ROBOTS IN ASSISTED
LIVING ENVIRONMENTS

UNOBTRUSIVE, EFFICIENT, RELIABLE AND MODULAR
SOLUTIONS FOR INDEPENDENT AGEING

Research Innovation Action

Project Number: 643892 Start Date of Project: 01/04/2015 Duration: 36 months

DELIVERABLE 3.5

ADL and mood recognition methods II

Dissemination Level	**Public**
Due Date of Deliverable	Project Month 24, March 2017
Actual Submission Date	7 April 2017
Work Package	WP3, *Modular conceptual home architecture design and ICT method development for efficient, robust and flexible elder monitoring and caring*
Task	T3.2
Lead Beneficiary	NCSR-D
Contributing Beneficiaries	TWG, S&C
Type	R
Status	Submitted
Version	Final

Project funded by the European Unions Horizon 2020 Research and Innovation Actions

Abstract

This report documents ADL recognition methods and their technical evaluation. The deliverable also comprises the prototype implementations of these methods.

History

Version	Date	Reason	Revised by
01	4 Sep 2016	Document structure	NCSR-D
02	4 Oct 2016	Audio event recognition (Section 3)	NCSR-D
03	26 Mar 2017	Indoor localization using BLE beacons (Section 5)	TWG
04	30 Mar 2017	Activity recognition in the RGB-D signal (Section 2)	NCSR-D
05	31 Mar 2017	Activity recognition using smart home sensors (Section 4)	S&C
06	3 Apr 2017	Final editorial editing	NCSR-D
07	5 Apr 2017	Internal review	RUB
Fin	7 Apr 2017	Addressing internal review comments, final document preparation and submission	NCSR-D

Executive Summary

This report documents ADL recognition methods and their technical evaluation. The deliverable also comprises the prototype implementations of these methods.

These methods implement the components foreseen by the ADL recognition architecture (D3.2 *Conceptual Architecture II*). Development and testing is driven by realistic experimental datasets, which are publicly released. The ADL recognition methods developed are:

- An audio event recognition method that extracts acoustic features and uses them to recognize events in audio streams.
- Machine vision methods that recognize motion and identify the onset and ending of activities such as bed transfer and chair transfer in video sequences. These are then used to measure the duration of bed transfer and chair transfer activities.
- Activity recognition through heuristic rules applies to events detected by home automation sensors. For example, a series events about opening and closing of drawers and cupboards in the kitchen is characteristic of the 'meal preparation' activity.
- Indoor localization through BLE beacons, giving the position of BLE-tagged objects (such as furniture, phones, etc.). This can be useful in fusion with other modalities, as it improves overall scene understanding. For example, tagging and localizing chairs (and, in general, movable furniture) can help infer activities by comparing furniture location against the 'obstacle blobs' seen by the robot's sensors.

Regarding machine vision, the work in this deliverable complements the motion detection tracking methods in D3.4 with methods for classifying motion trajectories as specific activities. To achieve this, we empirically compared a battery of computer vision techniques on a RADIO-prepared dataset of activities of daily living. Based on this comparison, we selected the a set of features that is sufficiently accurate and robust, while it can also be efficiently calculated allowing run-time operation.

The audio event recognition system records and segments audio and then processes it to extract all the acoustic features needed by event classifiers. Acoustic classifiers are reliable and performant, but are sensitive to the acoustic conditions prevalent in each environment. For this reason, RADIO has developed a calibration tool that collects sounds from each new environment and uses them to adapt classifiers.

Home automation-based activity recognition uses bottom-up multi-level reasoning, based on rules that use elementary sensor events to infer complex activities. The rule engine finds facts in the sensor data and matches them against the rules.

Finally, indoors positioning is based on infrastructure-based localization, where the indoors space is equipped with beacons that have known location. The moving device uses information provided by these beacons to estimate its location.

Abbreviations and Acronyms

ADL	Activities of Daily Living
BLE	Bluetooth Low Energy, a wireless network technology intended to provide considerably reduced power consumption and cost for a communication range similar to classic Bluetooth.
MFCC	Mel Frequency Cepstral Coefficients, a cepstral representation of an audio stream
MQTT	Message Queuing Telemetry Transport is a standard publish-subscribe-based messaging protocol
ROS	Robot Operating System, the robotics software framework assumed as the basis for development in RADIO
RSSI	Received signal strength indicator, a measurement of the power present in a received radio signal.

CONTENTS

Contents iv

List of Figures v

List of Tables vi

1 Introduction 1
- 1.1 Purpose and Scope . . . 1
- 1.2 Approach . . . 1
- 1.3 Relation to other Work Packages and Deliverables . . . 2

2 Activity Recognition in RGB 3
- 2.1 Feature extraction . . . 3
- 2.2 Activity classification . . . 4
  - 2.2.1 Dataset and classes . . . 4
  - 2.2.2 Classifiers . . . 6
  - 2.2.3 Experimental Evaluation . . . 7

3 Acoustic Activity Recognition 14
- 3.1 Overview . . . 14
  - 3.1.1 Low level audio events . . . 14
- 3.2 Human activity recognition . . . 15
- 3.3 Experiments . . . 16
  - 3.3.1 Datasets . . . 17
  - 3.3.2 Experimental Results . . . 17

4 Activity Recognition using Home Automation Sensors 20
- 4.1 Meal preparation . . . 20
- 4.2 Watching TV . . . 21
- 4.3 Going out . . . 21
- 4.4 Getting up from bed . . . 21

5 Indoors Localization using BLE Beacons 22
- 5.1 Background . . . 22
  - 5.1.1 Trilateration based on RSSI . . . 22
  - 5.1.2 Fingerprinting . . . 23
  - 5.1.3 Proximity-based . . . 23
  - 5.1.4 Using future BLE 5.x . . . 24
- 5.2 RADIO Absolute Localization Methods . . . 24
- 5.3 RADIO Relative Localization Methods . . . 26
- 5.4 Smart Home – BLE Gateway Cooperative Isolated Methods . . . 27
  - 5.4.1 Identification of furniture close to TV through relative localization and interface with the main controller . . . 27
- 5.5 Robot Tracking through RADIO Map Convergence . . . 29
- 5.6 Security through Theft Detection Methods . . . 29

