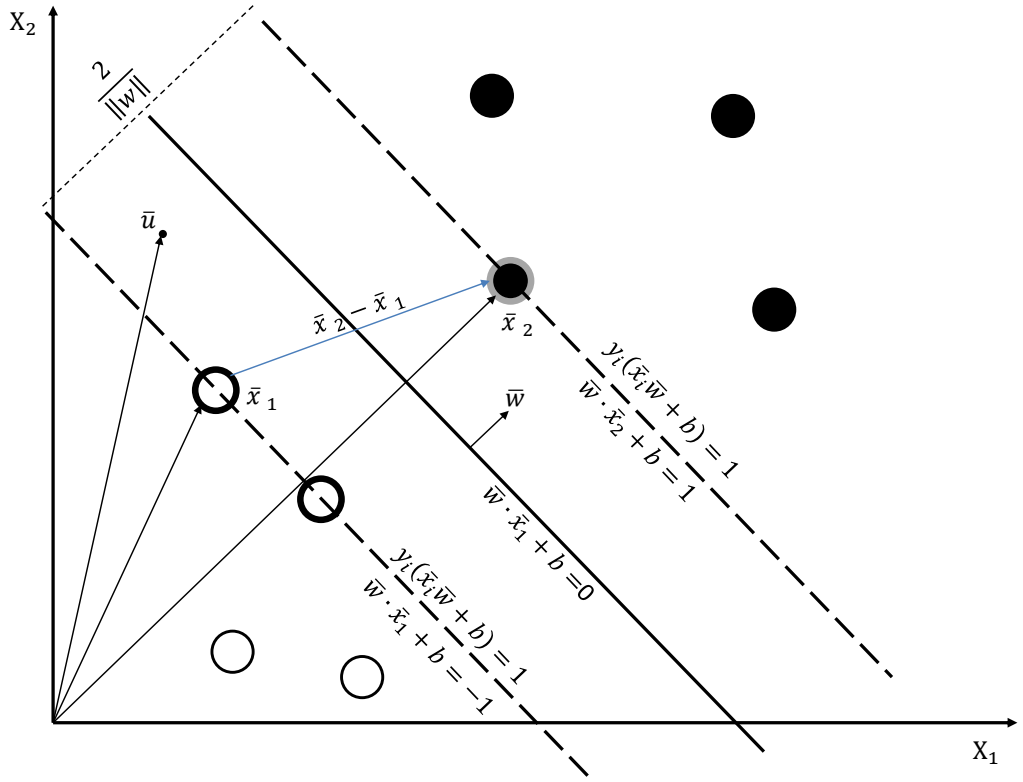


Figure 2.1: Maximum-margin hyperplane and margins for an SVM trained with samples from two classes. The white circles belong to the first class and the black circles belong to the second class.

where b is a constant that needs to be determined. Additional constraints are necessary in order to calculate $\bar{w}$ and b:

$$\begin{cases} \bar{w} \cdot \bar{x}_i + b \leq -1, \text{for } \bar{x}_i \text{ of the first class} \\ \bar{w} \cdot \bar{x}_i + b \geq 1, \text{for } \bar{x}_i \text{ of the second class} \end{cases} \tag{2.8}$$

For mathematical convenience a new variable y_i is introduced such that $y_i = 1$ for samples of the first class and $y_i = -1$ for samples of the second class. Multiplying the respective y_i in (2.8) we obtain:

$$\begin{cases} y_i(\bar{w} \cdot \bar{x}_i + b) \geq 1, \text{for } \bar{x}_i \text{ of the first class} \\ y_i(\bar{w} \cdot \bar{x}_i) + b) \geq 1, \text{for } \bar{x}_i \text{ of the second class} \end{cases} \tag{2.9}$$

Now we have the same equation for samples of both classes. From (2.9),

$$y_i(\bar{w} \cdot \bar{x}_i + b) - 1 \geq 0 \Rightarrow y_i(\bar{w} \cdot \bar{x}_i + b) = 1 \text{ for } \bar{x}_i \text{ in the margin} \tag{2.10}$$

In order to find the distance between the two hyperplanes of Figure 2.1, we can compute:

$$width = (\bar{x}_1 - \bar{x}_2) \cdot \frac{\bar{w}}{\|\bar{w}\|}, \tag{2.11}$$

where $\bar{x}_1$ and $\bar{x}_2$ are support vector, i.e., a sample of the first class and second class, respectively on the margins of the hyperplane. Using equation (2.10) it is possible to conclude that $\bar{x}_1 \cdot \bar{w} = 1 - b$ and $-\bar{x}_2 \cdot \bar{w} = 1 + b$. Hence, from equation (2.11), $width = \frac{2}{\|\bar{w}\|}$. The goal is to maximize this width, so:

$$max\left(\frac{2}{\|\bar{w}\|}\right) \Leftrightarrow max\left(\frac{1}{\|\bar{w}\|}\right) \Leftrightarrow min\left(\|\bar{w}\|\right) \Leftrightarrow min\left(\frac{1}{2}\|\bar{w}\|^2\right). \tag{2.12}$$

To solve this quadratic optimization problem, introducing the Lagrange multipliers α is necessary:

$$L = \frac{1}{2}\|\bar{w}\|^2 - \sum_{i=1}^{n} \alpha_i \left[y_i(\bar{w} \cdot \bar{x}_i + b) - 1\right]. \tag{2.13}$$

Then it is necessary to partial derive L with respect to anything that might vary, i.e., $\bar{w}$ and b, and equalize to 0:

$$\frac{\partial L}{\partial \bar{w}} = \bar{w} - \sum_{i=1}^{n} \alpha_i y_i \bar{x}_i = 0 \Rightarrow \bar{w} = \sum_{i=1}^{n} \alpha_i y_i \bar{x}_i, \tag{2.14}$$

$$\frac{\partial L}{\partial b} = -\sum_{i=1}^{n} \alpha_i y_i = 0 \Rightarrow \sum_{i=1}^{n} \alpha_i y_i = 0. \tag{2.15}$$

From (2.14), it is possible to see that $\bar{w}$ is expressed as a linear combination of the training samples. Only some α_i will be greater than 0 and the corresponding $\bar{x}_i$ are exactly the support vectors, which lie on the margin and satisfy equation (2.10). Replacing in (2.13) the value of $\bar{w}$ found in (2.14),

$$\begin{aligned} L &= \frac{1}{2}\left(\sum_{i=1}^{n} \alpha_i y_i \bar{x}_i\right)\left(\sum_{j=1}^{n} \alpha_j y_j \bar{x}_j\right) - \sum_{i=1}^{n} \alpha_i y_i \bar{x}_i \left(\sum_{j=1}^{n} \alpha_j y_j \bar{x}_j\right) - \underbrace{\sum_{i=1}^{n} \alpha_i y_i b}_{0} + \sum_{i=1}^{n} \alpha_i \\ &= \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=0}^{n}\sum_{j=0}^{n} \alpha_i \alpha_j y_i y_j \bar{x}_i \cdot \bar{x}_j \end{aligned} \tag{2.16}$$

At this stage, the maximization of expression (2.16) can be achieved by the use of the standard quadratic programming method described in [33]. Once the vector $\alpha^* = (\alpha_1^*, \alpha_2^*, ..., \alpha_N^*)$ solution of the maximization problem has been found, the optimal separating hyperplane is given by,

$$w^* = \sum_{i=1}^{n} \alpha_i^* y_i x_i, \tag{2.17}$$

$$b^* = -\frac{1}{2} \langle w^*, x_r + x_s \rangle, \tag{2.18}$$

where x_r and x_s are any support vector from each class satisfying, $\alpha_r, \alpha_s > 0$ and $y_r = -y_s = 1$.

If the samples are not linearly separable, it is necessary to perform a transformation ϕ from the current space into a space where things are more convenient (Figure 2.2). So now, it is necessary to maximize $\phi(\bar{x}_i) \cdot \phi(\bar{x}_j)$ and in order to do that, all we need is a function called Kernel function:

$$K(\bar{x}_i, \bar{x}_j) = \phi(\bar{x}_i) \cdot \phi(\bar{x}_j). \tag{2.19}$$

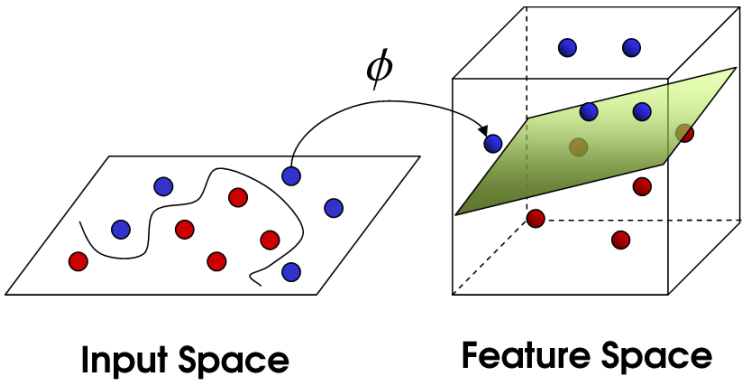


Figure 2.2: Transformation of spaces using a Kernel function [1].

There are several Kernel functions, such as:

- Polinomial: $K(\bar{x}_i, \bar{x}_j) = (c + \bar{x}_i \cdot \bar{x}_j)^d$, where c is a constant and d is the polynomial degree.
- Radial basis function (Gaussian): $K(\bar{x}_i, \bar{x}_j) = e^{-\frac{\left\|\bar{x}_i - \bar{x}_j\right\|^2}{2\sigma^2}}$, where σ is a free parameter that should be tuned.
- Sigmoid: $K(\bar{x}_i, \bar{x}_j) = \tanh(\gamma \bar{x}_i \cdot \bar{x}_j + c)$, where γ is the slope.

In multi-class classification there are two major methods to use SVM: ”one-against-all” and ”one-against-one” approaches [17]. The first method consists of constructing one SVM per class which is trained to distinguish the samples of one class from the samples of all remaining classes. The second

method consists in constructing one SVM for each pair of classes. If n_{class} is the number of classes, then $n_{class} * (n_{class} - 1)/2$ SVMs are constructed and each one trains data from two classes.

2.1.1.3 *k*-Nearest Neighbours

The k-Nearest Neighbours (*k*-NN) algorithm is among the simplest of all machine learning algorithms. The basic idea of this method was proposed in 1951 by Fix and Hoges [12], and formalized by Royall [27]. *k*-NN is a non parametric algorithm, meaning that it does not make any assumptions about the probability distributions of the variables being assessed. This is very useful, since in real world, most of the practical data does not obey the typical theoretical assumptions made. *k*-NN is also a lazy algorithm because it does not use the training data points to do any generalization. So, the training phase is minimal and the target function is approximated locally. The disadvantage of this kind of algorithms is that the testing phase requires more space, time and memory.

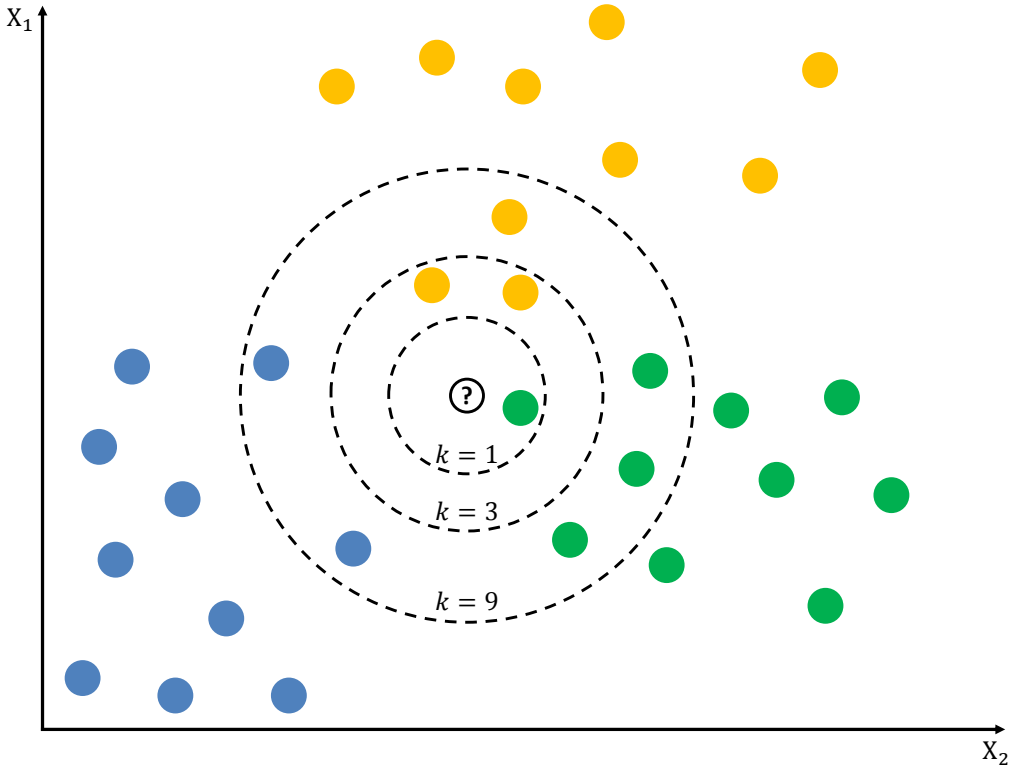


Figure 2.3: Example of *k*-NN classification.

The training set comprises vectors in a multidimensional feature space, each with a class label. On the other hand, the test set comprises unlabelled vectors, also in a multidimensional feature space. *k* is a user-defined constant and, in order to classify an unlabelled vector, the algorithm assigns the most frequent label among the *k* training samples nearest to that unlabelled vector. There are several distance

metrics, but the most widely used is Euclidean distance:

$$\delta(p,q) = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2 + (p_z - q_z)^2}. \tag{2.20}$$

Figure 2.3 illustrates an example of k-NN classification. For a $k = 1$ the unknown sample is classified as "green" because the nearest circle is green, while for a $k = 3$ is classified as "yellow" (one green circle vs. two yellow circles). For $k = 9$ a third class appears to vote, however the green class wins again with four votes against three yellow and two blue.

2.1.1.4 Dynamic Bayesian Mixture Model

Dynamic Bayesian Mixture Model (DBMM) was first proposed in [9], in order to increase classification performance on human activity recognition combining single (base) classifiers. In [10], the DBMM is extended by using the memory of the system for dynamic update of the weighted ensemble, adjusting the weights based on previous behaviours of the base classifiers.

DBMM is a dynamic probabilistic ensemble of classifiers that uses the concept of Bayesian Mixture Models (BMM) in a dynamic form in order to combine conditional probability outputs (likelihoods) from different single classifiers (Figure 2.4). A mixture model is a probabilistic model that assumes all the data points are generated from a mixture of a finite number of probability distributions. The general BMM is given as follows:

$$P(A) = \sum_{i=1}^{n} w_i \times P_i(A), \tag{2.21}$$

where n is the number of components (here represented by the number of classifiers); w_i is the weight of each Bayesian classifier output $P_i(A)$, and $\sum_{i=1}^{n} w_i = 1$.

The DBMM comprises a set of models $A = \{A_m^1, A_m^2, ..., A_m^T\}$, where A_m^t is a model with m attributes; i.e., observed variables generated for some dynamic process at each time instant $t = \{1, 2..., T\}$. The DBMM's general probability distribution function for each class C can be written as follows:

$$P(C,A) = \prod_{t=1}^{T} P(C^t|C^{t-1}) \times \sum_{i=1}^{n} w_i^t \times P_i(A|C^t). \tag{2.22}$$

Assuming that the process holds the Markov property (recursion) by taking the posterior of the previous time instant as the prior for the present time instant, (2.22) can be rewritten as follows:

$$P(C|A) \;=\; \beta \;\times\; \underbrace{P(C^t|C^{t-1})}_{\text{dynamic transitions}} \;\times\; \underbrace{\sum_{i=1}^{n} w_i^t \times P_i(A|C^t),}_{\text{mixture model with dynamic w}} \tag{2.23}$$

$$\text{with} \begin{cases} P(C^t|C^{t-1}) \equiv \frac{1}{C} \;\; (\text{uniform}), & t = 1 \\ P(C^t|C^{t-1}) = P(C^{t-1}|A), & t > 1 \end{cases},$$

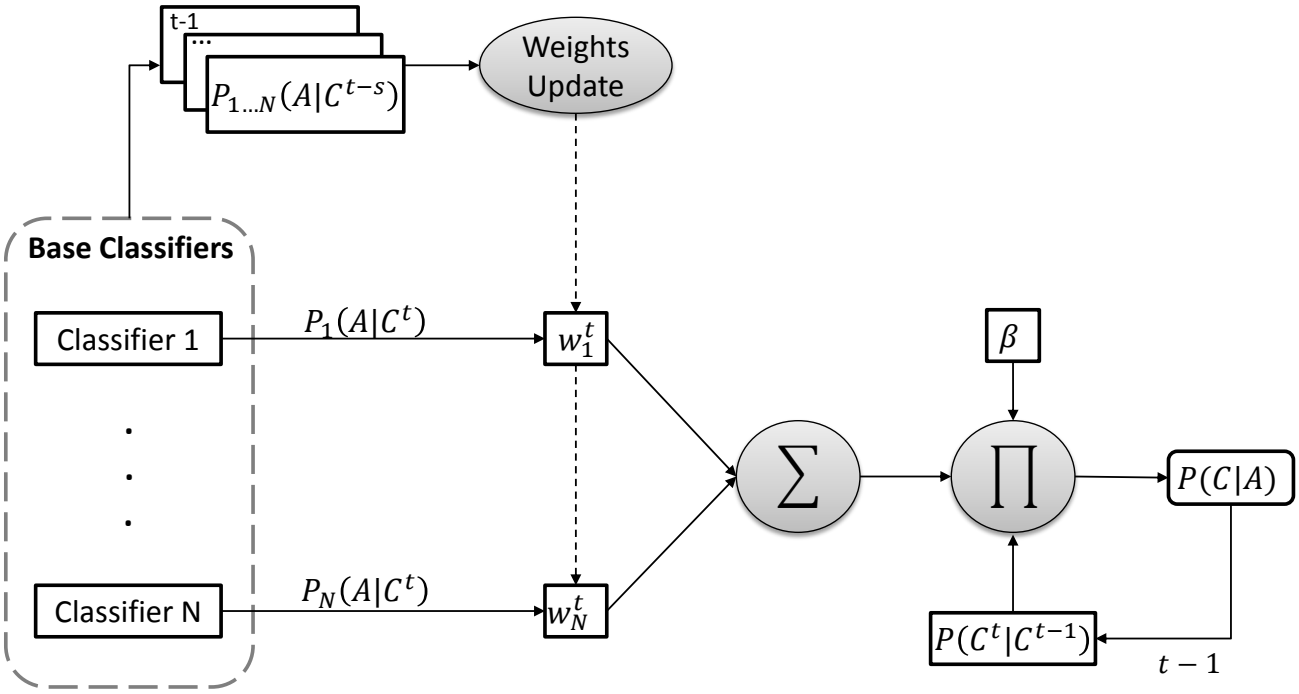


Figure 2.4: DBMM diagram.

where:

- $P(C^t \mid C^{t-1})$ is the transition probability distribution among class variables over time. A class C^t is conditioned to C^{t-1}.

- $P_i(A \mid C^t)$ is the posterior result of each base classifier at time t, $i = \{1, ..., n\}$.

- The weight in the model for each base classifier w_i^t is estimated using an Entropy-based confidence measure [9], and afterwards it is updated as explained in the next subsection.

- $\beta = \frac{1}{\sum_j \left(P(C_j^t|C_j^{t-1}) \times \sum_{i=1}^{n} w_i \times P_i(A|C_j^t)\right)}$ is a normalization factor, ensuring numerical stability once continuous update of belief is done.

Dynamic Update of Weights A weight is assigned to each base classifier, according to previous knowledge. Since base classifiers can change the performance over time, the local update of the weights will produce a higher belief when priority is assigned to a base classifier with more confidence on previous classifications. Assuming the memory of the system, there is a temporal information on the test set that contains previous posteriors for each base classifier $\Omega^s = \{P(A|C^{t-1})...P(A|C^{t-s})\}$. This information can be used together with the weights at the previous time instant w_i^{t-1} to update the ensemble model. The memory of the system is used during the classification by keeping the previous posteriors, and consequently, the entropy is acquired on each set of posteriors $H_i(\Omega^s)$ as follows:

$$H_i(\Omega^s) = -\sum_{j}^{s} \Omega^j \log(\Omega^j). \tag{2.24}$$

Knowing $H_i(\Omega^s)$ for each base classifier, the weights $P(w_i|H_i(\Omega^s))$ are estimated inversely proportional to the entropy:

$$P(w_i|H_i(\Omega^s)) = \frac{\left[1 - \left(\frac{H_i(\Omega^s)}{\sum_{i=1}^{n} H_i(\Omega^s)}\right)\right]}{\sum_{i}^{n}\left[1 - \left(\frac{H_i(\Omega^s)}{\sum_{i=1}^{n} H_i(\Omega^s)}\right)\right]},\ i = \{1,...,n\}, \tag{2.25}$$

where w_i^t is the result for each base classifier, and H_i is the current value of entropy given by (2.24). The denominator in (2.25) ensures that $\sum_i w_i = 1$. The following expression updates the current weights:

$$w_i^t = \frac{w_i^{t-1} \times P(w_i|H_i(\Omega^s))}{\sum_{i=1}^{n} w_i^{t-1} \times P(w_i|H_i(\Omega^s))}, \tag{2.26}$$

where w_i^t is the estimated weight for each base classifier (updated) and w_i^{t-1} is the previous weight at $t-1$.

2.1.2 Log-covariance matrices

The idea of log-covariance is based on [15], where examples of manifold Riemannian metrics and log-covariance applied in 2D image features for activity recognition were used. The rational behind of log-covariance is the mapping of the convex cone of a covariance matrix to the vector space by using the matrix logarithm as proposed in [4] (Figure 2.5). A covariance matrix form a convex cone, so that it does not lie in Euclidean space, e.g., the covariance matrix space is not closed under multiplication with negative scalars.

The log-covariance matrix L_S of a covariance matrix C_S is computed as follows. Suppose that the eigen-decomposition of C_S is given by $C_S = VDV^{'}$, where the columns of V are orthonormal eigenvectors and D is the diagonal matrix of eigenvalues. Then $L_S = log(C_S) = V\tilde{D}V^{'}$, where $\tilde{D}$ is a diagonal matrix obtained from D by replacing D's diagonal entries by their logarithms.

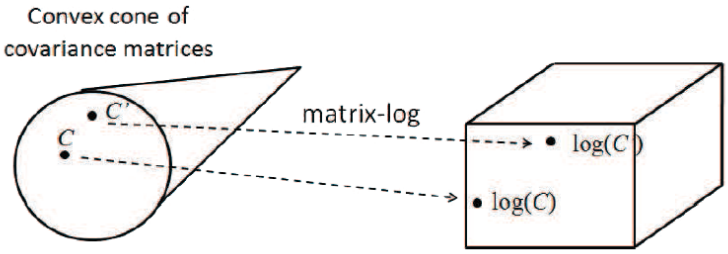


Figure 2.5: Matrix logarithm that maps covariance matrices from a convex cone to Euclidean space [15].

2.1.3 Activity classification measures

In this dissertation, three different classification performance measures were adopted to evaluate and compare our classification framework: accuracy, precision and recall [11]. In the classification context, the terms true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) compare the predictions of the classifier with the ground truth.

Accuracy is the proportion of both true positives and true negatives among the total number of cases examined, obtaining the following expression:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}. \tag{2.27}$$

Precision is the proportion of predicted positive cases that are correctly true positives, as shown below:

$$\text{Precision} = \frac{TP}{TP+FP}. \tag{2.28}$$

Recall is the proportion of true positive cases that are correctly predicted positive, obtained as follows:

$$\text{Recall} = \frac{TP}{TP+FN}. \tag{2.29}$$

2.2 State of the Art

Automatic human activity recognition has drawn much attention in the Robotics research community due to the growing demands from many applications, such as surveillance environments, assisted-living environments and Human-Robot Interaction (HRI).

There are several methods to recognize human activities, but the most common approach is through visual information. With the appearance of low cost vision sensors, these have become popular in the Robotics research community. They provide not only RGB information but also Depth information (RGB-D sensors), which is a powerful tool for a variety of applications, such as human activity recognition.

2.2.1 People Detection

The first step in human activity recognition is to detect the human body. Some methods are based solely on 2D visual information provided by cameras [21]. Local descriptors can be more robust to noise and occlusion scale, such as scale-invariant feature transform (SIFT) features [28] and histogram of oriented gradient (HOG) features [8]. More recently, with the popularization of RGB-D sensors much research has been done on people detection from depth information. In [36] a Microsoft Kinect is used to human detection, utilizing depth information only. In [22], a multipeople tracking algorithm designed to be applied on mobile service robots using RGB-D data is proposed. The data is processed by a detection module that filters the point cloud data, removes the ground and performs a 3D clustering of the remaining points. Then, a HOG-based people detection algorithm is applied to the projection onto the RGB image of the 3D clusters extended till the ground. Hence, a set of detections enters in the tracking module that performs detection-track association as a maximization of a joint likelihood composed by motion, colour appearance and people detection confidence.

2.2.2 Features Extraction

The second step for human activity recognition is the feature extraction, where meaningful characteristics of image frames are extracted in order to properly describe an activity. Features extraction is a crucial step in human activity recognition, since it has a lot of influence on the performance of the classifier.

There are several methods to extract features from image frames based on RGB and depth image or based on the 3D skeleton model provided by some RGB-D sensors. Local descriptors are also used to extract good features for human activity recognition. In [32], the human pose and motion are used as features computed from the skeleton model. Together with this set of features, HOG features descriptors are also used on both RGB and depth images. In [5], depth motions maps (DMM) are used as features, where each 3D depth frame is used to generate three 2D projected maps corresponding to front, side, and top views. For each projected map, the motion energy is calculated as the absolute difference between two consecutive maps. In order to reduce the intraclass variability, for example due to different subject heights, bicubic interpolation is used to resize all DMMs under the same projection view to a fixed size. Pixel values are then normalized between 0 and 1 and used as features. Using only features extracted from the 3D skeleton, it is possible to attain great performance on human activity recognition. In [9], it is considered the skeleton frame of reference, obtaining all joints relative to the torso centroid instead of using the sensor frame of reference. Thus, a set of 14 features are computed: the distances between hands and face, between the left and right hands, shoulders and feet, hip and feet, distance between the initial position of the hands at initial time and the next frames, using the Euclidean distance; the distance of the two hands to the face at the same time; the torso inclination; the difference between the initial hand position at initial time (for left and right hands) and the consecutive frames, as well as the left and right elbows and the head in x and y coordinates.

2.2.3 Activity Recognition Approaches

After extracting proper features, activity classification is the last step to take into account for human activity recognition. To achieve a good performance, it is essential to choose a suitable classification approach, using the extracted features. Basically, the algorithms can be divided into generative models and discriminative models. Generative models can generate synthetic data points and learn a model of class-conditional probability distribution functions and make their predictions. Popular generative models are Naïve Bayes, Hidden Markov Models (HMM), Bayesian networks and Mixture Models, such as Gaussian Mixture Model (GMM). On the other hand, discriminative models directly estimate posterior probabilities. Popular models are SVMs, artificial neural networks (ANNs) and k-NNs.

In [26], a GMM-based HMM is used to infer the human activities, using 3D positions of each skeleton joint. The authors evaluated their approach in a publicly available dataset, however the results attained are overcome by other state of the art approaches used in the same dataset. In [19], the human

activities and object affordances are modelled as a Markov random field where the nodes represent objects and sub activities, and the edges represent their relations over time. Classification is done using a structural SVM (SSVM) classifier. The authors evaluated their approach using a publicly available dataset and a dataset collected by them. The results attained for the first dataset shows the potential of their approach, but even so there are other works in the state of the art with higher classification performance. On the other hand, the results obtained for the collected dataset presents higher performance. The authors also test their approach in a real situation, using a robot to assist people, however they use object interaction to decide how the robot should react. In our work, we do not use object interaction but only the activity being performed to decide how the robot should react. To estimate unobserved actions, the authors in [35] use Bayesian Networks (BN) that integrate the evidence given by the observations. All extracted features are modelled as probability distributions and processed by seven different BN to estimate seven actions. All observations from the user model are integrated into the BN and the sum-product algorithm is applied. The authors evaluate their approach only in real time, using a mobile robot. Although the classification results are not very good, they show potential for a real time application. Other works use the combination of several methods to improve classification performance. In [34], a combination of dynamic time warping (DTW), Fourier temporal pyramid (FTP) representation and linear SVM is employed, whereas in [10] a probabilistic ensemble of classifiers is proposed having a Naive Bayes, a linear-kernel multiclass SVM and an ANN as base classifiers. The co-authored work presented in [10] was evaluated using two well known state of the art daily activity datasets, outperforming other state of the art approaches. The authors also evaluate their approach in real time, using a mobile robot, showing potential for a robot-assisted living application. However, differently of this research, their work has different feature models, and brings no reaction module of the robot, since their framework is focused on the activity recognition.

2.2.4 Activity Recognition Applications

As previously mentioned, human activity recognition has a wide range of applications. In the context of security and surveillance, one of the first objectives is to detect and track people, so as to support security personnel. Security surveillance systems endowed with automatic activity recognition can detect suspicious behaviours and create an alert immediately when security events are detected in order to prevent potentially dangerous situations. Some researchers perform detection of various types of violent behaviours such as fighting, punching, stalking [20], [24], [29].

Table 2.1: Some works in activity recognition in the last few years.

Work	Features	Classification Method	Described Application
Faria et al. [9] [10]	Spatio-temporal skeleton-based features.	Probabilistic ensemble of single classifiers (e.g., SVM, NBC, ANN) as a dynamic mixture model considering the Bayesian probability.	Monitoring of daily human activities and future work address to a robotic application in the scope of assisted living.
Vemulapalli et al. [34]	3D Skeleton-based features.	Combination of DTW, FTP representation and linear SVM.	NA.
Chen et al. [5]	Depth Motion Maps.	l_2-regularized collaborative representation classifier.	NA.
Piyathilaka et al. [26]	3D Skeleton-based features.	GMM based HMM.	NA.
Koppula et al. [19]	Object-based features; 3D Skeleton-based features; Object and skeleton temporal features.	Structural support vector machine (SSVM).	Robotic applications assisting humans: React according to the activity being performed; Proper manipulation of objects, knowing their affordances.
Sung et al. [32]	3D Skeleton-based features and HOG features descriptors.	Maximum entropy Markov model (MEMM).	NA.
Volkhardt et al. [35]	HOG detector to detect the user's pose; Motion histogram; Structural knowledge by localizing the user with respect to predefined room and object maps of the environment.	Bayesian Networks.	Mobile companion robot.
Lin et al. [20]	CBS(Change of Body Size) and Speed.	A GMM classifier is used for each feature vector. A Confident Frame-based Recognition algorithm (CFR) combines results from the multiple GMM classifiers and gives the recognition results.	Video Surveillance.

Another important application is in the context of assisted-living. Assisted living systems can help to support elderly and disabled people with their daily activities in order to help them maintain a healthy, safe and independent life. In a more specific way, robots endowed with activity recognition skills can continuously monitor the person and assist in simple daily activities as well as detect risk or unusual behaviours. In [19], a robotic application is presented to assist humans. One of the scenarios presented is taking medicine: a person opens the medicine container, takes the medicine, and waits as there is no water nearby. The robot assists the subject by bringing a glass of water on detecting the "taking medicine" activity.

Table 2.1 shows some works in activity recognition proposed recently. It summarizes the features used, as well as the classification method and applications.

Chapter 3

Activity Recognition Framework

This chapter describes the necessary steps to develop the proposed activity recognition framework. An overview of the sensor used for data acquisition is done and the collected dataset is described. The proposed set of features for activity recognition is explained in detail and finally the probabilistic classification model adopted is described.

3.1 Data Acquisition

3.1.1 Microsoft Kinect

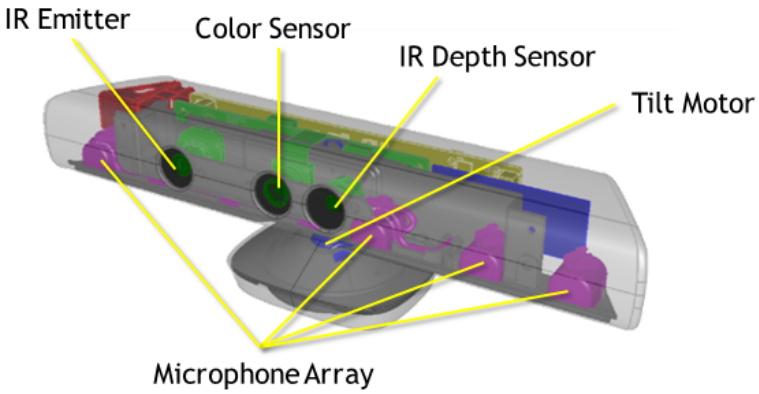


Figure 3.1: Kinect and its components [2].

In order to recognize human activities, it is required some kind of sensor to aquire data from the environment. In this work, a Microsoft-Kinect is used, due to its low cost and its capability to provide RGB images and depth information simultaneously. Given its interesting characteristics, it is suitable

for robotic applications in indoor environments. Kinect and its main components can be seen in Figure 3.1. It has an RGB camera, an infrared (IR) emitter and an IR depth sensor (IR camera), a multi-array microphone and a motorized tilt system.

Kinect's base technology for depth measurements is structured light. The IR emitter emits infrared light beams in a pattern of speckles that are reflected back to the sensor and read by the IR camera. This reflected pattern is correlated against a reference pattern stored in the memory of the Kinect, obtained by capturing a plane at a known distance from the sensor. For each speckle projected on an object whose distance is different than that of the reference plane, its position in the IR image will be shifted, originating a disparity image. From the disparity image, it is possible to compute the distance to the sensor, and therefore the 3D coordinates for each pixel, applying a triangulation method [18]. Figure 3.2 helps to understand how this method works and how can depth be obtained. The depth coordinate system has its origin at the perspective center of the IR camera and k is an object point. As this point is closer to the sensor than the reference plane, the location of the speckle on the image plane will be displaced D in the X direction and a disparity d will be measured by the IR camera. From the similarity of triangles the following relations can be obtained:

$$\frac{D}{b} = \frac{Z_o - Z_k}{Z_o}, \tag{3.1}$$

and

$$\frac{d}{f} = \frac{D}{Z_k}, \tag{3.2}$$

where Z_k is the depth of the point k, b is the base length and f is the focal length of the IR camera. Substituting D from equation (3.2) into equation (3.1):

$$\frac{Z_k d}{fb} = \frac{Z_o - Z_k}{Z_o} \Leftrightarrow \frac{Z_o d}{fb} = \frac{Z_o}{Z_k} - 1 \Leftrightarrow Z_k = \frac{Z_o}{1 + \frac{Z_o d}{fb}}. \tag{3.3}$$

Equation (3.3) allows to compute depth from the constant parameters determined by calibration Z_o, f and b.

Accordingly to [18], the expected error on Kinect's depth measurements is proportional to the distance squared, as well as the depth resolution. Both the resolution of the RGB and depth image can go up to 640 x 480 pixels per frame at 30 frames per second (fps). However, Kinect has a limited depth range of 0.8 to 4 meters in which the measures can be quite accurate. Even in indoor applications, this limited range can be a barrier difficult to overcome. Table 3.1 summarizes some Kinect specifications.

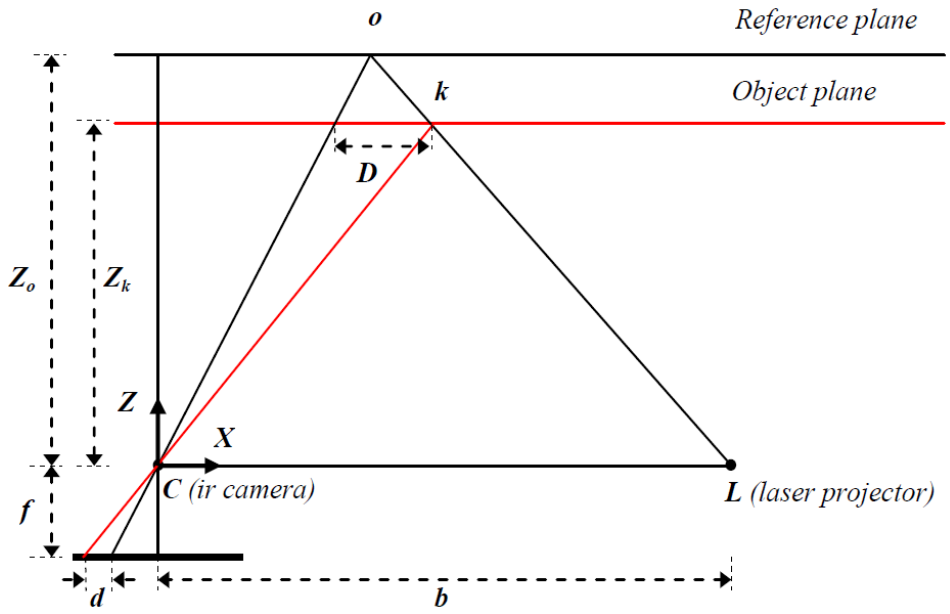


Figure 3.2: Depth and disparity relation [18].

Table 3.1: Kinect specifications [2] [3].

Feature	Value
RGB image resolution	640 x 480 @30fps
Depth image resolution	640 x 480 @30fps
Depth operation range	0.8m-4m
Viewing angle	43°vertical by 57°horizontal field of view
Vertical tilt range	±27°

3.1.2 Dataset of Daily Activities and Risk Situations

A dataset of daily activities and risk situations was acquired to train the activity recognition framework. This dataset (Figure 3.3) comprises video sequences of two male subjects and two female subjects performing eight different activities in a living-room. The sequences were taken using a stationary Kinect at 30fps that records the skeleton joints coordinates. The daily activities are: *1-walking, 2-standing still, 3-working on computer, 4-talking on the phone, 5-sitting down*; and the unusual or risk situations are: *6-jumping, 7-falling down, 8- running*. Altogether, the dataset contains 28013 frames of samples spread by the 8 activities as shown in Table 3.2.

This dataset is a challenging one, once there is significant intraclass variation among different realizations of the same activity. For example, sometimes the phone is held with the left hand while sometimes is held with the right hand. Another challenging feature is that the activity sequences are registered from different views, i.e., from the front, back, left side, and so on.

Table 3.2: Number of frames in the dataset.

Activity	Number of frames
walking	3961
standing still	4214
working on computer	3826
talking on the phone	3155
running	2088
jumping	2987
falling down	2632
sitting down	5150
Total	28013

Figure 3.3: Few examples of the dataset (RGB with skeleton joints and depth images) which was created to learn some daily and risk situations.

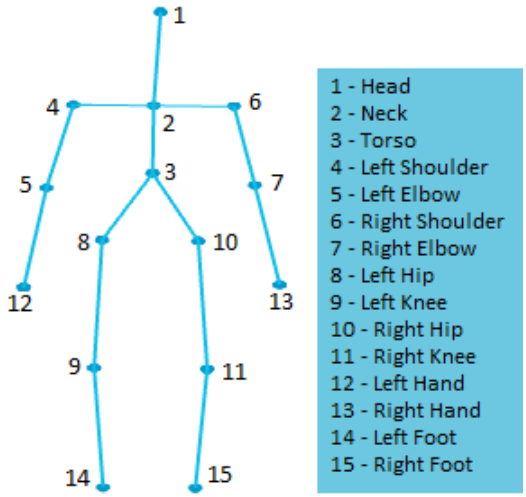


Figure 3.4: Fifteen joints skeleton model provided by the OpenNi's tracker package for ROS.

3.2 3D Skeleton-based Features

We can detect and track the human skeleton using the Microsoft Kinect and the OpenNi's tracker package for ROS. This package tracks the skeleton at a rate of 30 frames per second providing the three-dimensional Euclidean coordinates of fifteen joints of the human body, with respect to the sensor as shown in Figure 3.4.

Using this information, a set of proposed features for activity recognition is computed as follows:

- Euclidean distances among the joints, all relative to the torso centroid, obtaining a 15×15 symmetric matrix with a null diagonal. Let *(x,y,z)* be the 3D coordinates of two body joints b_j with $j = 1, 2, ..., 15$ and b_i with $i = 1, 2, ..., 15$, then $\forall \{b_i, b_j\}$, the distances were computed as follows:

$$\delta(b_j, b_i) = \sqrt{(b_j^x - b_i^x)^2 + (b_j^y - b_i^y)^2 + (b_j^z - b_i^z)^2} \tag{3.4}$$

  Subsequently, the null diagonal is removed, obtaining a 14×15 matrix $\mathbf{M}$ to compute its *log-covariance* as follows:

$$\mathbf{M}_{lc} = \mathbf{U}(\log(\text{cov}(\mathbf{M}))), \tag{3.5}$$

  where $\text{cov}(\mathbf{M}_{i,j}) = (M_i - \mu_i)(M_j - \mu_j)$; $\log(\cdot)$ is the matrix logarithm function (logm) and $\text{U}(\cdot)$ returns the upper triangle matrix composed by 120 feature elements.

- The global skeleton velocities, assuming the 3D coordinates of 14 joints in the case of having the torso centroid as origin; and 15 joints in the case of having the sensor frame as origin were computed as follows:

$$v_j = \frac{\sqrt{(b^t_{j_x} - b^{t-t_w}_{j_x})^2 + (b^t_{j_y} - b^{t-t_w}_{j_y})^2 + (b^t_{j_z} - b^{t-t_w}_{j_z})^2}}{f_{rate} \times t_w}, \tag{3.6}$$

  where v_j is the velocity of a specific skeleton joint j; b_{j_d} represents the position $d = \{x, y, z\}$ of a skeleton body joint j in the current time t, and $t - t_w$ represents some preceding frames, herein $t_w = 10$. If t_w is too big, important information is lost. On the other hand, if t_w is to small, irrelevant data will be used because the human motion does not change significantly in so little time; the frame rate is set to $f_{rate} = 1/30$.

- Differently of the aforementioned velocities in the torso frame of reference, herein, relative to the sensor frame, for all joints, for each dimension individually, we computed the difference $\delta(b^t_{j_d}, b^{t-t_w}_{j_d})$ between the position at a given frame and the preceding 10^{th} frame. Using these values, we computed the velocities of the same joints for each dimension individually, $v_j = \frac{b^t_{j_d} - b^{t-t_w}_{j_d}}{f_{rate} \times t_w}$, obtaining additional 45 features.