References 30

LIST OF FIGURES

1 Dependencies between this deliverable and other deliverables. . . . 2
2 Activity classification diagram . . . 5
3 The RADIO lab space . . . 6
4 Screenshots captured from the RADIO Dataset . . . 7
5 Different screen-shots captured from the Barcelona Dataset . . . 8
6 The RoboMAE multimodal annotation tool. . . . 9
7 Conceptual architecture of the high-level audio event recognition scheme . . . 14
8 RSSI based trilateration . . . 22
9 Fingerprinting based localization . . . 23
10 Proximity based localization . . . 24
11 Angle of Arrival/Departure based localization . . . 24
12 RADIO localization system . . . 25
13 Indicative network constellation that delivers relative localization services . . . 28
14 The FHAG trials area map as mapped by the RADIO robot. . . . 29
15 Antitheft System Architecture . . . 30

LIST OF TABLES

1 List of prototypes of the methods described in D3.5 1
2 Evaluation on different amounts of training samples per class, training on B1 and testing on B1 and B2 10
3 Results of GB an ET classifiers when trained on B2 and tested on B1 and B2 12
4 Average results of the two best methods 13
5 Average processing time 13
6 Average Confusion Matrix of the GB algorithm 13
7 Adopted short-term audio features 15
8 Audio segment classification results 18
9 Comparison between classification methods and feature sets for the high-level activity recognition task. Performance is quantified based on the F1 measure 18
10 High-level event recognition based on all meta-features for the best classifier (SVM) . . 19
11 Path Loss Exponents for Different Environments 25

1 INTRODUCTION

1.1 Purpose and Scope

This report documents the sensor data analysis methods developed during M16-M24 of Task 3.2. These methods recognize the *Activities of Daily Living (ADL)* identified in D2.7 *Guidelines for balancing between medical requirements and obtrusiveness II* as the objective for ADL recognition for the second phase of method development.

This document reports on the methods developed in Task 3.2 during M16-M24 and their technical evaluation. The prototype implementations of these methods (also developed in Task 3.2) are available as public software repositories (Table 1).

1.2 Approach

This deliverable is prepared in Task 3.2, *ADL and emotion recognition method development*. This task adapts or develops sensor data analysis methods that implement the ADL recognition components foreseen in the conceptual architecture.

The development of these methods started with a review of applicable machine perception methods, leading to the second iteration of the system list of components and architecture (D3.2 *Conceptual architecture II*). We then proceeded by assuming this background and the previously developed RADIO methods (D3.4) as a starting point for the development of the final RADIO ADL recognition stack. Development and technical validation was based on experimental datasets prepared by recording healthy adults (outside the target group) perform ADLs at the NCSR-D *Roboskel Lab* and at the TWG *AAL House*. The technical validation results reported in this deliverable are based on these datasets, and these datasets have also been released publicly released.

Table 1: List of prototypes of the methods described in D3.5

	Method	Repository
Sect. 2: RGB-D	Bed and chair transfer	`https://github.com/RADIO-PROJECT-EU/ros_visual`
Sect. 3: audio	Talking, watching TV, listening to music, housework	`https://github.com/radio-project-eu/AUROS`
Sect. 4: smart home sensors	Meal preparation, watching TV, going out	Proprietary
Sect. 5: BLE beacons	Base localization	`https://github.com/radio-project-eu/relative_multihop_localization`
	Location of robot and BLE-tagged furniture	`https://github.com/radio-project-eu/indoor_relative_localization`
	Co-registration of BLE map with SLAM map	`https://github.com/radio-project-eu/map_convergence`
Datasets	Bed and chair transfer, walking, talking, watching TV	`http://hdl.handle.net/21.15101/HOME`

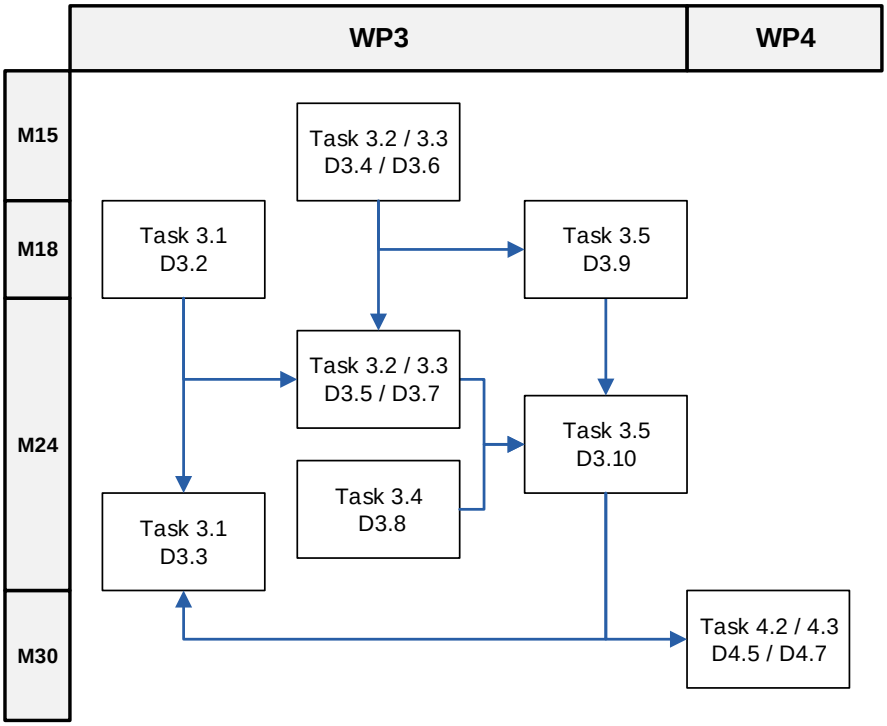

Figure 1: Dependencies between this deliverable and other deliverables.

The methods of the RADIO recognition stack will now be tested at FHAG and FZ during the first iteration of the pilot studies (D6.7 *Pilot report I*). The outcomes of the first pilot study will drive inter-component fusion work (on-going in WP4), to prepare the final prototype that will be evaluated by the final pilot studies (D6.8 *Pilot report II*).

1.3 Relation to other Work Packages and Deliverables

This document is part of a cluster of closely related deliverables. Work around is organized as follows:

- *Conceptual architecture II (D3.2):* Report, used to guide development work by setting technical requirements on the ADLs that need to be recognized and defining the design and interconnections of the recognition components.
- *ADL and Mood Recognition Methods* (D3.4 and this document, D3.5), *Network Robustness and Efficiency Methods* (D3.6 and D3.7), and *Social Network Analysis Component* (D3.8): Reports, documenting the methods developed in order to recognize the activities and mood of the primary users. These reports are complemented by their open source prototypes found at the repositories indicated in each report and also collected in the *Conceptual Architecture* documents.
- *Integrated Data Analysis System II:* Software, integrating the prototypes above into a coherent system.
- *Conceptual Architecture III (D3.3):* Report, documenting the final design and interconnections of the recognition components after changes and adjustments carried out during integration work.