- The angles variation of certain joints play a crucial role in carrying out many activities. We are interested in knowing whether a person is sitting or standing, so we compute the angles of both right and left elbows in the triangle formed by the hands, elbows and shoulders as well as the angles of the hip joints in the triangle formed by the shoulders, hips and knees and the angles of the knees in the triangles formed by the feet, knees and hips (Figure 3.5). The angle θ_i is given by:

$$\theta_i = \arccos\left(\frac{(\delta_{j_{12}})^2 + (\delta_{j_{23}})^2 - (\delta_{j_{13}})^2}{2 \times \delta_{j_{12}} \times \delta_{j_{23}}}\right), \tag{3.7}$$

  where $\delta_{j_{12}}$ is the distance between two joints, e.g. j_1 and j_2, that are forming a triangle in the skeleton. We have 2+2+2=6 features for angles, since we are considering the left and right side for the body joints. In addition, we compute the difference between these angles at a current frame and the preceding 10^{th} frame, $\theta_{v_i} = \theta^t_i - \theta^{t-10}_i$, obtaining additional 2+2+2=6 features.

Thus, in total, we attained a set with 206 spatio-temporal skeleton-based features, useful to discriminate different classes of activities.

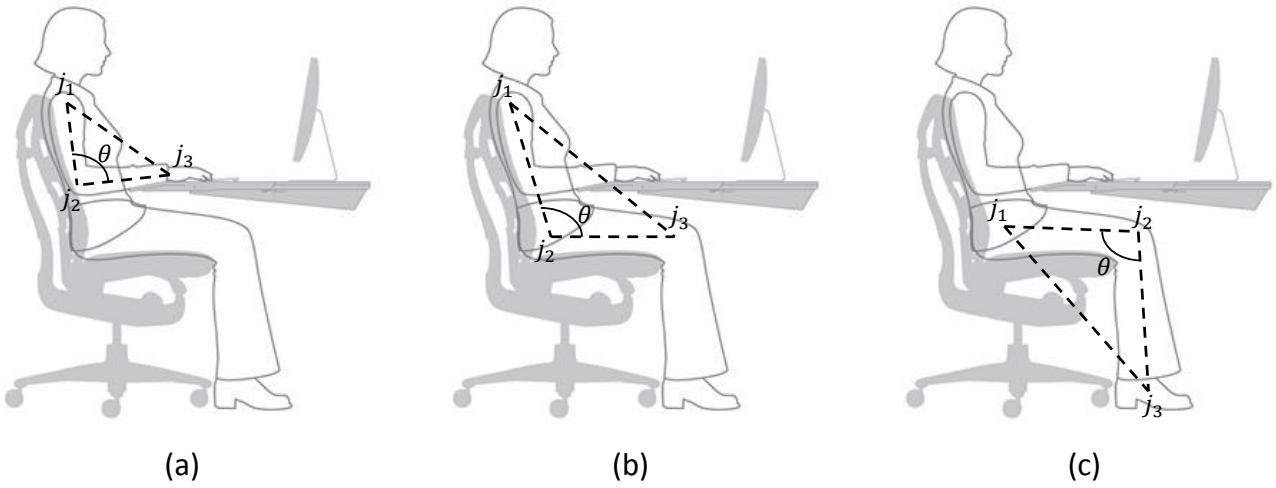


Figure 3.5: Angles used as features. (a) shows the angle of elbows in the triangle formed by the hands, elbows and shoulders. (b) shows the angle of the hip joints in the triangle formed by the shoulders, hips and knees. (c) shows the angle of the knees in the triangle formed by the feet, knees and hips.

3.2.1 Features pre-processing

Before using the features set in the classification module, a pre-processing step is applied. Normalization, standardization or filtering may be a requirement for many machine learning estimators, as they can behave badly if no pre-processing is applied to the features set. So, in the dataset case, we apply a moving average filter which smooths data by replacing each data point with the average of the neighbouring data points defined within the span. This smoothing process is given by the difference equation

$$y_s(i) = \frac{1}{2N+1}\left(y(i+N) + y(i+N-1) + \ldots + y(i-N)\right), \tag{3.8}$$

where $y_s(i)$ is the smoothed value for the ith data point, N is the number of neighbouring data points on either side of $y_s(i)$, and $2N+1$ is the span. In this work, $N = 5$ was determined empirically, giving a span of 11.

Subsequently, a normalization step is applied to the features set in such a way that the values of minimum and maximum obtained during the training stage were applied on the testing set as follows:

$$\mathbf{F}_{tr_i} = \frac{\mathbf{F}_{tr_i} - \min(\mathbf{F}_{tr})}{\max(\mathbf{F}_{tr}) - \min(\mathbf{F}_{tr})}, \quad \text{and} \quad \mathbf{F}_{te_i} = \frac{\mathbf{F}_{te_i} - \min(\mathbf{F}_{tr})}{\max(\mathbf{F}_{tr}) - \min(\mathbf{F}_{tr})}, \tag{3.9}$$

where $\mathbf{F}_{tr}$ is the set of features for training and $\mathbf{F}_{te}$ is the set of features for test; i is an index to describe a set of features in a specific frame; $\max(\cdot)$ and $\min(\cdot)$ are functions to get the global maximum and

minimum value of a feature set.

In the real-time case, we did not apply the moving average filter because it returns worse results. The normalization step is done in the same way as in the offline tests because we keep the maximum and minimum values of the training set.

3.3 Probabilistic Classification Framework

After features extraction, the next step is the classification. As already mentioned in chapter 1, a probabilistic ensemble of classifiers called DBMM is used. A detailed theoretical explanation of this method is done in section 2.1.1.4. Several classifiers can be used in the DBMM as base classifiers. In this work, a DBMM was designed using a NBC, a multi-class SVM classifier with a linear-kernel and a k-NN. The SVMs were trained according to a one-vs-one scheme, with the Cost parameter C set to 1.0 and classification outputs were given in terms of probability estimates. The k-NN was trained using 20 neighbours determined empirically, and classification outputs were given in terms of probability estimates as well.

Figure 3.6 describes the DBMM designed for this work. A training is previously done to define the initial weights and the likelihoods for each base classifier. For each frame, features are extracted and each base classifier returns the posterior probability for each activity. These posteriors are then used to update the weights and posteriorly to perform the DBMM fusion, as explained in section 2.1.1.4. The DBMM returns a new posterior for each activity which is used in the next frame as the prior probability for the fusion. After N frames, the activity for which the DBMM returned the highest posterior is assumed as the activity being performed.

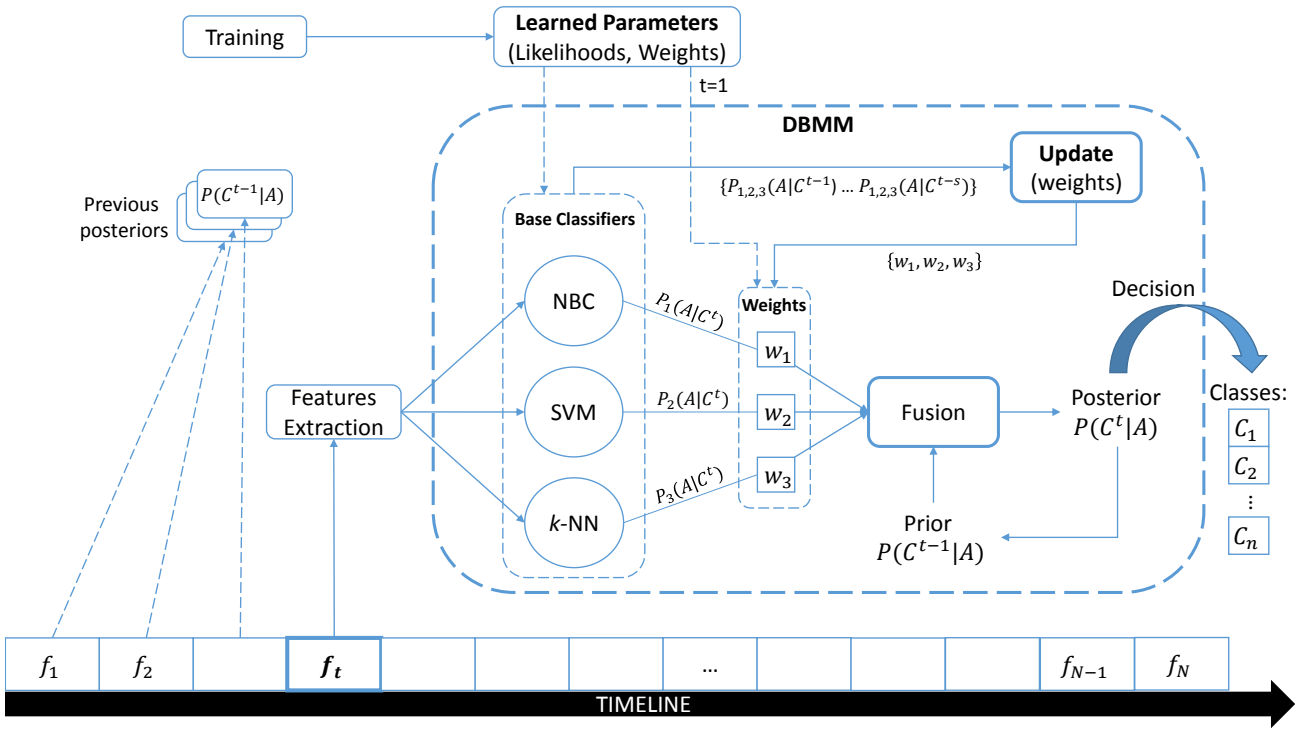


Figure 3.6: DBMM approach at current (time t) frame classification.

Chapter 4

Artificial Cognitive System Implemented in ROS

In this chapter an overview of ROS implementation of the developed artificial cognitive system is given and the hardware and drivers necessary to run this system are described. Details of the implementation in ROS of each module that makes up the system are given, such as navigation module, classification module and reaction module.

4.1 System Overview

The proposed artificial cognitive system was implemented in ROS, using a mobile robot as shown in Figure 4.1. The system comprises three main modules: one module in charge of autonomous navigation in an indoor environment, other module for recognizing the learned human activities from visual input, and the other in charge of triggering a reaction according to the activity detected.

In order to properly test the system in real scenarios, a mobile robot is used. Therefore, a personal robot endowed with cognitive skills, capable of monitoring the behaviours of a person should be able to autonomously navigate an indoor environment. The navigation module uses the odometry and laser scans from the robot to map the environment and be located on this map, randomly navigating, avoiding obstacles collision.

While the robot is navigating, Kinect is sending RGB-D data to the classification module. Once a skeleton is detected, the robot stops and the features extraction process starts. Then, classification is done using the DBMM and an activity is recognized. Once the system knows the activity being

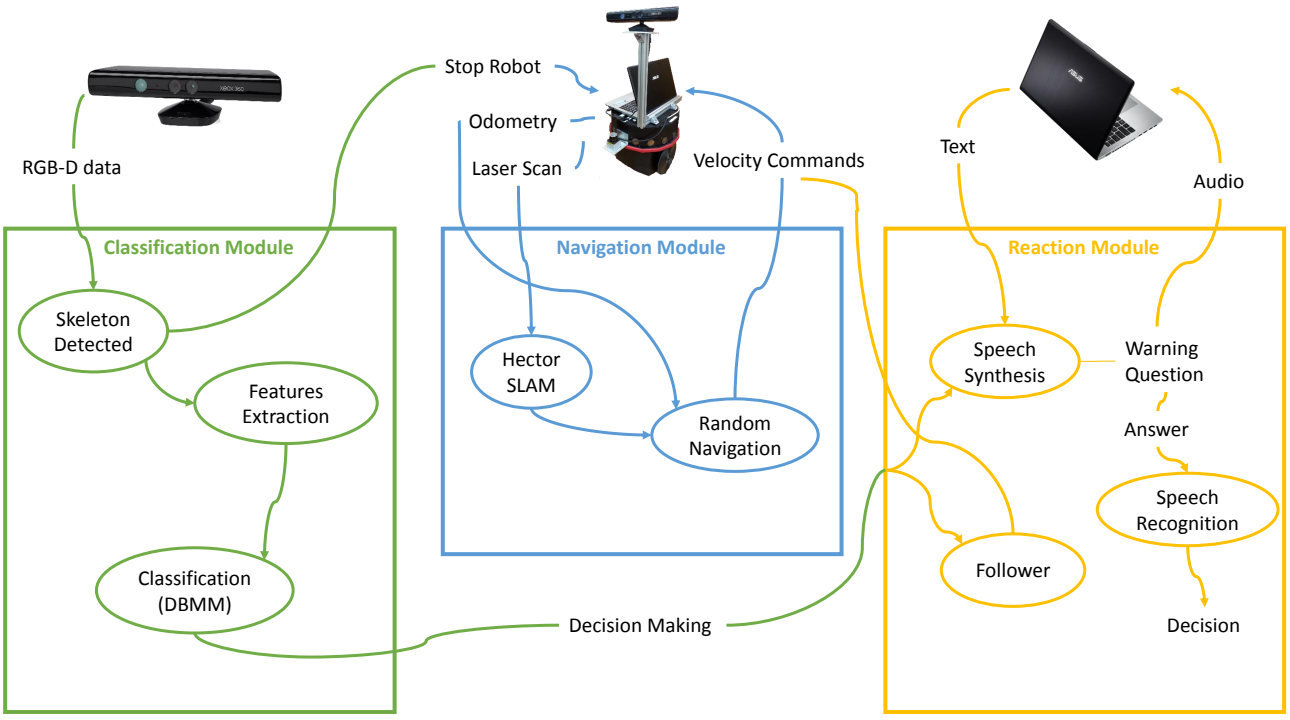


Figure 4.1: System overview.

performed, the reaction module is in charge of selecting what the robot should do next. Each activity has a predefined reaction associated in a lookup-table, including warnings, questions or changes in navigation.

4.2 Hardware and Drivers

The mobile platform used in this work is a Nomad Scout mobile robot (Figure 4.2). This mobile robot is equipped with a Hokuyo URG-04LX Laser Range Finder, a Raspberry Pi to control the platform motors and encoders used for odometry. An elevated structure was built to support the Microsoft Kinect, and an Asus laptop is used, mounted on top of the robot.

ROS already provides a large set of software to develop robotic applications. In this case, the OpenNi's driver is used to acquire all the data from the Kinect. In order to make the bridge between the hardware of the robot and ROS, a robot driver, implemented in [7] was used. This driver allows the robot to send odometry information to ROS and ROS to send velocity commands to the wheels of the robot.

Figure 4.2: The robotic setup used for experiments. It is possible to see the Nomad Scout robot with the elevated structure and Kinect on top of it. On top of the robot is the Asus laptop that was used to control the platform and in the small silver support in the middle of the robot is the Hokuyo URG-04LX Laser Range Finder.

4.3 Navigation Module

The navigation module (Figure 4.3) comprises three main ROS nodes: the *move_base* node, the *isr_hector_mapping* node and the *simple_navigation_goals* node.

The *move_base* node consists of a global planner that produces global trajectories between two points in the world map, and a local planner to follow that path in the most optimal manner. The global planner uses a global costmap built dynamically from received occupancy grid map, every time the map is updated. Based on this costmap, the global planner performs a tree search to find the optimal path. The local planner uses a local costmap which is also built dynamically with raw data available from range sensors, in this case a laser range finder. The obstacles are updated and a dynamic window approach is implemented to collision avoidance [13], having the global path as reference. The *move_base* node subscribes the */odom* topic for the local planner and the */tf* topic with the sensor transforms. It also subscribes the */move_base_simple/goal* topic to get the destination coordinates and generate the best path to reach that point. In order to execute the planned path, this node sends a stream

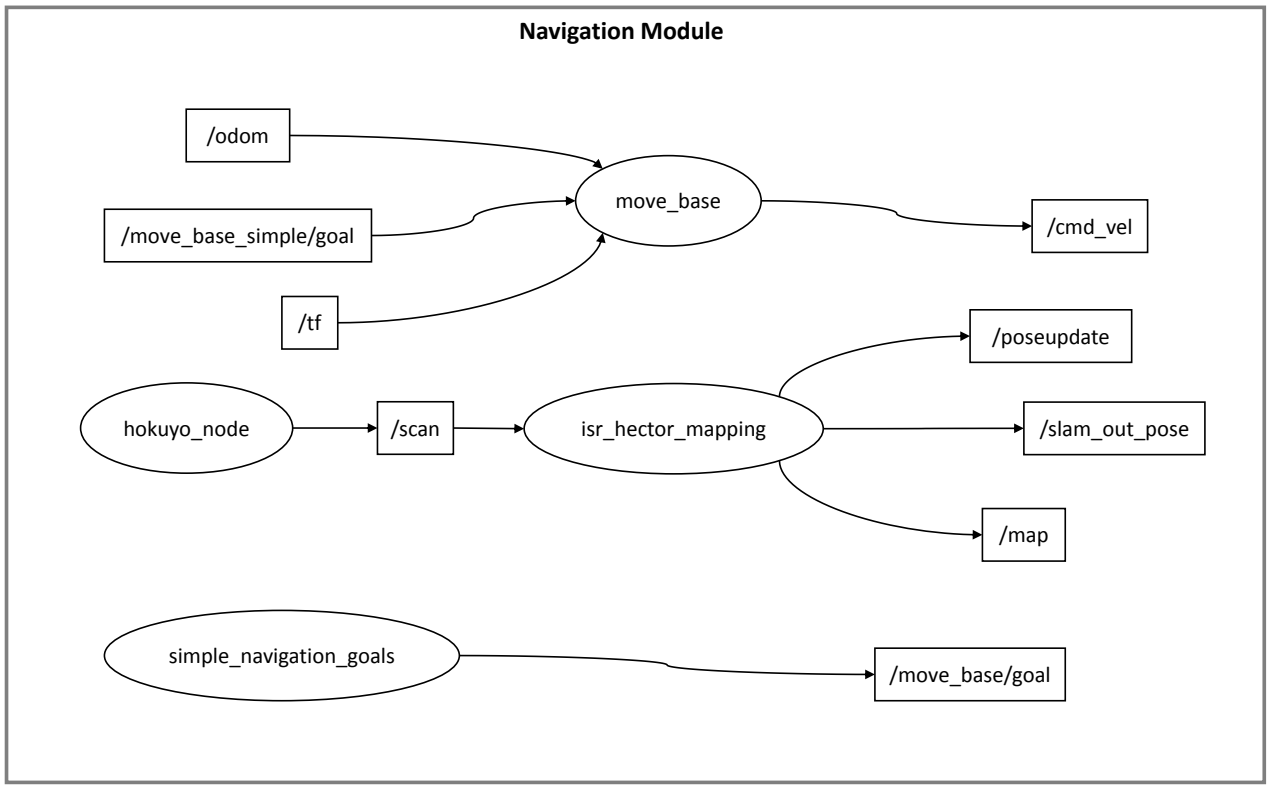


Figure 4.3: Navigation module implementation. Nodes represented on ellipses and topics on rectangles.

of velocity commands through the */cmd_vel* topic to the wheels of a mobile robot.

The *isr_hector_mapping* SLAM node estimates the current pose of the robot and updates the map of the environment. This node was modified in [7], based on Hector Team SLAM's method, which was modified to receive an initial map of the static environment. In this work, we do not use an a priori map of the environment, but one can be used in the future, if it is available. This node subscribes to the */scan* topic to get the readings of the laser used by the SLAM system. Then, it gets the map data from the */map* topic, which is latched, and updated periodically. It publishes the estimated robot pose with or without a Gaussian estimate of uncertainty using the */poseupdate* and */slam_out_pose* topics, respectively.

The *simple_navigation_goals* node was created to the robot randomly navigate until a person is detected. This node publish to the */move_base/goal* topic random goals, i.e., x and y coordinates and angle *yaw* for robot orientation, every five seconds. Once the skeleton is detected, the robot stops and starts the activity recognition performed by the classification module.

4.4 Classification Module

In order to link the Kinect with ROS, the OpenNi's driver is used. The PrimeSense NiTE 2.0 middleware library made possible to track the human skeleton without being necessary any kind of calibration or starting pose (e.g. Psi pose). In figure 4.4, the *openni_tracker* node is responsible for detecting the human skeleton and track it while it is within the range of the Kinect sensor. This node broadcasts 15 skeleton joints using the */tf* topic. The *tf_listener* node is continuously listening for new *tf* frames and as soon as the skeleton frames are detected, it immediately gets theirs x,y,z coordinates. The 3D coordinates are provided in the sensor frame of reference, however, a transformation is applied in order to have the coordinates in the skeleton frame of reference. Both information is kept and saved in two different text files. The *classifica* node reads five seconds of data from both text files and compute the features described in section 3.2. This node was written in Python in order to use the open source machine learning library scikit-learn [25]. This library has many simple and efficient tools that can be used for classification. A NBC and a multi-class SVM classifier with a linear-kernel were previously trained and the train data was saved. The node uses this data and the features extracted to obtain the individual classification from each base classifier. Then, the DBMM combines the individual classifications in order to recognize the activity being performed.

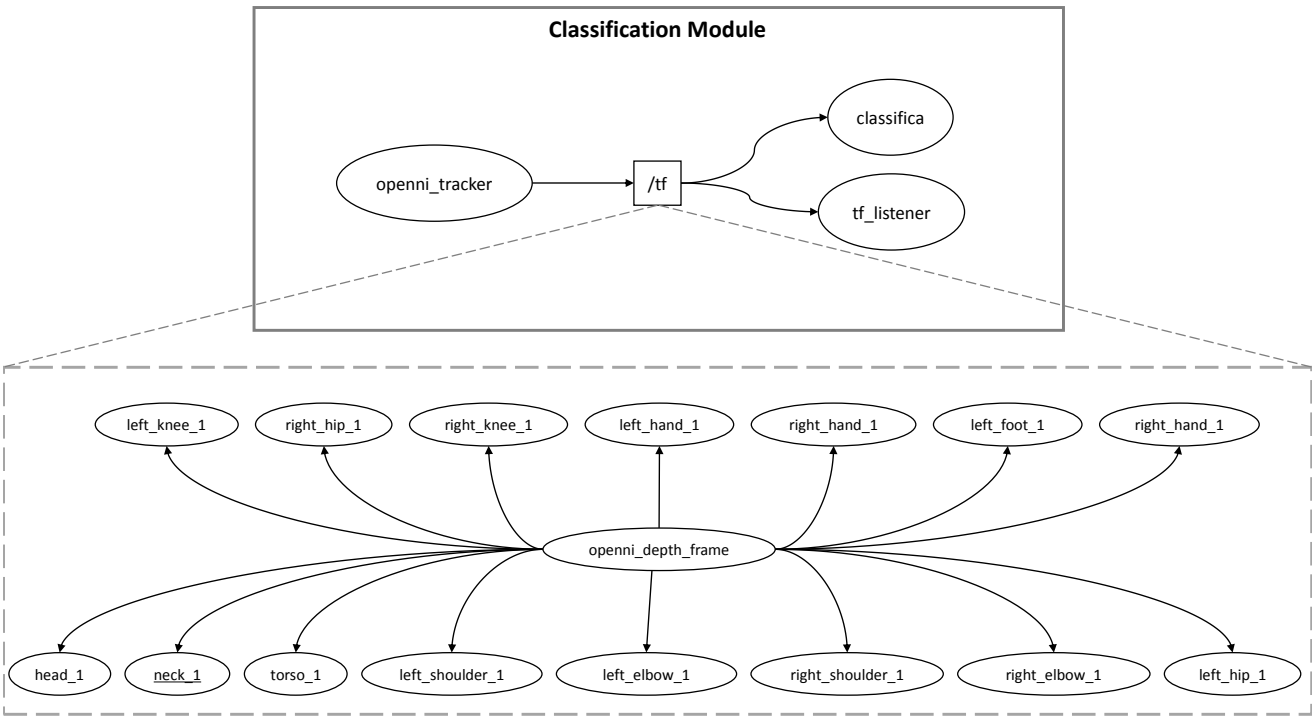


Figure 4.4: Classification module nodes and topics. In the dashed rectangle are presented the *tf* frames for each skeleton joint and the *tf* frame *openni_depth_frame* as the sensor frame of reference.

4.5 Reaction Module

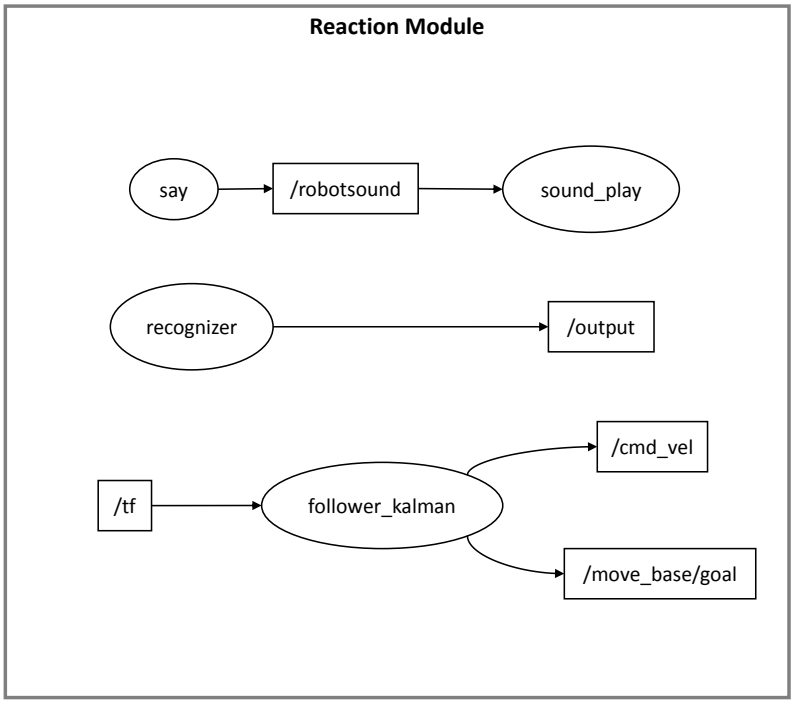


Figure 4.5: Reaction module nodes and topics.

After recognizing the activity being performed, the system should be able to make a decision, using for this purpose the reaction module (Figure 4.5). Once an activity is recognized by the classification framework, the proper reaction is taken from the look-up table to be executed by the mobile robot.

The reaction of the robot depends on what activity is detected (Figure 4.6). In the event of the person telling the robot to follow him/her, the *follower_kalman* node is executed. The robot will follow the person, keeping a safety distance of 2.5 meters and an orientation margin of ± 0.2 radians. In order to do that, the node will get the torso coordinates using the *tf* topic and computes the distance remaining to be 2.5 meters away. By keeping the mentioned distance, we ensure that the person remains within the range of Kinect and the robot do not exceed the social space, defined in 1966 by Hall [16] as a radius between 1.2m and 3.6m (4-12 feet). This node can use the */cmd_vel* topic or the */move_base_goal* topic. In the first case, if a positive value is obtained, the robot will move that distance. If the value is negative, the robot will move backwards. Using the */cmd_vel* topic the robot is not aware of what surrounds it, so this method is not collision free. On the other hand, using the */move_base_goal* topic, the robot takes into account the obstacles around it. As there is no sensor in the rear of the robot, if the distance is less than 2.5 meters the robot stands still, orientating itself with the person, turning on itself. The advantage

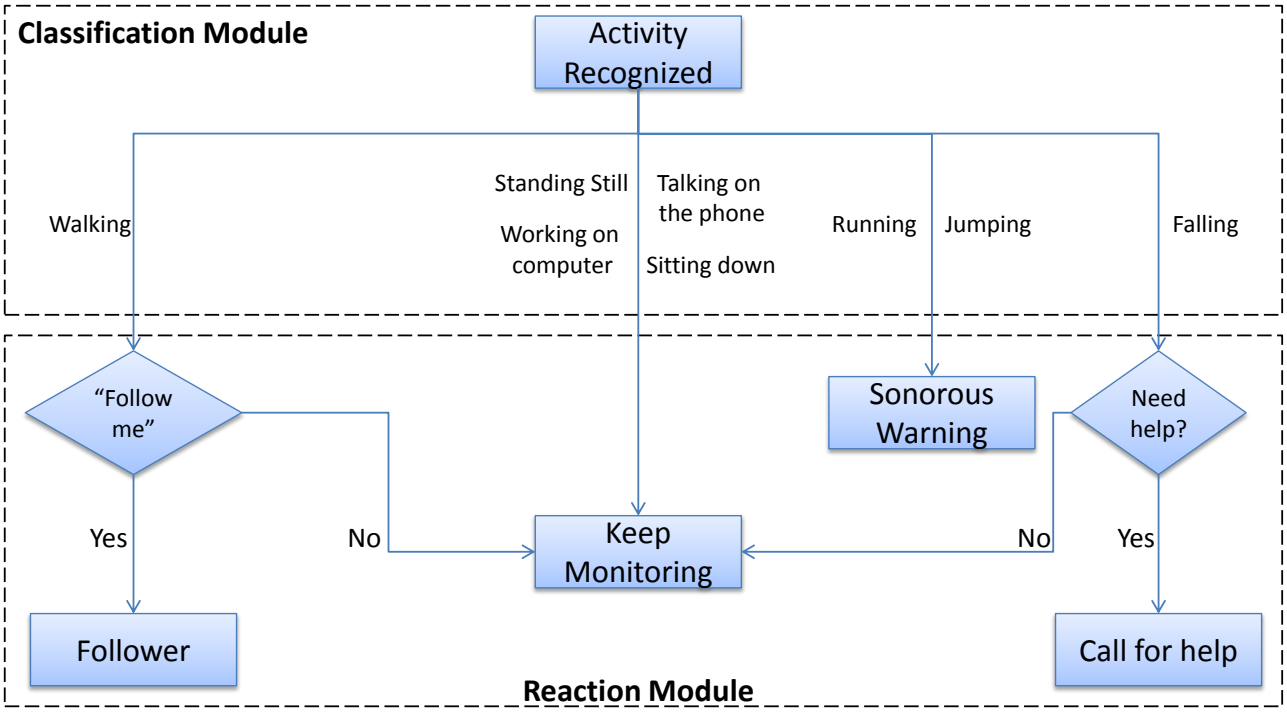


Figure 4.6: Decision tree in reaction module.

in using the */cmd_vel* topic is that we have total control of the robot velocity and the robot performs a smoother motion when compared with the one using the */move_base_goal* topic.

A Kalman filter is used to estimate the trajectory of the person one second ahead in order to avoid collision with the robot. If a collision trajectory is detected, the robot should step away, in order to the person walk through safely. For the prediction of the human motion, a position model was adopted, where the state includes position $(x(k);y(k))$ of the human target:

$$\begin{cases} x(k) = x(k-1) + v_x(k-1) \times \Delta t \\ y(k) = y(k-1) + v_y(k-1) \times \Delta t \end{cases} \tag{4.1}$$

with $\Delta t = t(k) - t(k-1)$.

The a priori estimate of the Kalman filter is given by:

$$x_e(k) = \phi(k-1)x_e(k-1), \tag{4.2}$$

where ϕ is the state transition model, which in this work is the identity. The error covariance matrix P for the new x_e is given by:

$$P(k) = \phi(k-1)P(k-1)\phi(k-1)^T. \tag{4.3}$$

Then, a measure $y(k)$ is made and the new estimate is given by:

$$x_e(k) = x_e(k) + K(k-1)(y(k) - x_e(k)), \tag{4.4}$$

where K is the Kalman filter gain. This gain is computed as follows:

$$K(k) = P(k)(P(k) + R(k))^{-1}. \tag{4.5}$$

where R is the error covariance matrix of the sensor. The new x_e has an error covariance matrix given by:

$$P(k) = (I - K(k)) \times P(k) \times (I - K(k))^T + K(k)R(k)K(k)^T. \tag{4.6}$$

Using the torso coordinates as measures, it is possible to compute the x velocity v_x and y velocity v_y as follows:

$$\begin{cases} v_x(k) = (x_{torso}(k) - x_{torso}(k-1)) \times \Delta t \\ v_y(k) = (y_{torso}(k) - y_{torso}(k-1)) \times \Delta t \end{cases} \tag{4.7}$$

The node estimates the position of the person one second ahead and if the estimate distance between person and robot is less than one meter, the robot will execute a pre-defined maneuver in order to avoid collision with the person that is coming in the robot direction. In order to perform that maneuver, velocities are sent to the robot wheels using the */cmd_vel* topic.

If the activity performed is running or jumping, the robot will warn the person that it is not allowed to behave like that. Using the voice synthesizer package *sound_play* from ROS, it is possible to convert text into audio. The *say* node takes text as input and publishes the data through the */robotsound* topic. The *sound_play* node subscribes the same topic and transforms the data into audio, using the computer speakers.

If the activity detected is falling on the floor, the robot should be able to assume that a risk situation has come up. The robot will ask if the person needs some help, using the same *sound_play* node as before. In order to recognize the person's answer the *pocketsphinx* package is used. This package

recognizes a single word or a stream of words from a vocabulary file previously created. In this case, the vocabulary comprises the following words: ”no”, ”yes”, ”please”, ”help”, ”follow”, ”me”. The package can recognize combinations of these words, such as ”please help me”. The *recognizer* node receives audio from a microphone as input and translate it into text publishing to the */output* topic. If the robot gets a positive answer (e.g. ”yes”), then, it will call a doctor or a relative; otherwise, with a negative reply the robot will keep monitoring.

Finally, if the activity recognized is standing still, working on computer, talking on the phone or sitting down, the robot will keep monitoring the person, since these activities do not show any risk situation.

Chapter 5

Experimental Results

In this chapter, the performance of the proposed system is assessed and validated. First, the activity recognition framework is validated offline, using the collected dataset and a public available benchmark dataset. Then, several experiments are carried out on-the-fly in order to test and validate the integrated system developed.

5.1 Performance on datasets

5.1.1 Performance on original dataset

Before testing the proposed classification framework on-the-fly, using a mobile robot, experiments were also done offline, using the collected dataset described in 3.1.2.

The validation technique adopted for assessing the results was the leave-one-out cross-validation (LOOCV). The idea is to verify the capacity of generalization of the classifier by using the strategy of ”new person”, i.e., learning from different persons and testing with an unseen person. As this dataset comprises four subjects, four tests are performed. The classification results are presented in a confusion matrix and with the performance measures of Accuracy, Precision, Recall of the overall of the four tests. Figure 5.1 shows the results in a single confusion matrix. Table 5.1 shows the performance in terms of Precision (Prec) and Recall (Rec) of this approach for each activity. The results show that using DBMM with the proposed features, improvements in the classification were obtained in comparison with using the base classifiers alone. The overall results attained were: accuracy 93.41%, precision 93.61% and recall 92.25%. For comparison purposes, Table 5.2 summarizes the results from single classifiers and

an average ensemble compared with DBMM, showing the improvement achieved using the described skeleton-based features in 3.2.

Table 5.1: Performance on the dataset ("new person"). Results are reported in terms of Precision (Prec) and Recall (Rec).

	DBMM	
Activity	Prec	Rec
walking	89.63%	99.73%
standing still	94.86%	98.13%
working on computer	95.93%	93.20%
talking on the phone	93.64%	87.96%
running	92.81%	85.20%
jumping	92.52%	88.83%
falling down	97.24%	90.04%
sitting down	92.27%	94.88%
Average	93.61%	92.25%

Table 5.2: Global results using single classifiers, a simple average ensemble (AV) and the DBMM.

Method	Acc.	Prec.	Rec.
NBC	82.90%	85.79%	82.67%
SVM	88.47%	89.02%	87.62%
k-NN	87.98%	90.09%	87.06%
AV	85.29%	87.74%	84.68%
DBMM	**93.41**%	**93.61**%	**92.25**%

5.1.2 Performance on UTKinect Action Dataset

In order to confirm the effectiveness of the classification framework, the proposed method was also evaluated on a second dataset: UTKinect [37]. This dataset contains 10 types of human actions in an indoor environment: *walk, sit down, stand up, pick up, carry, throw, push, pull, wave, clap hands*. Each action was performed by 10 different persons for 2 times: 9 males and 1 female.

This dataset presents some challenging differences when compared with the dataset collected for this work. First, the durations of the action clips vary dramatically. The length of sample actions ranges from 5 to 120 frames which can lead to lack of information for some actions. Second, another difficulty is added by the presence of occlusions, caused by human-object interaction or by the absence of some body parts in the field of view.

We use the features described in section 3.2, however some modifications were necessary. The skeleton model used in UTKinect dataset has 20 joints, so the features are computed using 20 joints instead of 15, totalling 322 features instead of 206. Since the smaller sample action has only 5 frames,

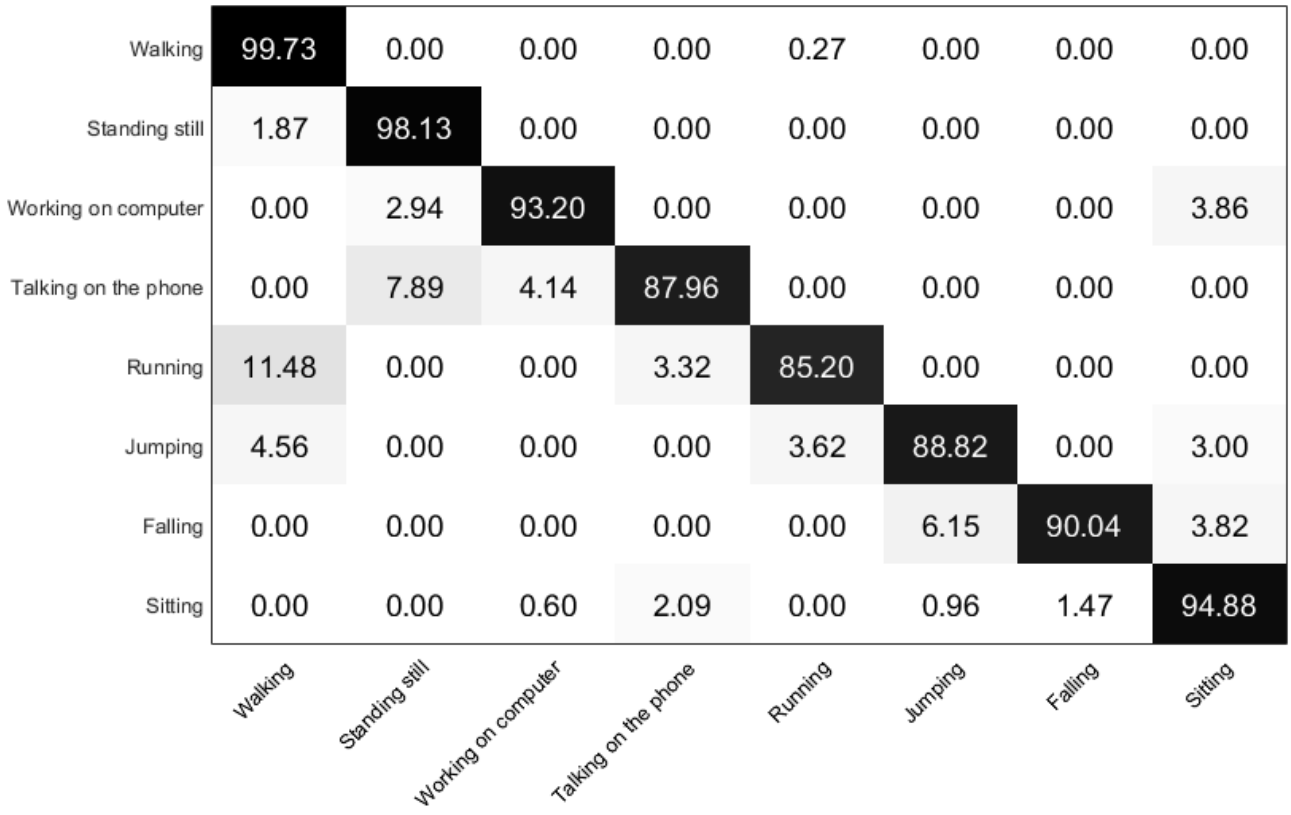


Figure 5.1: Confusion matrix obtained from the DBMM classification applied on the dataset

it makes no sense to use a temporal window of 10 frames. Because of that, for this dataset, a temporal window of 1 frame is used.

To compare the results attained with other state of the art works, the experiment protocol proposed in [37] is used. Since there are 10 subjects performing each activity two times, 20 tests were performed using LOOCV. The results attained are presented in a confusion matrix in Figure 5.2. Table 5.3 shows the overall accuracy of DBMM (91.29%) compared with some selected works of the state-of-the-art. Our results outperforms the ones attained by some other works, including the authors of the dataset [37]. Reminding that the focus of this thesis is a real-time application, some works have better results than ours in this dataset, because they processes more features. In order to outperform the two works referred in table 5.3, more features should be added to the bag of features. Some examples of features to improve the classification performance on this dataset, and to outperform all other works, is the energy model of the autocorrelation applied over the difference of two consecutive poses of the 3D skeleton computed from the joint coordinates as shown in [10]. However, since the focus of this work is a real time application, we do not attempt to use other features beyond of the proposed ones in this work in order to keep an acceptable processing time. Even so, with the proposed features we obtained competitive results.

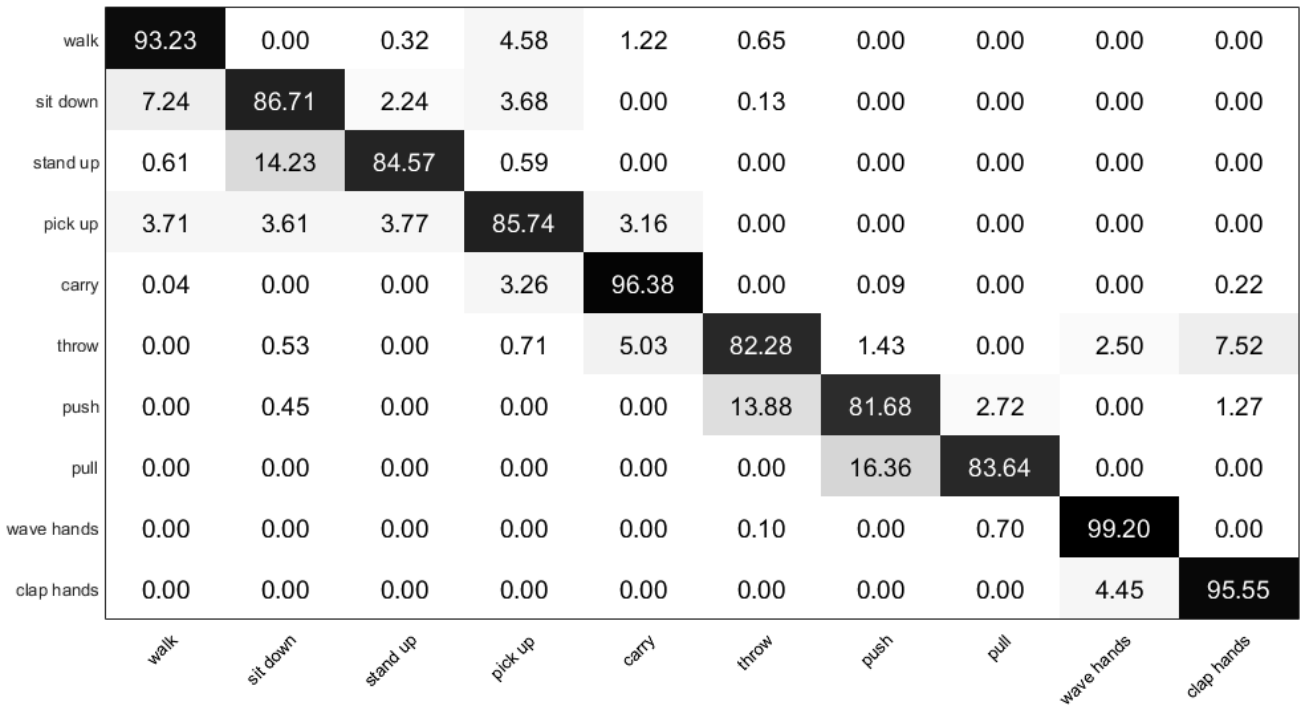


Figure 5.2: Confusion matrix obtained from the DBMM classification applied on UTKinect dataset.