These dependencies are also graphically depicted in Figure 1.

2 ACTIVITY RECOGNITION IN RGB

2.1 Feature extraction

In order to capture and classify the activity performed by a tracked person we had to decide upon a set of visual features that were sufficient to describe and characterize the subject's posture, at any given time. Towards that direction we performed various experiments using both state-of-the-art and traditional computer vision techniques. Our decision upon the final implementation and architecture was made by taking into consideration three major target-expectations, which needed to be optimised in parallel. In particular we focused on:

- Maximize system accuracy in terms of activity classification
- Maximize system robustness in terms of how it's efficiency is affected when the recording conditions change (ie. background environment, point of view, lighting conditions etc.)
- Minimize processing time (how fast the available hardware can process and classify a given frame)

After exhaustive experimentation on different approaches we decided to work on a set of hand-crafted features, which were sufficiently accurate and robust, very fast to calculate and also able to take into account and hence, smooth, the additional error inherited by the tracking mechanism in terms of bounding-box accuracy. The proposed features are being extracted directly from the frame region marked by the tracking system (i.e., the bounding boxes, see D3.4, Section 3) and aim to model its physical properties (height and width), the evolution of those properties over time and also visual characteristics regarding the spatio-temporal changes of two consecutive bounding-boxes with the same *id* in terms of visual content. In more detail, given two bounding-boxes $bbox_t$ and $bbox_{t-1}$, with the same *id*, extracted from two consecutive frames at times t and $t-1$ respectively, the set of selected features extracted for the bounding box at time t consists of:

1. **Box-Ratio:** The ratio of the height by width of a bounding-box at time t

$$BR = \frac{Height_t}{Width_t}$$

2. **Box-Ratio-Delta:** The difference between the box-ratios of $bbox_t$ and $bbox_{t-1}$, normalised by the elapsed time between $frame_t$ and $frame_{t-1}$.

$$\delta BR = BoxRatio_t - BoxRatio_{t-1}$$

3. **Normalized Projected Velocity:** The distance between the center points of $bbox_t$ and $bbox_{t-1}$, normalised by the area of $bbox_t$ and the elapsed time between $frame_t$ and $frame_{t-1}$.

$$V_{x,y} = \frac{d(x,y)}{A \cdot dt}$$

4. **Normalized Projected Acceleration:** The difference of the Normalized Projected Velocity feature between $bbox_t$ and $bbox_{t-1}$, normalised by the elapsed time between $frame_t$ and $frame_{t-1}$.

$$\alpha(x,y) = \frac{dV(x,y)}{dt}$$

5. **Horizontal Normalized Velocity:** The difference between the top-left corner points of $bbox_t$ and $bbox_{t-1}$ in the horizontal axis, normalised by the area of $bbox_t$ and the elapsed time between $frame_t$ and $frame_{t-1}$.

$$V_x = \frac{dx}{A \cdot dt}$$

6. **Horizontal Normalized Acceleration:** The difference of the Horizontal Normalized Velocity feature between $bbox_t$ and $bbox_{t-1}$, normalised by the elapsed time between $frame_t$ and $frame_{t-1}$.

$$\alpha_x = \frac{dV_x}{dt}$$

7. **Vertical Normalized Velocity:** The difference between the top-left corner points of $bbox_t$ and $bbox_{t-1}$, in the vertical axis, normalised by the area of $bbox_t$ and the elapsed time between $frame_t$ and $frame_{t-1}$.

$$V_y = \frac{dy}{A \cdot dt}$$

8. **Vertical Normalized Acceleration:** The difference of the Vertical Normalized Velocity between $bbox_t$ and $bbox_{t-1}$, in the vertical axis, normalised by the elapsed time between between $frame_t$ and $frame_{t-1}$.

$$\alpha_y = \frac{dV_y}{dt}$$

9. **Projected Normalized Height:** The vertical coordinate of the top-left corner point of $bbox_t$, normalized by the area of $bbox_t$

$$H = \frac{y_t}{A}$$

10. **Relative Velocity from the sensor:** The difference between the distance of the subject from the sensor, in $bbox_t$ and $bbox_{t-1}$, normalized by the time elapsed between $frame_t$ and $frame_{t-1}$. Distance S from the sensor is calculated using the depth camera.

$$V_S = \frac{d(S_t, S_{t-1})}{dt}$$

11. **Magnitude of the Relative Velocity from the sensor:**

$$|V_S|$$

12. **Depth-Std:** The standard deviation of all the depth values in a bounding-box

$$std(Depth_t)$$

, where $Depth_t$ the set of depth values in $bbox_t$

2.2 Activity classification

Using the set of features described in the above section, our goal was to classify human activities in terms of frame-wise posture and activity performed. We experimented with different classification approaches using datasets specifically collected to realistically emulate operational conditions as closely as possible. We will now describe in detail the different datasets along with the set of the target classification labels, the methods and algorithms used for activity classification and finally we present and discuss our experimental results in terms of classification performance and computational cost using the different approaches. In Figure 2 we illustrate the overall architecture of the Activity Recognition System.

2.2.1 Dataset and classes

RADIO Dataset In order to increase the complexity of the dataset and create more realistic scenarios that could represent the problem more naturally we developed a second dataset, which worked as a core for building the system's architecture. The dataset consists of 240 video recordings including the depth and RGB information streams. In total 20 people participated in the recordings performing 6 different scenarios. The recordings were made under two different lighting conditions (natural and artificial light)