Table 5.3: Comparison of approaches that use the UTKinect dataset in terms of overall accuracy. Columns 3 an 4 point out the feature types used by the approaches.

Method	Acc	SK joints	Depth
Vemulapalli *et al.* [34]	97.08%	X	
Slama *et al.* [30]	95.25%		X
Proposed framework (DBMM)	**91.29**%	X	
Xia *et al.* (dataset authors) [37]	90.92%	X	
Slama *et al.* [31]	88.50%	X	

5.2 Performance on-the-fly using a mobile robot

The experimental tests using the proposed approach for a real time application are a little bit different than the experimental tests on the dataset. In this case, the robot will acquire 5 seconds of RGB-D sensor data for features extraction and classification. Only the NBC and SVM were used as base classifiers for the DBMM fusion, because they are enough for obtaining good results, thus, avoiding spending more processing time using other base classifiers. After 5 seconds of frames classification, a final decision is made for activity recognition and to trigger a proper robot reaction. The proposed framework is capable of recognizing different activities transitions that happens sequentially in case of a person transit from one activity to another one, e.g., a person that is standing and sequentially pass to a sitting down position and consequentially working on the computer.

Figure 5.3: Scenarios for experimental tests. Entrance of the ISR (left) and ISR shared experimental areas (right).

Three tests were carried out for each activity with three different subjects. One of the subjects was already ”seen” in the training, while the rest are ”unseen” subjects. The subjects were divided in two scenarios shown in Figure 5.3. One subject performed the activities at the entrance of the Institute of Systems and Robotics (ISR) and the other two in a shared room intended for experimental tests, also at the ISR. Figure 5.4 shows some examples of tests of daily activities and unusual or risk situations that the mobile robot correctly recognized.

All activities were classified with a large percentage of certainty, so that the overall performance of classification is shown in Figure 5.5. The overall results attained in real-time experiments were: accuracy 90.55%, precision 90.84% and recall 90.55%.

Table 5.4 shows the results in terms of recall of each test for each subject. Looking at the results attained, it is possible to conclude, as expected, that the best performance is achieved for the ”seen” person (subject 1). However, the difference of results between subjects is not very significant in most cases, which indicates that the fact of being or not a ”seen” person is not a key factor for the performance of the classification. The most important factor in a real-time application is that in the end, the activity being performed is correctly recognized.

Since the robot correctly classified the activity performed, it also successfully reacted accordingly to the situation. Figure 5.6 shows a sequence of events from an activity that is being recognized (in this case falling) to react according to this activity. First, the skeleton of a person is detected and tracked, initiating the monitoring stage. Then, the person falls on the floor and the robot correctly recognize the risk situation ”falling”. Detecting such a behaviour, the robot asks if the person needs help. The robot

receives an affirmative answer from the person and immediately calls for help.

Table 5.4: On-the-fly results in terms of recall for 3 different subjects. One subject seen and two unseen.

		Activity								Overall
	Test	walking	standing still	working on computer	talking on the phone	running	jumping	falling down	sitting down	
Subject 1 (seen)	1	96.30	100	100	59.26	85.19	85.19	85.19	96.30	88.43
	2	96.30	100	100	100	85.19	88.89	95.45	96.30	95.27
	3	92.59	100	92.59	100	85.19	88.89	92.86	96.30	93.55
	Average	95.06	100	97.53	86.42	85.19	87.65	91.17	96.30	92.42
Subject 2 (unseen)	1	66.67	100	96.30	100	96.30	81.48	74.07	70.37	85.65
	2	81.48	85.19	96.30	92.59	85.19	92.59	74.07	92.59	87.50
	3	81.48	100	88.89	100	85.19	92.59	95.45	92.59	92-02
	Average	76.54	95.06	93.83	97.53	88.89	88.89	81.20	85.18	88.39
Subject 3 (unseen)	1	82.14	96.30	100	100	73.33	96.30	85.19	88.89	90.27
	2	92.86	96.30	100	100	80.00	92.60	100	85.19	93.37
	3	82.14	92.59	100	100	73.33	96.30	81.48	85.19	88.88
	Average	85.71	95.06	100	100	75.55	95.07	88.89	86.42	90.84
Overall Average		85.77	96.71	97.12	94.65	83.21	90.54	87.09	89.30	90.55

Figure 5.4: Shots of tests of activity recognition (‘unseen” person) using a mobile robot.

The on-the-fly tests were performed using an Asus laptop with an Intel i7-4700HQ 2.4GHz CPU and 16Gb of RAM, running Ubuntu 12.04 and ROS Hydro. The average computation time, since features extraction to the classification or reaction, if applicable, was 0,1435 seconds. This low computation time demonstrates that the proposed system is computationally efficient.

Regarding the following task with collision avoidance with person, rigorous tests were not carried out, reminding that this is an additional ability provided to the robot. However, whenever the robot was performing this task, it was successful, with few exception, namely perform the collision avoidance maneuver too soon. In the future, this ability should be evaluated in a proper way and improved if necessary.

	Walking	Standing still	Working on computer	Talking on the phone	Running	Jumping	Falling	Sitting
Walking	85.77	3.25	0.00	0.82	4.48	3.29	1.19	1.19
Standing still	0.41	96.71	0.00	1.64	0.00	1.23	0.00	0.00
Working on computer	0.41	0.00	97.12	0.00	0.82	0.00	1.65	0.00
Talking on the phone	0.82	2.47	0.00	94.65	0.41	1.65	0.00	0.00
Running	5.27	2.88	2.22	1.98	83.21	1.48	2.96	0.00
Jumping	0.00	3.70	0.00	4.11	0.00	90.54	0.00	1.65
Falling	2.47	5.52	0.41	1.23	1.23	1.23	87.09	0.81
Sitting	0.00	4.12	5.35	0.00	0.00	1.23	0.00	89.30

Figure 5.5: DBMM on-the-fly classification confidence (average) presented in a confusion matrix.

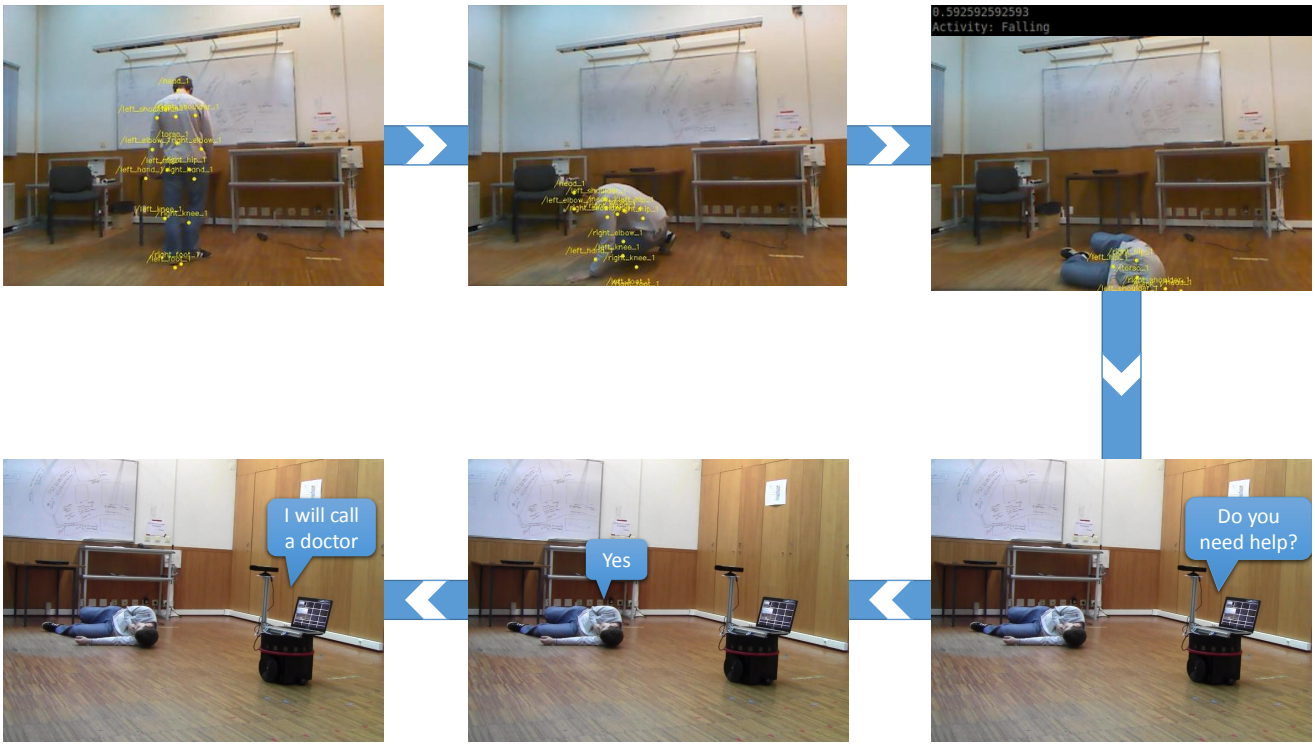


Figure 5.6: Sequence of events on detecting a person falling and reacting.

Chapter 6

Conclusions and Future Work

Human activity recognition is a fundamental step in understanding the human behaviour in several scenarios. In this work, a fully integrated robotic application was developed in order to recognize human activities in real scenarios. A dynamic probabilistic ensemble of classifiers (DBMM) was implemented for daily activity recognition using a proposed spatio-temporal 3D skeleton-based features. This approach was tested in datasets and on-the-fly, bearing in mind an assisted-living application.

Analysing the experimental results in datasets, it is possible to conclude that DBMM truly improves the classification performance, corroborating the conclusions in [10]. DBMM results outperforms other single classifiers in terms of overall accuracy, precision and recall measures.

The robotic application was developed in ROS and comprises a navigation module, a classification module and a reaction module. In the on-the-fly tests, the robot was able to recognize the activity being performed with great confidence, proving that DBMM achieves good results online. Once recognized the activity, the robot was able to make a decision according to the situation, assisting a person if needed, showing that the proposed framework has good potential for robot-assisted living.

Regarding the goals initially proposed in this work, the main contributions are:

- Extending the use of DBMM to real-time applications using proposed discriminative 3D skeleton-based features, which can successfully characterize different daily activities;
- Combining different ROS modules running in parallel towards a real time robot-assisted living application;
- Assessment and validation: (i) leave-one-out cross validation of the activity recognition using

our dataset; (ii) comparison of different classification models using the proposed features (NBC, SVM and *k*-NN); (iii) triggering of robot (re)actions given a recognized activity.

In spite of the good results, there is always room for improvement, so further work should be developed in order to achieve a proper companion robot. The navigation module should be refined so that the robot navigates the environment in a more efficient way, instead of randomly.

More activities and reactions could be added as well, to cover more possible situations. Additional contextual information should be added, such as ”who”, ”where”, ”when” in order to fully understand human behaviours. The same activity may have different behaviour interpretations depending on the context in which it is performed (e.g. where it is performed).

Finally, in terms of hardware, a better equipped robot can assist a human in a more effective way. A simple robotic arm brings a lot of possibilities with regard to assist elderly or disabled humans.

Bibliography

[1] Svm, Website: http://i.imgur.com/wuxyo.png, accessed on 16/08/2015.

[2] Website: https://msdn.microsoft.com/en-us/library/jj131033.aspx/, accessed on 16/08/2015.

[3] Website: https://msdn.microsoft.com/en-us/library/microsoft.kinect.depthimageformat.aspx/, accessed on 16/08/2015.

[4] Vincent Arsigny, Pierre Fillard, Xavier Pennec, and Nicholas Ayache 5. Log-Euclidean metrics for fast and simple calculus on diffusion tensors. *Magnetic Resonance in Medicine, 56(2):411–421.*, 2006.

[5] Chen Chen, Kui Liu, and Nasser Kehtarnavaz. Real-time human action recognition based on depth motion maps. *Journal of Real-Time Image Processing*, 2013.

[6] V. Cortes, C.; Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[7] Jorge da Silva Perdigão. Collaborative-control based navigation of mobile human-centered robots. Master's thesis, Institute of Systems and Robotics, University of Coimbra, 2014.

[8] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR'05*, 2005.

[9] Diego R. Faria, Cristiano Premebida, and Urbano Nunes. A probalistic approach for human everyday activities recognition using body motion from RGB-D images. In *IEEE RO-MAN'14, * Kazuo Tanie Award Finalist*, 2014.

[10] Diego R. Faria, Mario Vieira, Cristiano Premebida, and Urbano Nunes. Probabilistic human daily activity recognition towards robot-assisted living. In *IEEE RO-MAN'15: IEEE International Symposium on Robot and Human Interactive Communication. Kobe, Japan, August*, 2015.

[11] Tom Fawcett. An introduction to roc analysis. *Pattern Recogn. Lett.*, 27(8):861–874, June 2006.

[12] E. Fix and J.L Hodges. Discriminatory analysis, non-parametric discrimination: consistency properties. Technical report, USAF Scholl of aviation and medicine, Randolph Field, 1951. 4.

[13] Dieter Fox, Wolfram Burgard, and Sebastian Thrun. The dynamic window approach to collision avoidance. *IEEE Robot. Automat. Mag.*, 4(1):23–33, 1997.

[14] Jean-Luc Gauvain and Chin-Hui Lee. Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains. *Speech and Audio Processing, IEEE Transactions on*, pages 291–298, 1994.

[15] Kai Guo. *Action Recognition using Log-Covariance Matrices of Silhouette and Optical-Flow Features*. PhD thesis, Boston University, College of Engineering, 2012.

[16] E.T. Hall. *The hidden dimension*. Doubleday Anchor Books. Doubleday, 1966.

[17] Chih-Wei Hsu and Chih-Jen Lin. A comparison of methods for multiclass support vector machines. *Trans. Neur. Netw.*, 13(2):415–425, March 2002.

[18] Kourosh Khoshelham and Sander Oude Elberink. Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors*, 12(2):1437, 2012.

[19] Hema S Koppula, Rudhir Gupta, and Ashutosh Saxena. Learning human activities and object affordances from RGB-D videos. In *IJRR journal*, 2012.

[20] Weiyao Lin, Ming-Ting Sun, Radha Poovendran, and Zhengyou Zhang. Human activity recognition for video surveillance. In *ISCAS*, pages 2737–2740. IEEE, 2008.

[21] P. Menezes, L. Brethes, F. Lerasle, P. Danes, and J. Dias. Visual tracking of silhouettes for human-robot interaction. In *In Proceeding of International Conference on Advanced Robotics (ICAR01)*, pages 971–976, 2003.

[22] Matteo Munaro and Emanuele Menegatti. Fast RGB-D People Tracking for Service Robots. *Autonomous Robots*, 2014.

[23] United Nations. *World Population Ageing, 2013*. Economic & social affairs. UNITED NATIONS PUBN, 2014.

[24] Wei Niu, Jiao Long, Dan Han, and Yuan-Fang Wang. Human activity detection and recognition for video surveillance. In *Multimedia and Expo, 2004. ICME '04. 2004 IEEE International Conference on*, pages 719–722, 2004.

[25] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[26] Lasitha Piyathilaka and Sarath Kodagoda. Gaussian mixture based HMM for human daily activity recognition using 3d skeleton features. In *IEEE 8th Conf. on Ind. Electronics and App.*, 2013.

[27] R.M. Royall. *A class of non-parametric estimates of a smooth regression function*. Dept. of Statistics, 1966.

[28] Paul Scovanner, Saad Ali, and Mubarak Shah. A 3-Dimensional sift descriptor and its application to action recognition. In *MULTIMEDIA '07 Proceedings of the 15th international conference on Multimedia*, pages 357–360, 2007.

[29] Samsu Sempena, Nur Ulfa Maulidevi, and Peb Ruswono Aryan. Human action recognition using dynamic time warping. In *ICEEI*, pages 1–5. IEEE, 2011.

[30] Rim Slama, Hazem Wannous, and Mohamed Daoudi. Grassmannian representation of motion depth for 3D human gesture and action recognition. In *22nd International Conference on Pattern Recognition, ICPR 2014, Stockholm, Sweden, August 24-28, 2014*, pages 3499–3504, 2014.

[31] Rim Slama, Hazem Wannous, Mohamed Daoudi, and Anuj Srivastava. Accurate 3D action recognition using learning on the Grassmann manifold. *Pattern Recognition*, 48(2):556–567, 2015.

[32] Jaeyong Sung, Colin Ponce, Bart Selman, and Ashutosh Saxena. Unstructured human activity detection from RGBD images. In *ICRA'12*, 2012.

[33] Robert J. Vanderbei. LOQO: an interior point code for quadratic programming. *Optimization Methods and Software*, 11(1-4):451–484, 1999.

[34] Raviteja Vemulapalli, Felipe Arrate, and Rama Chellappa. Human action recognition by representing 3D skeletons as points in a lie group. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[35] Michael Volkhardt, Steffen Mueller, Christof Schroeter, and Horst-Michael Gross. Real-time activity recognition on a mobile companion robot. In *55th Int. Scientific Colloquium*, 2010.

[36] L. Xia, C.-C. Chen, and J. K. Aggarwal. Human detection using depth information by kinect. In *Workshop on Human Activity Understanding from 3D Data in conjunction with CVPR (HAU3D)*, 2011.

[37] Lu Xia, Chia-Chih Chen, and J. K. Aggarwal. View invariant human action recognition using histograms of 3D joints. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 20–27, 2012.

Appendix A

Paper Accepted and Presented at the IEEE RO-MAN 2015 Conference

Probabilistic Human Daily Activity Recognition towards Robot-assisted Living

Diego R. Faria, Mario Vieira, Cristiano Premebida and Urbano Nunes

Abstract— **In this work, we present a human-centered robot application in the scope of daily activity recognition towards robot-assisted living. Our approach consists of a probabilistic ensemble of classifiers as a dynamic mixture model considering the Bayesian probability, where each base classifier contributes to the inference in proportion to its posterior belief. The classification model relies on the confidence obtained from an uncertainty measure that assigns a weight for each base classifier to counterbalance the joint posterior probability. Spatio-temporal 3D skeleton-based features extracted from RGB-D sensor data are modeled in order to characterize daily activities, including risk situations (e.g.: falling down, running or jumping in a room). To assess our proposed approach, challenging public datasets such as MSR-Action3D and MSR-Activity3D [1] [2] were used to compare the results with other recent methods. Reported results show that our proposed approach outperforms state-of-the-art methods in terms of overall accuracy. Moreover, we implemented our approach using Robot Operating System (ROS) environment to validate the DBMM running on-the-fly in a mobile robot with an RGB-D sensor onboard to identify daily activities for a robot-assisted living application.**

I. Introduction

Nowadays with the advances of technology and the broad research worldwide, a cognitive robot can act as human assistant in the context of robot-assisted living, and also having the potential to offer social and entertaining interaction experiences through human-robot interaction. For that, in order to enable this natural human-robot interaction, the robot needs to infer the human intentions, their daily routine and potential risk situations by observing them. In this work, we focus our attention in the domain of human daily activity recognition. In this context, a robot that can recognize daily activities will be useful for assisted care: human-robot or child-robot interaction (e.g. in coping tasks); and also monitoring elderly and disabled people regarding their activities at home. In our previous work [3], we proposed a Dynamic Bayesian Mixture Model (DBMM) that was applied as a probabilistic loop, where the model recursively uses the prior information to reinforce current classification as a first-order Markov process. Herein, we are extending this model by using the memory of the system for dynamic update of the weighted ensemble, adjusting the weights based on previous behaviors of the base classifiers to improve the performance of classification. We validated the DBMM performance using different datasets and also using a mobile robot in an on-the-fly application for monitoring tasks. In the scope of human daily activity recognition, experimental results show that our proposed probabilistic ensemble is robust and with better performance than single classifiers and state-of-the-art approaches as well. Notice that, our framework relies only on 3D skeleton-based features, which is enough to characterize different classes of activities. The main impact of this work are the following:

- Employing a local update of weights on the DBMM using the memory of the system (i.e. previous base classifier behaviors) to obtain better classification performance.
- Modeling meaningful spatio-temporal features relying on skeleton distances, energy model and autocorrelation of joint translational movements, which can successfully characterize different activities.
- Assessment and validation: (i) comparing with single classifiers and state-of-the-art activity recognition approaches; and (ii) on-the-fly tests using a mobile robot for robot-assisted living.

The remainder of this paper is organized as follows. Section II covers selected related works. Section III introduces our approach, detailing the extended model with dynamic update of weights. The proposed skeleton-based features is presented in section IV. Section V presents the performance of the DBMM using state-of-the-art datasets and using a mobile robot for assisted living. Finally, Section VI brings the conclusion and future work.

II. Related Work

By looking to recent advances of works that use RGB-D sensors, several works focus on human-pose detection for human activity recognition [4] [5]. In [6], a maximum entropy Markov model (MEMM) for human activities classification was adopted, where features were modeled using the Histogram of Oriented Gradient (HOG). In [7], each activity is modeled into sub-activities, while object affordances and their changes over time were used with a multi-class Support Vector Machine (SVM) classifier. In [8], a bag of kinematic features was used with a set of SVMs, for activity classification. Other works on the recognition of human activities focus their research on how to model the attributes efficiently, to successfully obtain reliable classification [9] [10] [11]. In [12], a descriptor which couples depth and spatial information to describe humans body-pose was proposed. This approach is based on segmenting masks from depth images to recognize an activity. Sparse coding and temporal pyramid

This work has been supported by the Portuguese Foundation for Science and Technology, COMPETE and QREN programs under Grant AMS-HMI12 RECI/EEI-AUT/0181/2012. The authors are with Institute of Systems and Robotics, Dept. of Electrical and Computer Engineering, University of Coimbra, Polo II, 3030-290 Coimbra, Portugal (emails: diego, mario, cpremebida, urbano@isr.uc.pt).

matching is proposed in [13] for human action recognition. They use depth data for a learning algorithm that employs a discriminative class-specific dictionary. In [14], a feature descriptor for action recognition is presented. Depth motion maps are built given projection views in order to capture motion cues. Later on, a compact feature representation is obtained by using local binary patterns. Regarding our proposed framework, it allows the combination of different classifier models, which is advantageous to increase the classification performance. The DBMM dynamically reinforces the classification as a probabilistic loop, updating the initial learned weights given a confidence level to generate a distribution conditioned to the previous posteriors. Moreover, the DBMM approach has success in obtaining better results compared with benchmarked methods for activity recognition.

III. Probabilistic Classification Model: DBMM

DBMM is an ensemble of classifiers designed to combine a set of single classifiers (also referred as base classifiers) towards obtaining more accurate results than any of its individual members. For that, a probabilistic approach was adopted, using the concept of mixture models in a dynamic form in order to combine conditional probabilities. A weight is assigned to each base classifier, according to previous knowledge (learning process), using an uncertainty measure as a confidence level, and can be updated locally during the online classification. In our solution, the local weight update assigns priority to the base classifier with more confidence along the temporal classification, since they can vary along the different frame classifications. Figure 1 depicts an example of DBMM classification, where base classifiers are integrated as weighted posterior distributions, and previous posteriors and weights are used to update the model. The DBMM uses a set of models $A = \{A_m^1, A_m^2, ..., A_m^T\}$ where A_m^t is a model with m attributes; i.e., observed variables generated for some dynamic process at $t = \{1, 2..., T\}$. The DBMM probability distribution function for each class $P(C,A) = \prod_{t=1}^{T} P(C^t|C^{t-1}) \times \sum_{i=1}^{n} w_i \times P_i(A|C^t)$ can be rewritten holding the Markov property by taking the posterior of previous time instant as the new prior as follows:

$$P(C|A) \;=\; \beta \;\times\; \underbrace{P(C^t|C^{t-1})}_{\text{dynamic transitions}} \;\times\; \underbrace{\sum_{i=1}^{n} w_i^t \times P_i(A|C^t)}_{\text{mixture model with dynamic w}},$$

$$\text{with} \begin{cases} P(C^t|C^{t-1}) \equiv \frac{1}{C} \;\text{(uniform)}, & t = 1 \\ P(C^t|C^{t-1}) = P(C^{t-1}|A), & t > 1 \end{cases}, \tag{1}$$

where:

- $P(C^t \mid C^{t-1})$ is the transition probability distribution among class variables over time. A class C^t is conditioned to C^{t-1}. This means a non-stationary behavior applied recursively, then reinforcing the classification at time t.
- $P_i(A^t)$ is the posterior result of each base classifier at time t, $i = \{1, ..., n\}$.
- The weight in the model for each base classifier w_i^t is initially estimated using an entropy-based confidence on the training set (offline) as shown in our previous work [3], and afterwards ($t > 5$) it is updated as explained in the next subsection.
- $\beta = \frac{1}{\sum_j \left(P(C_j^t|C_j^{t-1}) \times \sum_{i=1}^{n} w_i \times P_i(A|C_j^t) \right)}$ is a normalization factor, ensuring numerical stability once continuous update of belief is done.

A. Dynamic Update of Weights using the System's Memory

During a classification task, base classifiers can change the performance over time. Thus, the local update of the weights during the on-line classification will benefit from the fact that the adjusted weights will produce a higher belief when priority is assigned to a base classifier with more confidence on previous classifications. We update the ensemble model using the temporal information on the test set as the memory of the system (set with previous posteriors for each base classifier $\Omega_i^s = \{P(C|A)^{t-1} ... P(C|A)^{t-s}\}$ together with the weights at the previous time instant w_i^{t-1}. Thus, in order to apply an update on the current weights, we compute:

$$w_i^t = \frac{w_i^{t-1} \times P(w_i|H_i(\Omega^s))}{\sum_{i=1}^{n} w_i^{t-1} \times P(w_i|H_i(\Omega^s))}, \tag{2}$$

where w_i^t is the estimated weight for each base classifier (updated); w_i^{t-1} is the previous weight at $t-1$. In order to obtain $H_i(\Omega^s)$, we use the memory of the system during the classification by keeping the previous posteriors (up to 5^{th} order), and consequently, we acquire the the entropy on each set of posteriors $H_i(\Omega^s)$ as follows:

$$H_i(\Omega^s) = -\sum_{j}^{s} H_i(\Omega^j) \log(H_i(\Omega^j)). \tag{3}$$

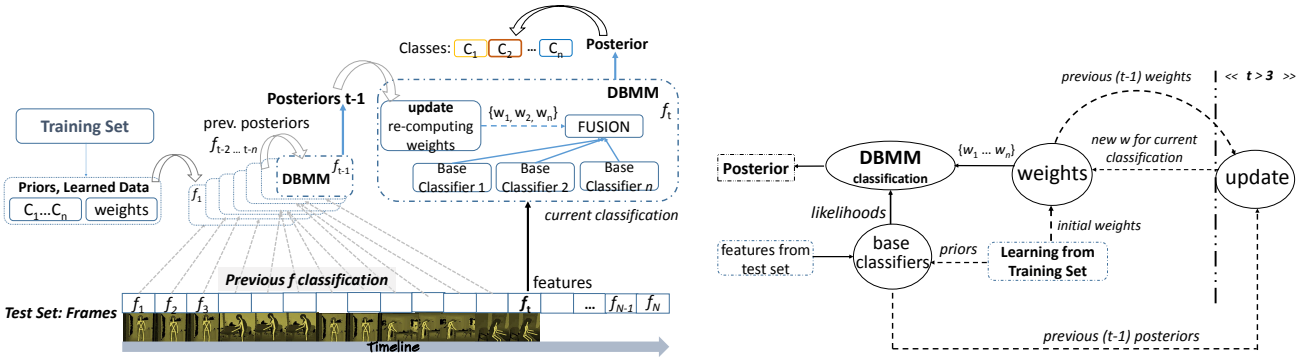
Knowing $H_i(\Omega^s)$ for each base classifier, the weights $P(w_i|H_i(\Omega^s))$ are estimated inversely proportional to the entropy:

$$P(w_i|H_i(\Omega^s)) = \frac{\left[1 - \left(\frac{H_i(\Omega^s)}{\sum_{i=1}^{n} H_i(\Omega^s)}\right)\right]}{\sum_{i}^{n} \left[1 - \left(\frac{H_i(\Omega^s)}{\sum_{i=1}^{n} H_i(\Omega^s)}\right)\right]}, \; i = \{1, ..., n\}, \tag{4}$$

where w_i is the result for each base classifier, and H_i is the current value of entropy given by (3). The denominator in (4) ensures that $\sum_i w_i = 1$.

B. Base Classifiers for DBMM Fusion

In this work, we have used the Naive Bayes Classifier (NBC), Support Vector Machines (SVM) and an Artificial Neural Network (ANN) as base classifiers for the DBMM. The NBC assumes the features are independent from each other given a class, $P(C_i|A) = \alpha P(C_i) \prod_{j=1}^{m} P(A_j|C_i)$. For the linear-kernel multiclass SVM implementation, we adopted the LibSVM package [15], trained according to the 'one-against-one' strategy, with *soft margin* (or Cost) parameter set to 1.0, and classification outputs were given in terms of probability estimates. The ANN adopted is a multilayer



(a) DBMM classification approach

(b) DBMM update cycle

Fig. 1: Example of DBMM during frame to frame classification in activity recognition. The left image shows that during the dynamic classification, initially the weights are learned from the training set, and later on, during the test they are updated.

feedforward network (with 40 neurons in the hidden layer), where the hidden layer transfer function is a hyperbolic tangent sigmoid and a normalized exponential (*softmax*) is used for the output of the transfer function as posterior probability estimates, conditional on the input, i.e., $\sum_{i=1}^{n} P(C_i|x) = 1$.

IV. Spatio-Temporal Skeleton-based Features

It is of utmost importance to find discriminative features of daily activity relying on existing relations between skeleton body parts to model their motion by correlating different time instants. The skeleton detection and tracking is made using depth images, adopting the OpenNi's software development kit for RGB-D sensor to obtain the joint locations of the human body.

We defined a set $\mathbf{F}$ with 51 features per frame to discriminate daily activities. Features based on skeleton joint distances, velocities and difference of skeleton poses along different frames are used in this work. Three types of spatio-temporal features are substantiated in the energy concept: 1) energy-based features using the joint velocities, 2) log-energy entropy-based features using skeleton poses, and 3) sample autocorrelation-based features using the distances of skeleton poses in different time instants. The velocities energy of the upper joints of the skeleton (i.e. seven joints: head; left and right shoulders, hands and elbows) are computed as follows:

$$E_{uv} = \sum_{j=1}^{N}(V_{jx})^2 + \sum_{j=1}^{N}(V_{jy})^2 + \sum_{j=1}^{N}(V_{jz})^2, \quad \text{with} \quad V_{jd} = \frac{\mathbf{S}^t_{jd} - \mathbf{S}^{t-s}_{jd}}{\Delta T}, \quad d = \{x,y,z\}, \tag{5}$$

where for each dimension $\{x,y,z\}$, $\mathbf{S}_j$ is a vector of dimension 7×1, whose elements are the skeleton joints; for the computation of V_{jd}, the numerator corresponds to the skeleton joints distances from t to t_s preceding frames (herein, $s = 10$), and the denominator corresponds to the elapsed time $\Delta T = f_{rate} \times \varpi$ (a frame rate $f_{rate} = 1/30$ and a temporal slide window $\varpi = 10$ were used).

The second feature is based on the sum of log-energy entropy $logE_s$ using the global skeleton joints in each dimension as follows:

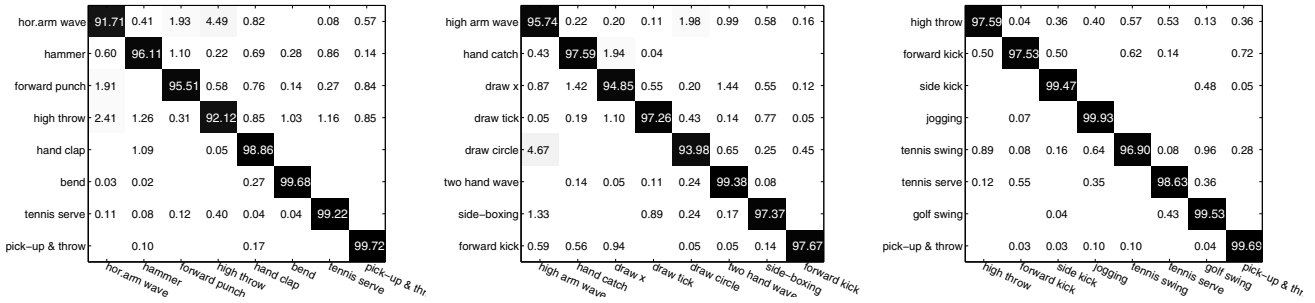
$$logE_s = \sum_j \log(\mathbf{S}^2_{jx}) + \sum_j \log(\mathbf{S}^2_{jy}) + \sum_j \log(\mathbf{S}^2_{jz}). \tag{6}$$

The two aforementioned features enclose key poses of movements, i.e., when the skeleton joints alternately show acceleration and deceleration in repeated movements that leads to changes in the energy model representation. This information helps the characterization of drastic changes in direction and velocities of the skeleton. The energy model (5) is applied to the upper body part and the log-entropy (6) is applied to all body joints.

The third feature is based on the autocorrelation function employed on the difference of skeleton poses at time t and $t-1$. The first step before computing the autocorrelation is to obtain the translation of each skeleton joint S_j from a time instant $t-1$ to the current time instant t by employing the Euclidean distance $\delta_{\{S^t_{jd}, S^{t-1}_{jd}\}} = \sqrt{(S^t_{jd} - S^{t-1}_{jd})^2}$, $d = \{x,y,z\}$, obtaining a matrix of $N \times d$ (i.e., number of joints N and d-dimensional space). Subsequently, the sample autocorrelation is computed by:

$$r(\tau) = \frac{\frac{1}{T-1}\sum_{t=1}^{T-\tau}\left(\delta^t_{\{S^t,S^{t-1}\}} - \mu^t_\delta\right)\left(\delta^{t+\tau}_{\{S^t,S^{t-1}\}} - \mu^{t+\tau}_\delta\right)}{\sigma^2} \tag{7}$$

where $\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}\left(\delta_{\{S^t,S^{t-1}\}} - \mu_\delta\right)^2$ is the sample variance and μ_δ is the sample mean value; and τ is the lag variable of a process at different times. Since we are working with 3D skeleton arranged in a matrix $\delta_{\{S^t,S^{t-1}\}}$ of 20×3 (joints by 3 dimensions), then in order to facilitate the autocorrelation computation, we applied a self-convolution, whereas the autocorrelation is alike to a convolution, apart from it does not need to flip an input about the origin. Thus, 2D convolution in spatial form for finite intervals is achieved by $f * g = c(i,j) = \sum_k^p \sum_l^q f(k,l) \times g(i-k, j-l)$, where $f = \delta_{\{S^t,S^{t-1}\}}$, and g which commonly has the role of the filter in convolution, herein it is in charge of the shift of f with respect to itself (rotates about the origin) in the plane $p \times q$. A resulting matrix that is given by $f * g$ has a



(a) AS1: Prec:96.59%; Rec:96.61%; Acc:96.61% (b) AS2: Prec:96.76%; Rec:96.73%; Acc:96.73% (c) AS3: Prec:98.66%; Rec:98.65%; Acc:98.65%

Fig. 2: Classification results: cross-validation confusion matrix for each action set using the DBMM for the "new person" setting (training five persons and testing on other "unseen" five). Global Prec.: 97.34%, Rec.: 97.33%, Acc.: 97.33%.

dimension of $m \times n$ $(2 \times \text{size}(\delta_{\{S^t,S^{t-1}\}}) - 1) = 39 \times 5$, was then reshaped as a feature vector $\mathbf{r}$ of $(m \times n) \times 1$ elements to compute the autocorrelation energy $E_r = \sum_i \mathbf{r}_i^2$.

Additionally, a set of features based on Euclidean distances of the skeleton joints $\delta_{\{S^d_{j1},S^d_{j2}\}}$ was used, as similarly presented in [3]: 1) the minimum distance from hand (left or right) to the head, e.g. $\min(\delta_{\{S^d_{j1},S^d_{j2}\}}, \delta_{\{S^d_{j1},S^d_{j3}\}})$; 2) the minimum distance from elbow (left or right) to the head; 3) the minimum distance from hand (left or right) to the center of the skeleton; 4) distance from the left hand to the right hand; 5) distance from the head to the center of the hip; 6) distance from the central knee (mean coordinate taking into account the left and right knees) to the center of the hip; 7) the minimum distance from foot (left or right) to the head; 8) the hand with higher changes in directions (i.e., using the difference of the current position to a previous one); 9) six angles obtained from triangles formed by: shoulder, hand and elbow; hip, shoulder and knee; hip, knee and foot, all considering left and right sides. The angle computation is given by $\theta_i = \arccos(\delta^2_{j_{12}} + \delta^2_{j_{23}} - \delta^2_{j_{13}} \;/\; 2 \times \delta_{j_{12}} \times \delta_{j_{23}})$, where $\delta_{j_{12}}$ is the Euclidean distance between two joints. These angles are useful to discriminate stand and seated positions or torso inclination.

Then, a stage consisting of derivatives and accumulative values was employed on the aforementioned set of extracted features $\mathbf{F}$. We first applied a discrete derivative $y = \frac{\mathbf{F}^t - \mathbf{F}^{t-s}}{\Delta T}$ on each feature, where s represents a temporal slide window of ten frames. Subsequently, we accumulated each feature value over the frames: $y^t_{cum} = \sum_{k=1}^{t} \mathbf{F}_k$. Thus, with these two steps we obtained more 34 features, and $\mathbf{F}$ sums up to a total of 51 features. To ensure a higher classification performance, an essential step is employed; the extracted set of features are normalized in such a way that, values of minimum and maximum obtained during the training were applied on the normalization of the test set.

V. Assessment of the Proposed Framework on Datasets and Robotic Application

Experimental tests using a mobile robot and two datasets were performed to assess our framework. Looking at the performance attained, we can state that our framework has good potential for activity recognition in robot-assisted living.

A. Performance on MSR-Action3D Dataset

The MSR-Action3D dataset [1] contains skeleton data from depth images captured by an RGB-D sensor at 15Hz. MSR-Action3D comprises twenty actions, and each action was performed by ten subjects for three times. The actions cover various movement of arms, legs, torso and their combinations. For performance evaluation purposes, and concerning this dataset, we followed the same methodology as described in [1] [2], where the dataset is split into 3 action sets with eight actions each one as shown in Fig. 2. As stated in [1], AS1 and AS2 group actions with similar movements, while AS3 groups actions that are more complex. We follow the cross-validation test as defined by [2] and [16]. The tests were performed by training five subjects out of ten, and testing on the other five subjects (testing on "unseen persons"), e.g., training persons {1,3,5,7,9} and testing on persons {2,4,6,8,10}; afterwards the opposite (even, odd); then, training on persons {1...5} and testing on persons {6...10}, and so on. Taking into consideration 5×5 splits, there are 252 possible splits in total. The overall accuracy (average) was computed to compare our proposed framework with other state-of-the-art methods. Results show that our proposed framework outperforms other state-of-the-art benchmarked methods using this dataset up to the current date. The overall accuracy obtained with the DBMM was 97.33%, taking the average of all attained performances. Figure 2 presents the overall confusion matrix for the cross-subject classification for each action set. Table I summarizes the results attained by the DBMM in comparison with each single classifier and an averaged ensemble for AS1, AS2 and AS3, showing that our approach outperforms the other classifiers (all using our skeleton features). Finally, Table II presents the results of our DBMM approach in comparison with other state-of-the-art methods evaluated using the MSR-Action3D dataset. This table references some selected works, the ones with higher overall accuracy up to date.

Our approach using only 3D skeleton features outperforms other approaches that use features from skeleton, from depth

TABLE I: Accuracy on action sets using single classifiers, a simple averaged ensemble (AV) and the proposed DBMM.

Action Set	SVM	Bayes	ANN	AV	**DBMM**
AS1	92.8%	89.3%	90.8%	90.9%	96.6%
AS2	91.7%	88.4%	90.4%	90.1%	96.7%
AS3	94.6%	89.9%	92.7%	92.4%	98.6%
Average	93.0%	89.2%	91.3%	91.1%	97.3%

TABLE II: Comparison of approaches that use the MSR-Action3D in terms of overall accuracy. Columns 3 an 4 point out the feature types used by the approaches.

Method	Acc	SK joints	Depth
Proposed framework (DBMM)	**97.33**%	✗	
* Luo *et al.* [13]	97.26%	✗	✗
Chen *et al.* [14]	94.90%	✗	
Ohn-Bar and Trivedi [17]	94.84%	✗	✗
Yang, Zhang and Tian [18]	91.63%	✗	
Chaudhry *et al.* [19]	90.00%	✗	
Evangelidis *et al.* [20]	89.86%	✗	
Oreifej and Liu [16]	88.89%		✗
Wang *et al.* [2]	88.20%	✗	✗

*The approach in [13] obtained 96.7% when using only skeleton features

and even approaches that combine both.

B. Performance on MSR-DailyActivity3D Dataset

The MSR-DailyActivity3D [2] is another dataset with depth images and 3D skeleton data from an RGB-D sensor that was used herein to evaluate our approach. It contains 16 activities: *1-drink, 2-eat, 3-read book, 4-call cellphone, 5-write on a paper, 6-use laptop, 7-use vacuum cleaner, 8-cheer up, 9-sit still, 10-toss paper, 11-play game, 12-lie down on sofa, 13-walk, 14-play guitar, 15-stand up, 16-sit down* performed by 10 subjects twice, where one trial is in standing position, and the second in sitting position on a sofa. We followed the state-of-the-art methodology [2] for evaluation of our framework. This dataset has all 16 activities in a single scenario, i.e., a multi-class cross-subject classification. The tests were performed in the same way of the MSR-Action3D by training five subjects out of ten, and testing on the other five subjects ("unseen persons"). The results attained are shown by means of a confusion matrix in Fig. 3. To the best of our knowledge, our results outperforms other state-of-the-art methods applied on MSR-DailyActivity3D dataset up to the current date. The overall performance obtained with the DBMM approach are: precision of 97.39%; recall of 96.83%; and accuracy of 96.83%. Table III shows the overall accuracy of our approach compared with some selected works of the state-of-the-art, i.e. the ones with higher accuracy for this dataset up to the current date.