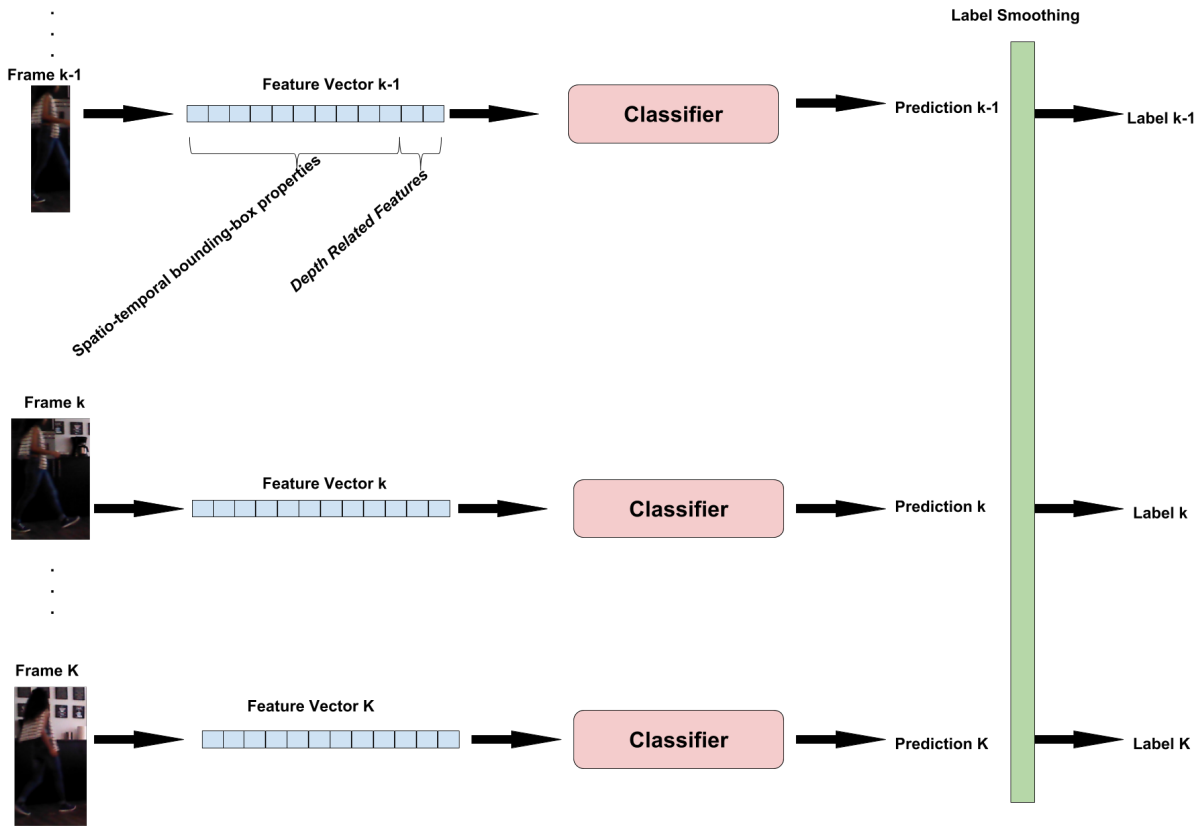

Figure 2: Activity classification diagram. Each feature vector consists of 12 features. The first ten features describe the physical spatio-temporal properties of the bounding-box while the last two features include depth related information extracted from the current frame. Spatio-temporal features of any given image take into account the physical properties of the current bounding-box as well as the related features of the preceding box.

and from two different points of view. The selected scenarios were specifically designed in collaboration with the medical experts in order to represent a wide range of daily living activities, given the RADIO assumptions. The dataset consists of 6 main activity classes, namely *Stand*, *Sit*, *Walk*, *Lie*, *from/to Stand* and *from/to Lie*. However the scenarios were designed such that higher-level activities (preparing-snack, eating, watching-TV, cocking etc.) could be annotated as well. However, for the purposes of this deliverable the data was annotated only on the low level class labels. For recording purposes we used an XTION-RGBD video sensor (https://www.asus.com/us/3D-Sensor/Xtion_PRO_LIVE/). In Figure-3 we show a sketch-up of the NCSR-D lab space that was used prepared for the purposes of the project. In Figure-4 we show different screenshots of the *RADIO Dataset* showing the different recording conditions, in terms of subject variability, lighting conditions and sensor setup.

Barcelona Dataset Final experimentation took place on a dataset captured at FHAG. This dataset is the most realistic one in terms of environmental conditions and also in terms of subjects as it consists of 29 recordings captured from four real residents of the nursing structure. The dataset consists again of the same six main activity classes, "*Stand*, "*Sit*, "*Walk*, "*Lie*", "*from/to Stand*" and "*from/to Lie*", and was captured in the facilities of the hospital. An XTION RGBD camera sensor was used again for the recordings, which took place in two different rooms; a hospital room and an indoor corridor, that connects the rooms to a nearby kitchen. Subjects were asked to act naturally and perform different activities in each room. In Figure 5 we show a set of screen-shots captured from the Barcelona Dataset, showing different aspects of the data.

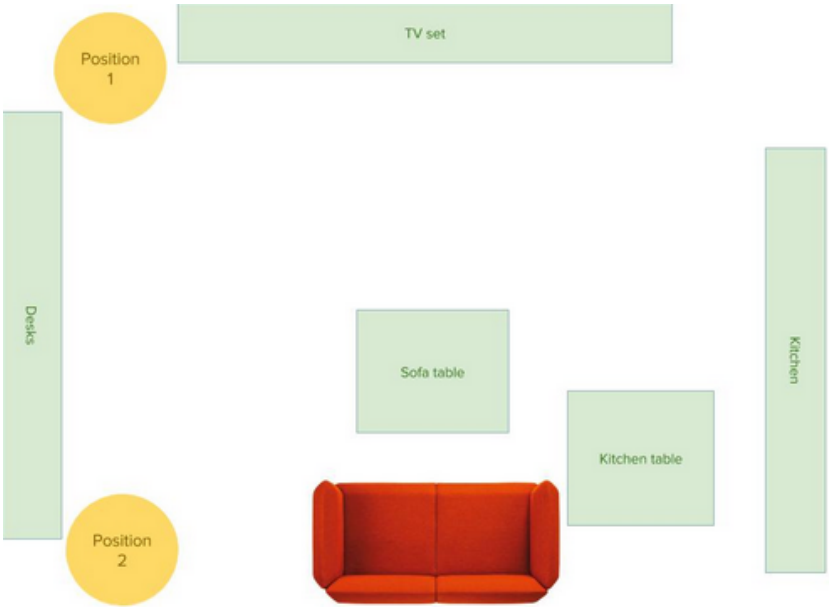

Figure 3: The NCSR-D lab space used for testing the RADIO system and also for collecting the RADIO Dataset. Position 1 and Position 2 show the positions where the robot was placed to get different points of view.

Annotation Tool To annotate the RADIO dataset, we adapted the RoboMAE open-source multimedia annotation tool to annotating activities of daily living.[1] The tool is used to tag a single or combinations of the following modalities: RGB and/or depth video, multi/single array audio stream and laser scans. A screenshot of the tool can be seen in Figure 6. The tool was used to annotate both the *RADIO* and the *Barcelona* datasets.

2.2.2 Classifiers

Our initial approaches were focused on activity classification using state-of-the-art Computer-Vision techniques, and in particular a deep-learning based approach using *Convolutional Neural Networks (CNNs)*. Deep Learning is a sub-field of machine learning concerned with algorithms inspired by the structure and function of the brain called artificial neural networks. CNNs are a specific neural network architecture very similar to ordinary Neural Networks (Hornik et al., 1989). CNNs are made up of neurons that have learnable weights and biases. Each neuron receives some inputs, performs a dot product and optionally also applies a non-linear function. CNN architectures make the explicit assumption that the inputs are images, which allows us to encode certain properties into the architecture. These then make the forward function more efficient to implement and vastly reduce the amount of parameters in the network. Despite the very promising early results (Papakostas et al., 2016), the proposed technique suffered from very high computational costs (classification of a single image was made in the order of seconds), which made it unfeasible for real time analysis. Thus, we extended our experiments by evaluating additional classifiers that could perform well on the aforementioned features and were able to perform the activity classification in a significantly shorter time frame (in the order of nanoseconds). The classifiers that we focused on, are traditional methods in the field of machine learning, that have been proven to be very efficient in many applications related to ours (Pirsiavash and Ramanan, 2012; Gall et al., 2011; Turaga et al., 2008). In particular we evaluated the following classification methods:

- *Linear SVMs and SVMs using the RBF kernel:* A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes new examples in one of the target labels, which in our case is a label describing the human activity (Platt, 1999; Hsu et al., 2003).

[1] `https://github.com/RADIO-PROJECT-EU/RoboMAE`

Figure 4: Screenshots captured from the RADIO Dataset. The first six images were captured from Position-1 and the last six from Position-2 (see Figure 3).

- *Random Forests:* Random Forests is an ensemble learning method used for classification and regression that uses a multitude of decision tries (Ho, 1995; Pal, 2005).
- *Gradient boosting:* Gradient boosting is an ensembling approach interpreted as an optimization task, widely used both in classification and regression tasks (Breiman, 1997; Friedman, 2002).
- *Extremely Randomized Trees (or Extra trees):* Extra trees are a modern classification tree-ensemble approach that is based on strong randomisation of both attribute and cut-point choice while splitting the tree node (Geurts et al., 2006), and has been adopted in various computer vision and data mining applications.

2.2.3 Experimental Evaluation

For evaluation purposes we tried to optimise our system based on the Barcelona dataset (cf. Section 2.2.1). To do so, we split the available recordings into two sub-datasets, *B1* and *B2*. The B1 dataset consists of ten recordings, and the B2 of nine. The total duration of the recordings is almost equal across the two datasets. Moreover, the number of samples at each class are also balanced between B1 and B2.

Figure 5: Different screen-shots captured from the 'Barcelona Dataset' showing different subjects performing daily-living activities at FHAG.

In Table 2 we demonstrate detailed evaluation results, for all five classifiers described in 2.2.2, when trained on B1 and tested on B1 and B2. A smaller amount of training samples leads to more uniformly distributed samples across the target classes, however it limits the available information regarding the problem and fails to effectively represent the subjects postural state. On the other hand, very high thresholds lead to model overfitting due to the very large number of samples in classes 'walk' and 'sit' compared to the other classes. Highlighted lines correspond to the two experiments with the higher average F1 measure when tested on B2 dataset. Overall, the Gradient Boosting (GB) and the Extra Trees (ET) algorithms, outperformed the rest in most scenarios. Random Forests (RF) performance was also acceptable in most cases, however GB and ET seem to be slightly more robust to changes and less dependent on the nature and the amount of the samples. Finally SVM and SVM-RBF algorithms failed in all cases to compete with the rest of the methods.

In Table 3 we show the evaluations for the best two classifiers when trained on B2 and tested again in both sub-datasets. In Table 4 we show the average results for the best classifiers and in Table 5 we compare the two best methods in terms of computational complexity.

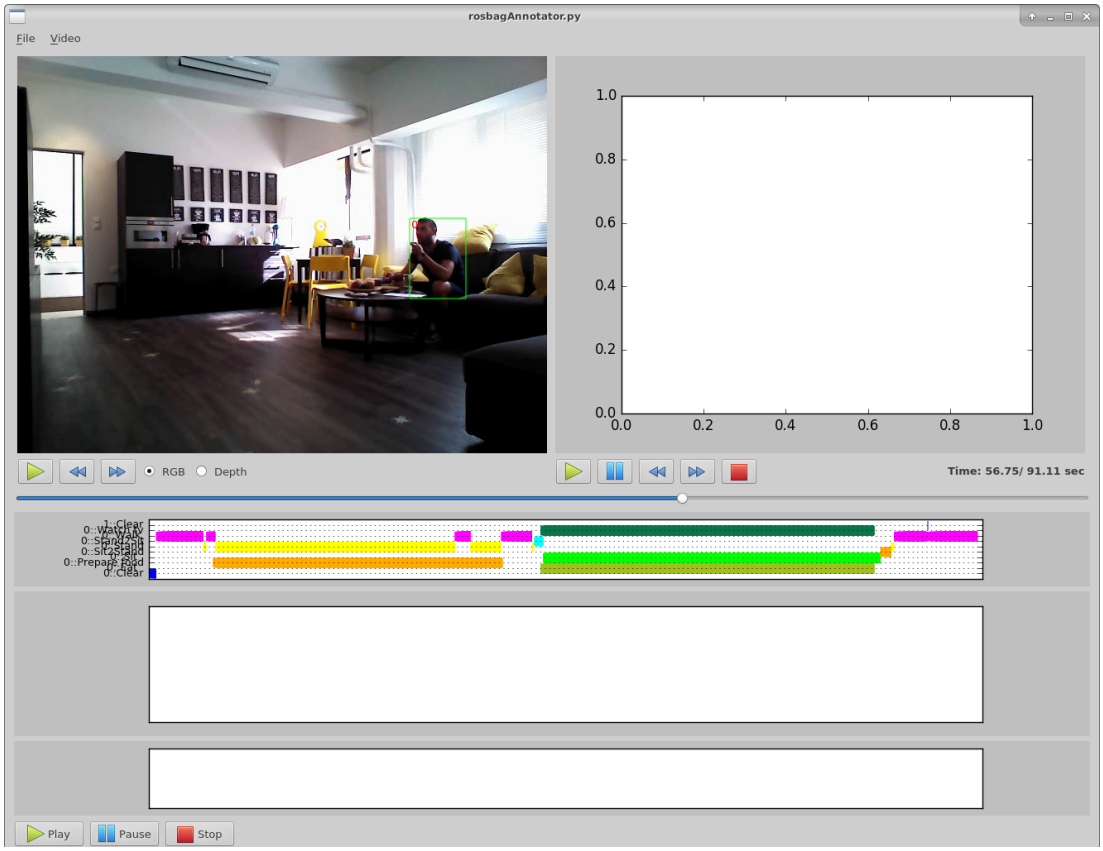

Figure 6: The RoboMAE multimodal annotation tool.