C. Performance using a Mobile Robot

In order to evaluate our approach using a mobile robot, we built a dataset (e.g. Fig. 4) with RGB-D image sequences and skeleton data to learn human daily activities, such as *1-walking, 2-stand/still, 3-talking on the phone, 4-working on a computer and 5-sitting*; and for suspicious or risk situations: *6-jumping, 7-falling down, 8-running*. We recorded 4 persons performing 3 times each activity during 30 up to 45 seconds.

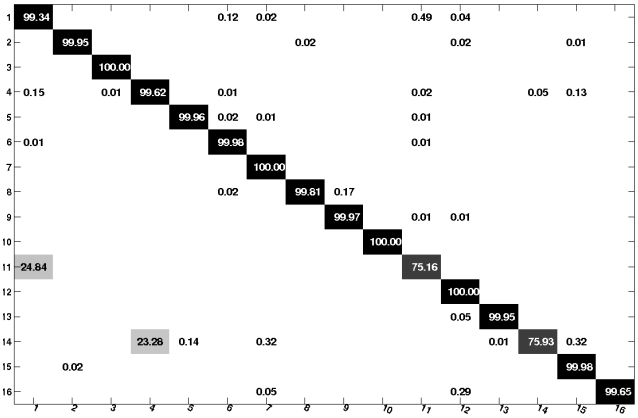


Fig. 3: Confusion Matrix obtained from the DBMM classification applied on the MSR-DailyActivity3D dataset.

TABLE III: Comparison of approaches that use the MSR-DailyActivity3D in terms of overall accuracy. Columns 3 an 4 point out the feature types used by the approaches.

Method	Acc	SK joints	Depth
Proposed framework (DBMM)	**96.83**%	✗	
Luo *et al.* [13]	95.00%	✗	✗
Xia and Aggarwal [21]	88.20%	✗	✗
Wang *et al.* [2]	85.75%	✗	✗

Robot Operating System (ROS) packages in *hydro* version were used to program the mobile robot to navigate in an indoor environment. For that, the robot has different sensors onboard, such as laser for mapping and localization, avoiding obstacle collision, and an RGB-D sensor for human body detection for skeleton tracking and human activity recognition. Reminding that, in this work, the focus of our attention is on the evaluation of our probabilistic approach for activity recognition on-the-fly, thus, herein we do not detail other robot functionalities (e.g., navigation and robot (re)actions). Once the skeleton is detected in a range of two up to five meters to the RGB-D sensor, the robot starts the activity recognition. In this experiment, a robot response is assigned for each activity that is recognized (e.g. during a monitoring task, when a usual activity is classified, the robot will just re-position itself to keep monitoring). For each risk situation detected, the robot is supposed to assist somehow, by sending warnings or calling relatives to report the current situation. Figure 5 shows the cognitive system for activity recognition in robot-assisted living (monitoring task) using ROS environment[1].

The strategy to test an on-the-fly application using a mobile robot is a little different than the evaluation on datasets. In this case, the DBMM classification is made in 3 up to 5 seconds to guarantee a confidence for a final decision, i.e., after recognizing the activity, the robot will respond with an action. Figure 6 shows few snapshots of the experiments of daily activities including a risk situation that

[1]A video demonstrating our approach for robot-assisted living can be seen at `https://youtu.be/FAfLj28_iSM`

Fig. 4: Few examples of the dataset (RGB and depth images) which was built to learn some daily and risk situations.

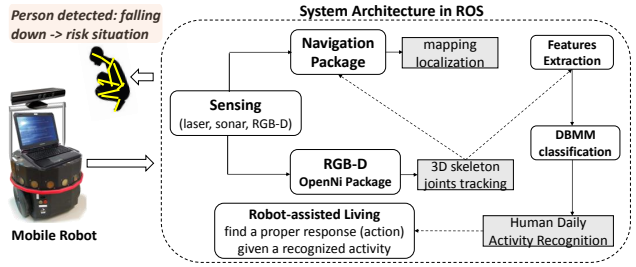


Fig. 5: Architecture in ROS of our artificial cognitive system for robot-assisted living.

our mobile robot correctly recognized. During the on-the-fly experiments using a mobile robot, all activities performed twice by two "unseen" persons were correctly classified. The overall confidence of classification in the context of robot-assisted living is presented in a confusion matrix as shown in Fig. 7, with overall accuracy of 90.46%. We noticed that the activities can be correctly classified with a high certainty within 3 up to 6 seconds of frames by frame classification. The activities *walking* and *running* were the ones with more misclassification due to their strong similarities.

VI. Conclusion and Future Work

A dynamic probabilistic ensemble of classifiers (DBMM) using a local update of weights was designed for activity recognition. The local weighting strategy to update the model has shown through experimental results to be very effective given a set of suitable features. Two well-known state-of-the-art datasets of human daily activities, Microsoft Research [1] [2], were used to evaluate the performance of our approach. The classification performance in terms of overall accuracy has shown that our proposed framework outperforms other methods in the scope of human daily activity recognition. In addition, we performed experimental tests of our approach running on-the-fly in a mobile robot for monitoring daily activities and risk situations, showing that it has potential to successfully be used in robot-assisted living applications. Future work will exploit and extend our framework for robot-assisted living and natural human-robot interaction scenarios.

Fig. 6: Few snapshots of daily activities recognition experiments ('unseen" person) using a mobile robot.

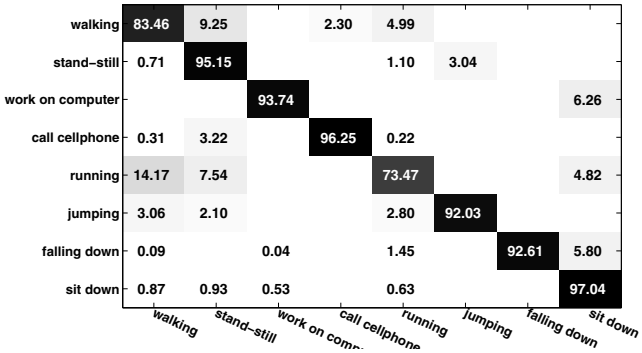


Fig. 7: DBMM on-the-fly classification confidence (average).

References

[1] W. Li, Z. Zhang, and Z. Liu, "Action recognition based on a bag of 3d points,," in *IEEE CVPRW: Human Comm. Behav. Analysis*, 2010.

[2] J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Mining actionlet ensemble for action recognition with depth cameras," in *IEEE CVPR*, 2012.

[3] D. R. Faria, C. Premebida, and U. Nunes, "A probalistic approach for human everyday activities recognition using body motion from RGB-D images," in *IEEE RO-MAN'14, * Kazuo Tanie Award Finalist*, 2014.

[4] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from a single depth image," in *IEEE CVPR*, 2011.

[5] J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Learning actionlet ensemble for 3d human action recognition," in *IEEE Transactions on PAMI*, 2013.

[6] J. Sung, C. Ponce, B. Selman, and A. Saxena, "Unstructured human activity detection from RGBD images," in *ICRA'12*, 2012.

[7] H. S. Koppula, R. Gupta, and A. Saxena, "Learning human activities and object affordances from RGB-D videos," in *IJRR journal*, 2012.

[8] C. Zhang and Y. Tian, "RGB-D camera-based daily living activity recognition," in *J. of Comp. Vision and Image Proc.*, 2012.

[9] X. Yang and Y. Tian, "Effective 3d action recognition using eigenjoints," *J. of Visual Comm. and Image Repr.*, vol. 25, pp. 2–11, 2013.

[10] L. Piyathilaka and S. Kodagoda, "Gaussian mixture based HMM for human daily activity recognition using 3d skeleton features," in *IEEE 8th Conf. on Ind. Electronics and App.*, 2013.

[11] B. Ni, Y. Pei, P. Moulin, and S. Yan, "Multilevel depth and image fusion for human activity detection," *IEEE Trans. on Cybern.*, 2013.

[12] R. Gupta, A. Y.-S. Chia, and D. Rajan, "Human activities recognition using depth images," in *21st ACM Int. Conf. on Multimedia*, 2013.

[13] J. Luo, W. Wang, and H. Qi, "Group sparsity and geometry constrained dictionary learning for action recognition from depth maps," in *ICCV'13*.

[14] C. Chen, R. Jafari, and N. Kehtarnavaz, "Action recognition from depth sequences using depth motion maps-based local binary patterns," in *IEEE Winter Conf. on App. of Computer Vision (WACV)*, 2015.

[15] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM TIST*, 2011, http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[16] O. Oreifej and Z. Liu, "HON4D: Histogram of oriented 4d normals for activity recognition from depth sequences," in *IEEE CVPR*, 2013.

[17] E. Ohn-Bar and M. M. Trivedi, "Joint angles similiarities and HOG2 for action recognition." in *CVPRW*, 2013.

[18] X. Yang, C. Zhang, and Y. Tian, "Recognizing actions using depth motion maps-based histograms of oriented gradients," in *ACM International Conf. on Multimedia*, 2012.

[19] R. Chaudhry, F. Ofli, G. Kurillo, R. Bajcsy, and R. Vidal, "Bio-inspired dynamic 3d discriminative skeletal features for human action recognition," in *Comp. Vision and Pattern Rec. WS. (CVPRW)*, 2013.

[20] G. Evangelidis, G. Singh, and R. Horaud, "Skeletal quads: Human action recognition using joint quadruples," in *ICPR*, 2014.

[21] L. Xia and J. Aggarwal, "Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera," in *CVPR*, 2013.

Appendix B

Paper Accepted for Presentation at the ROBOT' 2015 Iberian Conference

Real-time Application for Monitoring Human Daily Activity and Risk Situations in Robot-Assisted Living

Mário Vieira, Diego R. Faria and Urbano Nunes*

Institute of Systems and Robotics, Department of Electrical and Computer Engineering, University of Coimbra, Portugal

Abstract. In this work, we present a real-time application in the scope of human daily activity recognition for robot-assisted living as an extension of our previous work [1]. We implemented our approach using Robot Operating System (ROS) environment, combining different modules to enable a robot to perceive the environment using different sensor modalities. Thus, the robot can move around, detect, track and follow a person to monitor daily activities wherever the person is. We focus our attention mainly on the robotic application by integrating several ROS modules for navigation, activity recognition and decision making. Reported results show that our framework accurately recognizes human activities in a real time application, triggering proper robot (re)actions, including spoken feedback for warnings and/or appropriate robot navigation tasks. Results evidence the potential of our approach for robot-assisted living applications.

1 Introduction

Mobile robots endowed with cognitive skills are able to help and support humans in an indoor environment, providing increased availability, awareness and access, as compared to static systems. Thus, a robot can act not only as assistant in the context of robot-assisted living, but also offer social and entertaining interaction experiences between humans and robots. For that, the robot needs to be able to understand human behaviours, distinguishing human daily routine from potential risk situations in order to react in accordance. In this work, we focus our attention on the domain of human-centered robot application, more precisely, for monitoring tasks, where a robot can recognize daily activities and unusual behaviours to react according to the situation. In this context, a robot that can recognize human activities will be useful for assisted care, such as human-robot or child-robot interaction and also monitoring elderly and disabled people regarding strange or unusual behaviours. We use a robot with an RGB-D sensor (Microsoft Kinect) on-board to detect and track the human skeleton in order to extract

* This work was supported by the Portuguese Foundation for Science and Technology (FCT) under the Grant AMS-HMI12: RECI/EEI-AUT/0181/2012. The authors are with Institute of Systems and Robotics, Department of Electrical and Computer Engineering, University of Coimbra, Polo II, 3030-290 Coimbra, Portugal (emails: mvieira, diego, urbano@isr.uc.pt).

motion patterns for activity recognition. We present an application that combines different modules, allowing the robot localization and navigation in an indoor environment, and also to detect obstacles and human skeleton for motion tracking. In addition, we use modules for voice synthesizer and recognition, that will be triggered by our activity recognition module. The activity recognition module uses a Dynamic Bayesian Mixture Model (DBMM) [2] [1] for inference, in order to classify each activity, enabling the mobile robot to make a decision to react accordingly. The main contributions of this work are:

- Combining different ROS modules (navigation, classification and reaction module), towards a real time robot-assisted living application.
- Extending the use of DBMM to real-time applications using proposed discriminative 3D skeleton-based features, which can successfully characterize different daily activities.
- Assessment and validation: (i) leave-one-out cross validation of the activity recognition using our training dataset; (ii) comparison of different classification models using our proposed features; (iii) online validation of the integrated artificial cognitive system.

The remainder of this paper is organized as follows. Section 2 covers selected related work. Section 3 introduces our approach, detailing the proposed 3D skeleton-based features as well as the classification method. Section 4 describes how the approach is implemented in ROS. In section 5, the performance of the proposed application is presented. Finally, Section 6 brings the conclusion of this research pointing future directions.

2 Related Work

In order to have a fully operational robot-assisted living application, it is essential that the robot can recognize daily activities in real scenarios, in real-time. In spite of some proposed works that use inertial sensors for human activity recognition [3] [4], the most common approaches use vision-based depth sensors, even more nowadays, with low cost vision sensors (e.g. RGB-D sensors [5] [6]) that can track the entire human body accurately. In [7], a Microsoft Kinect sensor is used to track the skeleton and posteriorly extract the features. The action recognition is done using first order Hidden Markov Models (HMMs) and for every hidden state, the observations were modelled as a mixture of Gaussians. The work presented in [8] uses depth motion maps as features for activity recognition. Other works on the recognition of human activities focus their research on how to extract the right features in order to obtain better classification performance [9] [10] [11]. In the context of robot assisted living, [12] describes a behaviour-based navigation system in assisted living environments, using the mobile robot ARTOS. In [13] a PR2 robot is used to assist a person. The robot detects the activity being performed as well as the object affordances, enabling the robot to figure out how to interact with objects and plan actions. In [14], a mobile robot is used in a home environment to recognize activities in real-time by continuously tracking the pose and motion of the user and combining them with structural knowledge like the current

room or objects in proximity. In our work, we use a Nomad Scout with a laser Hokuyo to assist the localization and navigation module, and an RGB-D sensor on-board to detect and track a person. It is a small mobile robot that monitors a person in an indoor environment, recognizing daily and risky activities and reacts with defined actions, assisting the person if needed. Our activity recognition module is based on the framework proposed in [1], where the features are also skeleton-based, however, herein we model different skeleton-based features, and in addition, we use a new collected dataset.

3 Activity Recognition Framework

3.1 Extraction of 3D Skeleton-based Features

We have used a Microsoft Kinect sensor and the OpenNi's tracker package for ROS to detect and track the human skeleton. This package allows the skeleton tracking at 30 frames per second, providing the three-dimensional Euclidean coordinates of fifteen joints of the human body with respect to the sensor. Using this information, we compute a set of features as follows:

- Euclidean distances among the joints, all relative to the torso centroid, obtaining a 15×15 symmetric matrix with a null diagonal. Let *(x,y,z)* be the 3D coordinates of two body joints b_j with $j = 1,2,...,15$ and b_i with $i = 1,2,...,15$, then $\forall\ \{b_i, b_j\}$, the distances were computed as follows:

$$\delta(b_j, b_i) = \sqrt{(b_j^x - b_i^x)^2 + (b_j^y - b_i^y)^2 + (b_j^z - b_i^z)^2} \tag{1}$$

  Subsequently, we removed the null diagonal, obtaining a 14×15 matrix $\mathbf{M}$ to compute its *log-covariance* as follows:

$$\mathbf{M}_{lc} = \mathbf{U}(\log(\mathrm{cov}(\mathbf{M}))), \tag{2}$$

  where $\mathrm{cov}(\mathbf{M}_{i,j}) = (M_i - \mu_i)(M_j - \mu_j)$; $\log(\cdot)$ is the matrix logarithm function (logm) and $\mathrm{U}(\cdot)$ returns the upper triangle matrix composed by 120 feature elements. The rational behind of log-covariance is the mapping of the convex cone of a covariance matrix to the vector space by using the matrix logarithm as proposed in [15]. A covariance matrix form a convex cone, so that it does not lie in Euclidean space, e.g., the covariance matrix space is not closed under multiplication with negative scalers. The idea of log-covariance is based on [16], where examples of manifold Riemannian metrics and log-covariance applied in 2D image features for activity recognition were used.
- The global skeleton velocities, assuming the 3D coordinates of 14 joints in the case of having the torso centroid as origin; and 15 joints in the case of having the sensor frame as origin were computed as follows:

$$v_j = \frac{\sqrt{(b_{j_x}^t - b_{j_x}^{t-t_w})^2 + (b_{j_y}^t - b_{j_y}^{t-t_w})^2 + (b_{j_z}^t - b_{j_z}^{t-t_w})^2}}{f_{rate} \times t_w}, \tag{3}$$

  where v_j is the velocity of a specific skeleton joint j; b_{j_d} represents the position $d = \{x,y,z\}$ of a skeleton body joint j in the current time t, and $t - t_w$ represents some preceding frames, herein $t_w = 10$; the frame rate is set to $f_{rate} = 1/30$.

– Differently of the aforementioned velocities in the torso frame of reference, herein, relative to the sensor frame, for all joints, for each dimension individually, we computed the difference $\delta(b^t_{j_d}, b^{t-t_w}_{j_d})$ between the position at a given frame and the preceding 10^{th} frame. Using these values, we computed the velocities of the same joints for each dimension individually, $v_j = \frac{b^t_{j_d} - b^{t-t_w}_{j_d}}{f_{rate} \times t_w}$, obtaining additional 45 features.
– The angles variation of certain joints play a crucial role in carrying out many activities. We are interested in knowing whether a person is sitting or standing, so we compute the angles of both right and left elbows in the triangle formed by the hands, elbows and shoulders. We also compute the angles of the hip joints in the triangle formed by the shoulders, hips and knees and the angles of the knees in the triangles formed by the feet, knees and hips. The angle θ_i is given by:

$$\theta_i = \arccos\left(\frac{(\delta_{j_{12}})^2 + (\delta_{j_{23}})^2 - (\delta_{j_{13}})^2}{2 \times \delta_{j_{12}} \times \delta_{j_{23}}}\right), \tag{4}$$

where $\delta_{j_{12}}$ is the distance between two joints, e.g. j_1 and j_2, that are forming a triangle in the skeleton. We have 2+2+2=6 features for angles, since we are considering the left and right side for the body joints. In addition, we compute the difference between these angles at a current frame and the preceding 10^{th} frame, $\theta_{v_i} = \theta^t_i - \theta^{t-10}_i$, obtaining additional 2+2+2=6 features.

Thus, in total, we attained a set with 206 spatio-temporal skeleton-based features, useful to discriminate different classes of activities.

Features pre-processing: Before using the features set in the classification module, we perform a pre-processing step. Normalization, standardization or filtering may be a requirement for many machine learning estimators, as they can behave badly if no pre-processing is applied to the features set. So, in the dataset case, we apply a moving average filter with 5 neighbours data points to filter the noise, smoothing the data. Subsequently, a normalization step is applied in such a way that the values of minimum and maximum obtained during the training stage were applied on the testing set as follows:

$$\mathbf{F}_{tr_i} = \frac{\mathbf{F}_{tr_i} - \min(\mathbf{F}_{tr})}{\max(\mathbf{F}_{tr}) - \min(\mathbf{F}_{tr})}, \text{ and } \mathbf{F}_{te_i} = \frac{\mathbf{F}_{te_i} - \min(\mathbf{F}_{tr})}{\max(\mathbf{F}_{tr}) - \min(\mathbf{F}_{tr})}, \tag{5}$$

where $\mathbf{F}_{tr}$ is the set of features for training and $\mathbf{F}_{te}$ is the set of features for test; i is an index to describe a set of features in a specific frame; $\max(\cdot)$ and $\min(\cdot)$ are functions to get the global maximum and minimum value of a feature set. In the real-time case, we did not apply the moving average filter because it returns worse results. The normalization step is done in the same way as in the offline tests because we keep the maximum and minimum values of the training set.

3.2 Probabilistic Classification Model

In this work, we adopt an ensemble of classifiers called Dynamic Bayesian Mixture Model (DBMM) proposed in [2] [1]. DBMM uses the concept of mixture models in a

dynamic form in order to combine conditional probability outputs (likelihoods) from different single classifiers, either generative or discriminative models. A weight is assigned to each classifier, according to previous knowledge (learning process), using an uncertainty measure as a confidence level, and can be updated locally during the online classification. The local weight update assigns priority to the base classifier with more confidence along the temporal classification, since they can vary along the different frames. The key motivation of using a fusion model is because we are taking into consideration that an ensemble of classifiers is designed to obtain better performance than any of their individual classifiers, once there is diversity of the single components. Beyond of employing this classification model in an on-the-fly robot-assisted living application, we also compare the activity classification results with different well-known state-of-the-art classification models, such as Naive Bayes Classifier (NBC), Support Vector Machines (SVM) and k-Nearest Neighbours (k-NN). The DBMM general model for each class C is given by:

$$P(C|A) \;=\; \beta \;\times\; \underbrace{P(C^t|C^{t-1})}_{\text{dynamic transitions}} \;\times\; \underbrace{\sum_{i=1}^{n} w_i^t \times P_i(A|C^t),}_{\text{mixture model with dynamic w}} \qquad (6)$$

$$\text{with} \begin{cases} P(C^t|C^{t-1}) \equiv \frac{1}{C} \;\; (\text{uniform}), \, t = 1 \\ P(C^t|C^{t-1}) = P(C^{t-1}|A), \quad t > 1 \end{cases},$$

where $P(C^t \mid C^{t-1})$ is the transition probability distribution among class variables over time, which a class C^t is conditioned to C^{t-1}. This means a non-stationary behavior applied recursively, then reinforcing the classification at time t; $P_i(A|C^t)$ is the posterior result of each i^{th} base classifier at time t, becoming the likelihood in the DBMM model. The weight w_i^t in the model for each base classifier is initially estimated using an entropy-based confidence on the training set (offline), and afterwards ($t > 5$) it is updated as explained in our previous work [1]; $\beta = \frac{1}{\sum_j \left(P(C_j^t|C_j^{t-1}) \times \sum_{i=1}^{n} w_i \times P_i(A|C_j^t) \right)}$ is a normalization factor, ensuring numerical stability once continuous update of belief is done.

Base Classifiers for DBMM In this work, we have used the NBC, SVM and k-NN as base classifiers for the DBMM fusion. The NBC assumes the features are independent from each other given a class, $P(C_i|A) = \alpha P(C_i) \prod_{j=1}^{m} P(A_j|C_i)$. For the linear-kernel multiclass SVM implementation, we adopted the LibSVM package [17], trained according to the 'one-against-one' strategy, and classification outputs were given in terms of probability estimates. A k-NN was also combined into the DBMM fusion. An object is classified by a majority vote of its neighbours, with the object being assigned to the class most common among its k nearest neighbours. The classification outputs of the adopted k-NN were given in terms of probability estimates as well.

4 Robot-Assisted Living Architecture in ROS

The proposed artificial cognitive system was implemented in ROS and comprises three main modules, as shown in Figure 1: classification, navigation and reaction modules.

In order to properly test the system in real scenarios, a mobile robot is used. Therefore, a personal robot endowed with cognitive skills, capable of monitoring the behaviours of a person should be able to autonomously navigate in an indoor environment. The navigation module uses odometry and laser scans from the robot to map the environment and self-localization, randomly navigating, avoiding obstacles. We use the navigation stack available in ROS distributions, more specifically, the *move_base* package to generate an appropriate collision free trajectory. For simultaneous localization and mapping (SLAM) the *hector_slam* package is used. While the robot is navigating, the MS-Kinect sensor is sending RGB-D data to the classification module. Once a skeleton is detected, the robot stops and the feature extraction process starts. Then, classification is done using the DBMM and an activity is recognized. Once the system knows the human activity being performed, the reaction module is in charge to select what the robot should do next. For each human activity, a predefined reaction in a lookup-table was associated, including warnings, questions or changes in navigation (Figure 2). In the event of a person telling the robot to follow him/her, a safe distance of 2.5 meters is maintained. A Kalman filter is used to estimate the trajectory of the person one second ahead in order to avoid collision between the robot and the human. If a collision trajectory is estimated, the robot will step away, in order to the person walk through safely. For the prediction of the human motion, a position model was adopted, where the state includes position $(x(k); y(k))$ of the human target:

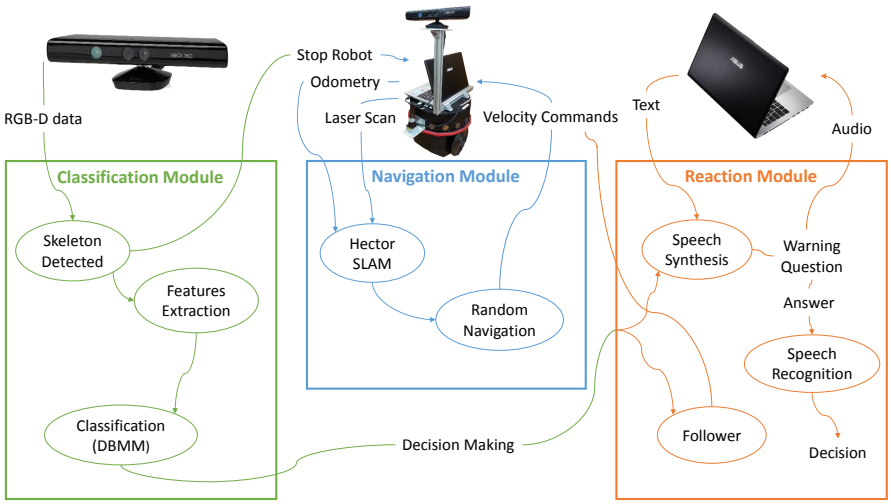


Fig. 1: System overview.

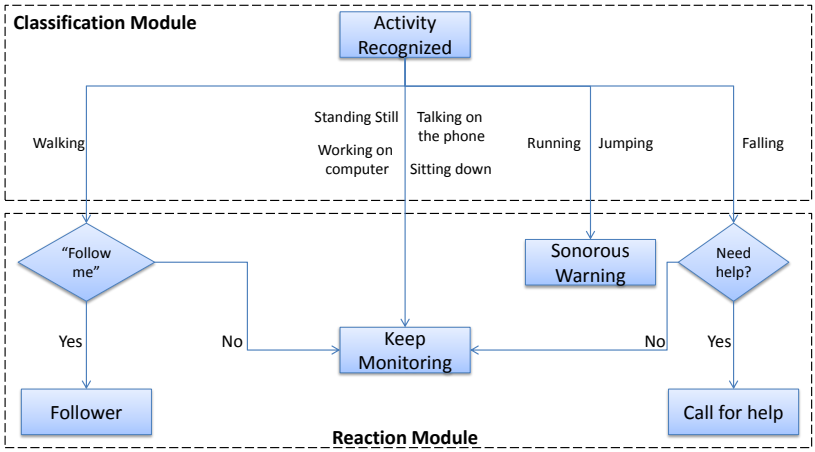


Fig. 2: Decision tree in reaction module.

$$\begin{cases} x(k) = x(k-1) + v_x(k-1) \times \Delta t \\ y(k) = y(k-1) + v_y(k-1) \times \Delta t \end{cases} \tag{7}$$

with $\Delta t = t(k) - t(k-1)$. Using the torso coordinates as measures, it is possible to compute the x velocity v_x and y velocity v_y. For speech synthesis, we use the *sound_play* package that given a text input, it will be synthesized into sound output. For speech recognition, we use the *pocketsphinx* package. This package recognizes a single word or a stream of words from a vocabulary file previously created. In our work, the vocabulary comprises the following words: "no", "yes", "please", "help", "follow", "me". The package can recognize combinations of these words, such as "please help me".

5 Experimental Results

5.1 Performance on collected dataset

A new dataset of daily activities and risk situations, more complete and challenging than the one used in our previous work [1], was collected to train the activity recognition module. This dataset (Figure 3) comprises video sequences of two male subjects and two female subjects performing eight different activities in a living room. The daily activities are: *1-walking, 2-standing still, 3-working on computer, 4-talking on the phone, 5 sitting down*; and the unusual or risk situations are: *6-jumping, 7-falling down, 8- running*. This dataset is a challenging one, once there is significant intra-class variation among different realizations of the same activity. For example, the phone is held with the left or right hand. Another challenging feature is that the activity sequences are registered from different views, i.e., from the front, back, left side, and so on. The classification results are presented in a confusion matrix and with the measures of Accuracy, Precision, Recall of the four tests. The idea is to verify the capacity of generalization of

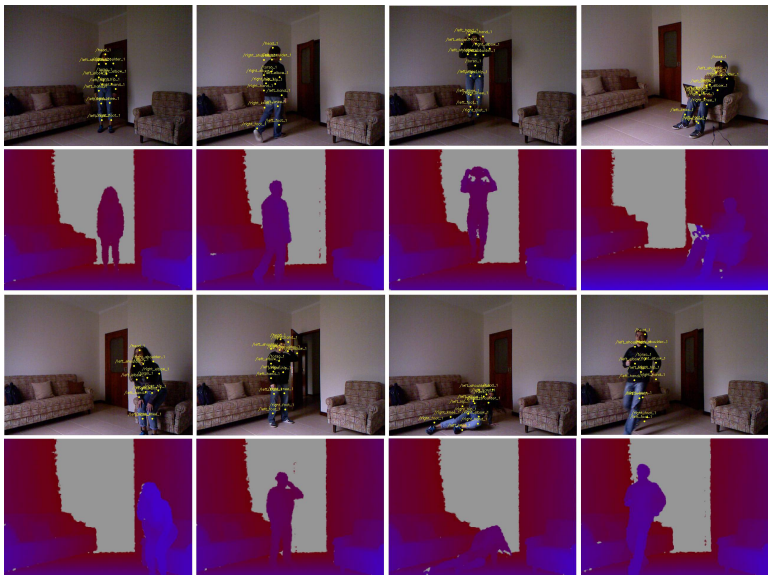


Fig. 3: Few examples of the dataset (RGB with skeleton joints and depth images) which was created to learn some daily and risk situations.

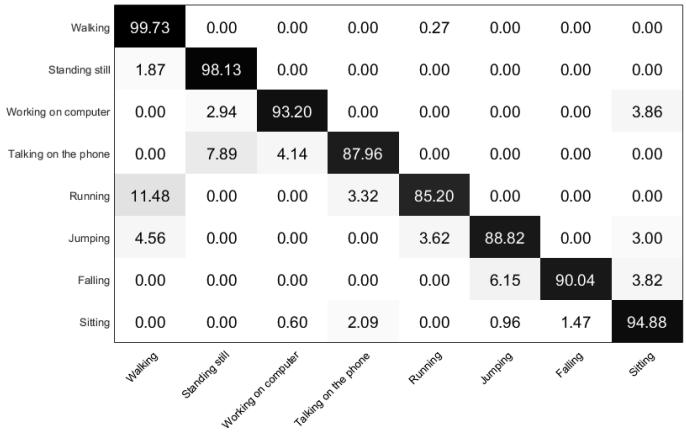


Fig. 4: Confusion matrix obtained from the DBMM classification applied on the dataset

the classifier by using the strategy of ”new person”, i.e., learning from different persons and testing with an unseen person. Figure 4 shows the results in a single confusion matrix. Table 1 shows the performance in terms of Precision (Prec) and Recall (Rec) of this approach for each activity. The results show that using DBMM, improvements in the classification were obtained in comparison with using the base classifiers alone. The overall results attained were: accuracy 93.41%, precision 93.61% and recall 92.25%.

Table 1: Performance on the dataset ("new person"). Results are reported in terms of Precision (Prec) and Recall (Rec).

Activity	DBMM	
	Prec	Rec
walking	89.63%	99.73%
standing still	94.86%	98.13%
working on computer	95.93%	93.20%
talking on the phone	93.64%	87.96%
running	92.81%	85.20%
jumping	92.52%	88.83%
falling down	97.24%	90.04%
sitting down	92.27%	94.88%
Average	**93.61%**	**92.25%**

Table 2: Global results using single classifiers, a simple average ensemble (AV) and the DBMM.

Method	Acc.	Prec.	Rec.
NBC	82.90%	85.79%	82.67%
SVM	88.47%	89.02%	87.62%
k-NN	87.98%	90.09%	87.06%
AV	85.29%	87.74%	84.68%
DBMM	**93.41%**	**93.61%**	**92.25%**

For comparison purposes, Table 2 summarizes the results from single classifiers and an average ensemble compared with DBMM, showing the improvement achieved using the described skeleton-based features. The SVM was trained with *soft margin* (or Cost) parameter set to 1.0, and the *k*-NN was trained using 20 neighbours.

5.2 Performance on-the-fly using a mobile robot

The experimental tests using the proposed approach for a real time application is a little bit different than the experimental tests on the dataset. In this case, the robot will acquire 5 seconds of RGB-D sensor data for features extraction and classification. Only the NBC and SVM were used as base classifiers for the DBMM fusion, because they are enough for obtaining good results, thus, avoiding spending more processing time using other base classifiers. After 5 seconds of frames classification, a final decision is made for activity recognition to trigger a proper robot reaction. The proposed framework is capable of recognizing different activities transitions that happens sequentially in case of a person transit from one activity to another one, e.g., a person that is standing and sequentially pass to a sitting down position and consequentially working on the computer. Figure 5 shows some examples of tests of daily activities and unusual or risk situations that the mobile robot correctly recognized. Three tests were carried out for each activity with three different subjects. One of the subjects was already "seen" in the training, while the rest are "unseen" subjects.

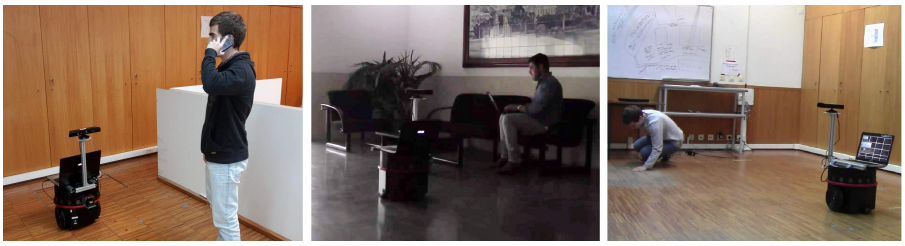


Fig. 5: Shots of tests of activity recognition (‘unseen” person) using a mobile robot.

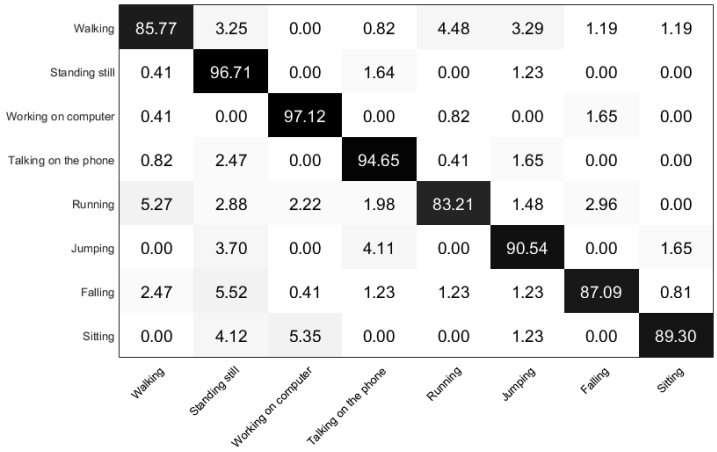


Fig. 6: DBMM on-the-fly classification confidence (average) presented in a confusion matrix

All activities were correctly classified, so that the overall performance of classification is shown in Figure 6. The overall (average) results attained in real-time experiments were: accuracy 90.55%, precision 90.84% and recall 90.55%. Table 3 shows the results in terms of recall of each test for each subject. Looking at the results attained, it is possible to conclude, as expected, that the best performance is achieved for the ”seen” person (subject 1). However, the difference of results between subjects is not very significant, which indicates that the fact of being or not a ”seen” person is not a key factor for the performance of the classification. The most important factor in a real-time application is that in the end, the activity being performed is correctly recognized. Since the robot correctly classified the activity performed, it also successfully reacted accordingly to the situation. Figure 7 shows a sequence of events from an activity that is being recognized (in this case falling) to react according to this activity. First, the skeleton of a person is detected and tracked, initiating the monitoring stage. Then, the person falls on the floor and the robot correctly recognizes the risk situation ”falling”. Detecting such a behaviour, the robot asks if the person needs help. The robot receives an affirmative answer from the person, recognizes the command and immediately calls for help.

Table 3: On-the-fly results in terms of recall for 3 different subjects. One subject seen and two unseen.

		Activity								Overall
	Test	walking	standing still	working on computer	talking on the phone	running	jumping	falling down	sitting down	
Subject 1 (seen)	1	96.30	100	100	59.26	85.19	85.19	85.19	96.30	88.43
	2	96.30	100	100	100	85.19	88.89	95.45	96.30	95.27
	3	92.59	100	92.59	100	85.19	88.89	92.86	96.30	93.55
	Average	95.06	100	97.53	86.42	85.19	87.65	91.17	96.30	**92.42**
Subject 2 (unseen)	1	66.67	100	96.30	100	96.30	81.48	74.07	70.37	85.65
	2	81.48	85.19	96.30	92.59	85.19	92.59	74.07	92.59	87.50
	3	81.48	100	88.89	100	85.19	92.59	95.45	92.59	92-02
	Average	76.54	95.06	93.83	97.53	88.89	88.89	81.20	85.18	**88.39**
Subject 3 (unseen)	1	82.14	96.30	100	100	73.33	96.30	85.19	88.89	90.27
	2	92.86	96.30	100	100	80.00	92.60	100	85.19	93.37
	3	82.14	92.59	100	100	73.33	96.30	81.48	85.19	88.88
	Average	85.71	95.06	100	100	75.55	95.07	88.89	86.42	**90.84**
Overall Average		**85.77**	**96.71**	**97.12**	**94.65**	**83.21**	**90.54**	**87.09**	**89.30**	**90.55**

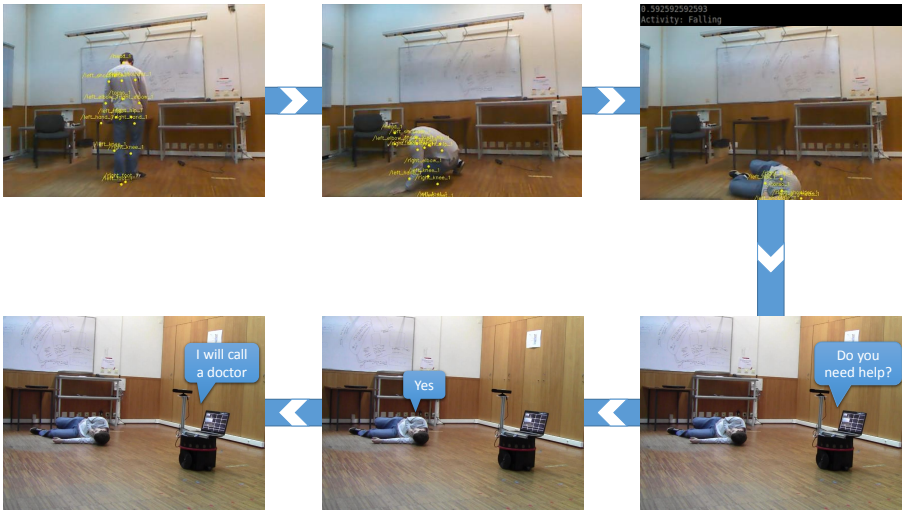


Fig. 7: Sequence of events on detecting a person falling and reacting

6 Conclusions and Future Work

The main contribution of this work is a robotic application for real-time monitoring of daily activities and risk situations in indoor environments. A dynamic probabilistic ensemble of classifiers (DBMM) was used for daily activity recognition using a proposed spatio-temporal 3D skeleton-based features. We collected a dataset to endow a robot to recognize daily activities, and we used this dataset to compare our approach with other

state-of-the-art classifiers. Using our proposed skeleton-based features, we attained relevant results using the DBMM classification, outperforming other single classifiers in terms of overall accuracy, precision and recall measures. More importantly, the experimental tests using a mobile robot presented good performance on the activity classification, allowing the robot to take appropriate actions to assist the human in case of risk situations, showing our framework has good potential for robot-assisted living. Future work will address addition of contextual information, such as "who", "where", "when" in order to fully understand human behaviours, as well as exploitation of our approach with more daily activities, risk situations and robot reactions.

References

1. D. R. Faria, M. Vieira, C. Premebida, and U. Nunes, "Probabilistic human daily activity recognition towards robot-assisted living," in *IEEE RO-MAN'15*, 2015.
2. D. R. Faria, C. Premebida, and U. Nunes, "A probalistic approach for human everyday activities recognition using body motion from RGB-D images," in *IEEE RO-MAN'14, * Kazuo Tanie Award Finalist*, 2014.
3. C. Zhu and W. Sheng, "Realtime human daily activity recognition through fusion of motion and location data," in *IEEE International Conference on Information and Automation*, 2010.
4. ——, "Human daily activity recognition in robot-assisted living using multi-sensor fusion," in *IEEE ICRA'09*, 2009.
5. "Microsoft kinect, Website: https://www.microsoft.com/en-us/kinectforwindows/, accessed on June/2015."
6. "Asus xtion, Website: http://www.asus.com/multimedia/xtion_pro_live/, accessed on June/2015."
7. G. T. Papadopoulos, A. Axenopoulos, and P. Daras, *Real-time Skeleton-tracking-based Human Action Recognition Using Kinect Data*. Springer International Publishing, 2014, ch. 3D and Augmented Reality, pp. 473–483.
8. C. Chen, K. Liu, and N. Kehtarnavaz, "Real-time human action recognition based on depth motion maps," *Journal of Real-Time Image Processing*, 2013.
9. J. Sung, C. Ponce, B. Selman, and A. Saxena, "Unstructured human activity detection from RGBD images," in *ICRA'12*, 2012.
10. L. Xia and J. Aggarwal, "Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera," in *CVPR*, 2013.
11. Y. Zhu, W. Chen, and G. Guo, "Evaluating spatiotemporal interest point features for depth-based action recognition," *Image and Vision Computing*, 2014.
12. S. A. Mehdi, C. Armbrust, J. Koch, and K. Berns, "Methodology for robot mapping and navigation in assisted living environments," in *2nd International Conference on Pervasive Technologies Related to Assistive Environments*, 2009.
13. H. S. Koppula, R. Gupta, and A. Saxena, "Learning human activities and object affordances from RGB-D videos," in *IJRR journal*, 2012.
14. M. Volkhardt, S. Mller, C. Schrter, and H.-M. Gross, "Real-time activity recognition on a mobile companion robot," in *55th Int. Scientific Colloquium*, 2010.
15. V. Arsigny, P. Fillard, X. Pennec, and N. A. 5, "Log-euclidean metrics for fast and simple calculus on diffusion tensors," *Magnetic Resonance in Medicine, 56(2):411–421.*, 2006.
16. K. Guo, "Action recognition using log-covariance matrices of silhouette and optical-flow features," Ph.D. dissertation, Boston University, College of Engineering, 2012.
17. C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM TIST*, 2011, http://www.csie.ntu.edu.tw/ cjlin/libsvm.