Data Balancing As mentioned in Section 2.2.1, in all datasets the distribution of the available training samples was very unbalanced across the six different target classes, since classes like 'walk' or 'sit' were very highly skewed. That fact, in most cases, leads to highly overfitted models, which were unable to perform well on new samples. To avoid such problems we tried randomly sub-sampling the number of training samples for each class, making the samples more uniformly distributed across the classes during training. It is important to note that these classification experiments annotate the moving objects identified and tracked using the method described in D3.4, Section 3.

As explained in Table 2, the smaller the amount of training samples the more uniformly distributed are the samples across the target classes. On the other hand, if we pick a high threshold as a maximum number of samples per class, models tend to suffer from overfitting problems, due to highly imbalanced classes. Gradient Boosting and Extra Trees significantly outperformed the other approaches in most scenarios. Both techniques seem to be more invariant to changes in the number of training samples per class and also achieve higher performance scores.

In Table 3 we give the results of Gradient Boosting and Extra-Trees algorithms when trained on B2 dataset and tested on both B1 and B2. All results are slightly worse compared to the results shown in Table 2, which indicates that the training samples included in B2 are less efficient on representing all the possible states of our problem compared to the samples included in B1. We partially expected such a behaviour since B2 consists of one recording less than the B1 dataset and thus, contains less training samples as well. Highlighted lines correspond to the two experiments with the maximum average F1 value when tested on the B1 dataset.

In Table 4 we show the average results for the best two experiments. Here, best is defined as the method that achieved maximum average F1 measure over all the experiments when trained in one dataset and tested on the other.

As shown in Table 4 Extra-Trees seem to be slightly more robust as they achieve higher Maximum F1

Table 2: Evaluation on different amounts of training samples per class, training on B1 and testing on B1 and B2

Test Dataset		**B1**			**B2**		
Max samples per class	Classifier	Min F1	Max F1	Avg F1	Min F1	Max F1	Avg F1
100	SVM-RBF	0.347	0.741	0.538	0.817	0.838	0.542
	SVM	0.365	0.785	0.602	0.196	0.881	0.590
	GB	0.649	0.910	0.805	0.257	0.889	0.697
	ET	0.524	0.868	0.759	0.269	0.853	0.652
	RDFR	0.943	0.998	0.985	0.299	0.864	0.676
200	SVM-RBF	0.403	0.837	0.620	0.200	0.869	0.583
	SVM	0.370	0.804	0.621	0.197	0.872	0.607
	GB	0.626	0.971	0.844	0.319	0.902	0.707
	ET	0.668	0.937	0.871	0.268	0.898	0.712
	RDFR	0.604	0.898	0.825	0.269	0.882	0.671
300	SVM-RBF	0.401	0.809	0.670	0.219	0.887	0.635
	SVM	0.355	0.897	0.639	0.193	0.878	0.609
	GB	0.752	0.958	0.891	0.302	0.899	0.713
	ET	0.668	0.958	0.871	0.264	0.911	0.716
	RDFR	0.655	0.927	0.872	0.307	0.893	0.696
500	SVM-RBF	0.436	0.833	0.703	0.232	0.887	0.667
	SVM	0.381	0.897	0.663	0.205	0.869	0.603
	GB	0.822	0.998	0.931	0.278	0.900	0.712
	ET	0.764	0.984	0.917	0.256	0.899	0.706
	RDFR	0.744	0.998	0.913	0.311	0.912	0.694
750	SVM-RBF	0.462	0.888	0.757	0.246	0.891	0.660
	SVM	0.404	0.873	0.691	0.253	0.889	0.657
	GB	0.847	0.998	0.940	0.275	0.906	0.717
	ET	0.856	0.998	0.952	0.263	0.912	0.728
	RDFR	0.818	0.998	0.942	0.293	0.905	0.676
1000	SVM-RBF	0.480	0.905	0.771	0.237	0.897	0.676
	SVM	0.308	0.877	0.697	0.252	0.881	0.662
	ET	0.935	0.998	0.977	0.278	0.924	0.735
	RDFR	0.899	0.998	0.967	0.258	0.910	0.666

Table continues on next page.

Table continues from previous page.

Test Dataset		**B1**			**B2**		
Max samples per class	Classifier	Min F1	Max F1	Avg F1	Min F1	Max F1	Avg F1
1250	SVM-RBF	0.355	0.927	0.758	0.164	0.904	0.679
	SVM	0.306	0.87	0.704	0.194	0.882	0.669
	GB	0.925	0.998	0.973	0.281	0.91	0.723
	ET	5	195	25	50	0.919	0.729
	RDFR	0.932	0.998	0.979	0.264	0.916	0.67
1500	SVM-RBF	0.34	0.928	0.746	0.083	0.913	0.658
	SVM	0.293	0.875	0.694	0.132	0.873	0.657
	GB	0.942	0.999	0.983	0.261	0.915	0.721
	ET	0.966	0.998	0.99	0.195	0.923	0.719
	RDFR	0.943	0.998	0.985	0.257	0.909	0.684
1750	SVM-RBF	0.340	0.930	0.753	0.083	0.914	0.657
	SVM	0.258	0.884	0.684	0.081	0.880	0.650
	GB	0.961	0.999	0.987	0.261	0.916	0.726
	ET	0.969	0.999	0.992	0.214	0.920	0.729
	RDFR	0.953	1	0.989	0.241	0.913	0.669
2000	SVM-RBF	0.343	0.936	0.763	0.084	0.914	0.651
	SVM	0.009	0.885	0.634	0	0.888	0.637
	GB	0.967	1	0.991	0.28	0.916	0.73
	ET	0.973	1	0.994	0.228	0.922	0.712
	RDFR	0.966	1	0.991	0.274	0.903	0.641
3000	SVM-RBF	0.269	0.929	0.713	0.074	0.915	0.632
	SVM	0	0.887	0.628	0	0.892	0.634
	GB	0.952	1	0.985	0.259	0.917	0.722
	ET	0.995	1	0.999	0.168	0.931	0.711
	RDFR	0.997	1	0.999	0.229	0.911	0.68
4000	SVM-RBF	0.886	0.926	0.708	0.07	0.916	0.631
	SVM	0	0.886	0.625	0	0.889	0.628
	GB	0.888	0.998	0.961	0.238	0.912	0.715
	ET	0.998	1	1	0.16	0.93	0.715
	RDFR	0.997	1	0.999	0.217	0.915	0.65

Table 3: Results of GB an ET classifiers when trained on B2 and tested on B1 and B2

Test Dataset		**B1**			**B2**		
Max samples per class	Classifier	Min F1	Max F1	Avg F1	Min F1	Max F1	Avg F1
100	GB	0.354	0.865	0.642	0.598	0.921	0.795
	ET	0.348	0.878	0.655	0.589	0.913	0.781
200	GB	0.275	0.909	0.642	0.737	0.941	0.875
	ET	0.3	0.898	0.668	0.703	0.938	0.853
300	GB	0.249	0.905	0.672	0.737	0.949	0.886
	ET	0.292	0.910	0.676	0.772	0.979	0.901
500	GB	0.17	0.912	0.663	0.858	0.998	0.946
	ET	0.199	0.915	0.635	0.849	0.998	0.934
750	GB	0.112	0.912	0.664	0.887	0.998	0.952
	ET	0.169	0.919	0.648	0.92	0.998	0.962
1000	GB	0.081	0.913	0.654	0.937	0.998	0.97
	ET	0.204	0.926	0.659	0.944	0.988	0.974
1250	GB	0.078	0.916	0.668	0.938	0.998	0.975
	ET	0.064	0.918	0.638	0.949	0.998	0.982
1500	GB	0.074	0.913	0.66	0.95	0.995	0.984
	ET	0.1	0.918	0.632	0.966	0.998	0.986
1750	GB	0.069	0.915	0.663	0.937	0.998	0.989
	ET	0.199	0.926	0.668	0.900	0.998	0.989
2000	GB	0.072	0.916	0.660	0.210	0.998	0.973
	ET	0.059	0.919	0.624	0.982	0.998	0.993
3000	GB	0.064	0.913	0.668	0.962	0.998	0.986
	ET	0.054	0.924	0.631	0.998	1.000	1.000
4000	GB	0.083	0.921	0.666	0.983	1.000	0.995
	ET	0.056	0.921	0.626	0.980	1.000	1.000

value (which corresponds to the "*walk*" class) and higher Minimum F1 value (which corresponds to the "*stand*" class). However, as shown in Table 5, Extra-Trees have significantly higher computational cost, which in terms of execution time translates to an order of magnitude. Thus, and since both methods are almost identical in terms of average F1, *we decided to proceed with the Gradient Boosting algorithm.*

Finally, in Table 6 we illustrate the average confusion matrix of the Gradient Boosting method, when trained on one dataset and tested on another.

As shown in Table 6 the most challenging class is the 'stand' class, where most samples are labeled as 'walk'. That is because, in all recordings there are very few samples annotated as 'stand' and the physical and spatio-temporal properties of a bounding-box when standing are very similar to those when walking. In addition, a slight confusion is observed between the transitions of 'sit' (*from/to Sit*) and the actual 'sit' class and between the transitions of 'lie' (*from/to Lie*) and the 'lie' class. That is also expected as the physical properties of the bounding-boxes are again very similar between the pairs of classes. To decrease that error we perform an online error smoothing, which slightly improves detection rate.

Table 4: Average results of the two methods that achieved the highest average F1 measure when trained in one dataset and tested on the other.

Max samples per class	Classifier	Min F1	Max F1	Avg F1
3000	GB	0.162	0.915	0.695
1750	ET	0.207	0.923	0.699

Table 5: Average processing time needed to classify an unknown sample to one of the six labels for the two most accurate classifiers. GB is significantly faster compared to ET.

Classifier	Average Processing time per frame
GB	0.334 msec
ET	66.206 msec

Table 6: Average Confusion Matrix of the GB algorithm when trained in one dataset and tested on another; using a maximum number of 3000 training samples per class, as this was the optimal setting for (see Table 4)

	Sit	**Stand**	**Walk**	**Lie**	**from/to Stand**	**from/to Lie**
Sit	0.283	0.007	0.002	0.000	0.017	0.007
Stand	0.005	0.010	0.040	0.000	0.008	0.000
Walk	0.001	0.024	0.419	0.000	0.010	0.001
Lie	0.000	0.000	0.000	0.005	0.000	0.002
from/to Stand	0.021	0.005	0.008	0.000	0.069	0.001
from/to Lie	0.011	0.001	0.000	0.002	0.001	0.047
Recall	0.893	0.173	0.918	0.719	0.665	0.772
Precision	0.885	0.192	0.897	0.804	0.664	0.844
F1	0.888	0.176	0.906	0.732	0.664	0.802

3 Acoustic Activity Recognition

3.1 Overview

The overwhelming majority of research efforts in robot perception for scene understanding focuses on computer vision, with audio processing typically aiming at speech understanding and human-robot interaction. However, *non-verbal* audio analysis can also play an important role in scene understanding, as extracting attributes of the environment that are hardly or not at all recognizable by visual sensors. Even in cases that visual information is sufficient for recognizing events and objects, machine hearing can be an important complementary channel of perception for multimodal fusion techniques.

We present here a method for recognizing high-level activities of daily living based on low-level audio segment classification decisions. The first step is an audio classifier which generates a sequence of *low-level audio events*, i.e., a sequence of labels that characterize short segments of the audio signal. Naturally, the selection of labels depends on the application at hand: for the RADIO application and use cases, we have set labels for the sounds typically occurring in a home environment. The second stage is the application of a *meta-classifier* that maps sequences of low-level, short events to higher-level and longer-term activities (e.g. watching TV, doing housework, etc.) The conceptual architecture of this rationale is illustrated in Figure 7.

The contribution of the work described here is the meta-classification methodology and the experiments on a real-world dataset from a living-room and kitchen space. The experiments have proven that the proposed meta-classifier is very accurate (more than 90% on activity labels) and more accurate than directly recognizing the high-level activities from the raw audio features.

This machine hearing system can be deployed as part of the RADIO robot or as a standalone device. This flexibility offers alternative configurations that aim different points of balance between clinical data collection requirements and privacy, for areas such as the bathroom that are out of reach for the robot.

3.1.1 Low level audio events

All audio input/output operations are handled by the PortAudio library (Bencina and Burk, 2001), a widely used audio library for both open-source and commercial applications. It is implemented in C and offers both C++ and Python bindings. The reason why we have selected this library is that it supports *blocking audio input/output* which means it allows the existence of a contiguous audio block for processing without having real-time constraints.

The 34 audio features on Table 7 are extracted on a short-term basis resulting in a sequence of 34-

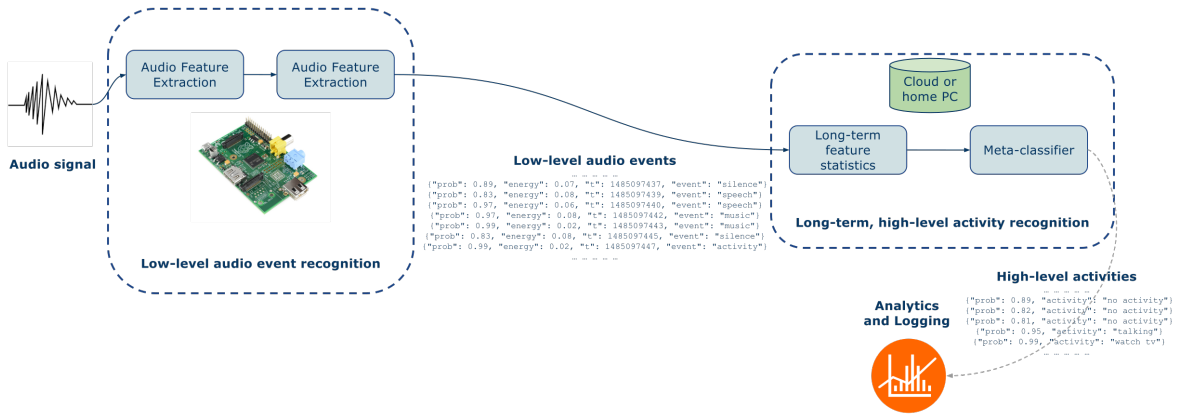

Figure 7: Conceptual architecture of the high-level audio event recognition scheme

Table 7: Adopted short-term audio features

Index	Name	Description
1	Zero Crossing Rate	Rate of sign-changes of the frame
2	Energy	Sum of squares of the signal values, normalized by frame length
3	Entropy of Energy	Entropy of sub-frames' normalized energies. A measure of abrupt changes
4	Spectral Centroid	Spectrum's center of gravity
5	Spectral Spread	Spectrum's second central moment of the spectrum
6	Spectral Entropy	Entropy of the normalized spectral energies for a set of sub-frames
7	Spectral Flux	Squared difference between the normalized magnitudes of the spectra of the two successive frames
8	Spectral Rolloff	The frequency below which 90% of the magnitude distribution of the spectrum is concentrated.
9-21	MFCCs	Mel Frequency Cepstral Coefficients: a cepstral representation with mel-scaled frequency bands
22-33	Chroma Vector	A 12-element representation of the spectral energy in 12 equal-tempered pitch classes of western-type music
34	Chroma Deviation	Standard deviation of the 12 chroma coefficients.

dimensional short-term feature vectors. The processing of the feature sequence on a mid-term basis is also adopted by dividing the signal into mid-term windows (segments). For each segment, the short-term processing stage is carried out and the feature sequence from each mid-term segment, is used for computing feature statistics. According to this rationale, *each mid-term segment is represented by a set of feature statistics* (e.g. the average value of the ZCR). The features are used in a probabilistic Support Vector Machine classifier (SVM) of audio segments.

Table 7 gives the adopted audio features. For further details, the reader can refer to the relevant bibliography (Giannakopoulos and Pikrakis, 2014; Theodoridis and Koutroumbas, 2008; Hyoung-Gook et al., 2005). The time-domain features (features 1–3) are directly extracted from the raw signal samples, while the frequency-domain features (features 4–8 and 22–34) are based on the magnitude of the *Discrete Fourier Transform (DFT)*. The cepstral domain, features 9–21, is the *Mel Frequency Cepstral Coefficients, MFCC*, computed by applying the Inverse DFT on the logarithmic spectrum.

3.2 Human activity recognition

Higher-level labels regarding everyday human activities that typically occur in the living room and kitchen area are estimated from the low level events. In particular, we have defined the following high-level events:

1. no activity (NO)
2. talking (TA)
3. watching TV (TV)
4. listening to music (MU)

5. cleaning up the kitchen (KI)
6. other activity (OT)

We aim to recognize the aforementioned activities in a long-term rate (e.g. every one minute) based on the short-term, low-level events. Towards this end, we have implemented the following meta-classification procedure: low-level audio segment decisions are used to extract feature statistics that are fed as input to a supervised model (meta-classifier), namely a Support Vector Machine. In particular, the adopted long-term feature statistics are the following:

- $F_i = \sum_{j=0}^{N} 1 : \lambda(j) = i$, where $i = 1, ..6,$, $\lambda(j)$ is the class label of the j-th audio segment and N is the total number of audio segments in a session (recording). These first 6 features are actually the distribution of short-term, low-level audio event labels as extracted by the audio classifier
- F_7 is the average number of transitions from silence to any of the non-silent audio classes per second
- F_8 to F_{11} are the mean, max, min and std statistics of the normalized segment energies.

This procedure leads to 11 features, i.e. a 11-D feature vector that represents the content of the whole recording session. Our goal is to recognize the high-level activity that occurs in the corresponding recording based on these long-term feature statistics. Note that the first seven feature statistics are based on the outputs of the low-level audio classifier, while features 8 to 11 are based on simple low-level energy values. As proven in the experimental section, these energy-based statistics lead to classification performance boosting. Several state-of-the-art classifiers have been adopted for mapping these long-term feature statistics to the 6 activity classes. In particular, the classification methods adopted are the following:

- k-Nearest Neighbor classifier
- Probabilistic SVMs (Platt, 1999)
- Random forests is an ensemble learning method used for classification and regression that uses a multitude of decision trees (Ho, 1995; Pal, 2005)
- Gradient boosting: an ensemble approach interpreted as an optimization task, widely used both in classification and regression tasks (Breiman, 1997; Friedman, 2002).
- Extremely Randomized Trees (or Extra trees) is a modern classification tree-ensemble approach that is based on strong randomization of both attribute and cut-point choice while splitting the tree node (Geurts et al., 2006), and has been adopted in various computer vision and data mining applications.

From an architectural point of view, the meta-classification stage for high-level event recognition only has access to low-level events, statistical derivatives of the audio signal. As no raw audio content is transported, this two-level architecture also allows for part of the audio processing to be executed off-board the robot or audio acquisition device. This allows the system the utilize computation services offered by the RADIO Home's local or cloud-based computation infrastructure.

Finally, the NSCR-D pyAudioAnalysis library (Giannakopoulos, 2015) has also been used to train and evaluate an SVM classifier that learns to discriminate between high-level activities *based on the low-level time sequences of short-term features*. However, as shown in the experiments section, this approach achieves much lower performance rates compared to the proposed meta-classification module.

3.3 Experiments

The goal of this section is to evaluate the performance of both the audio segment classifier and the meta-classification stage.

3.3.1 Datasets

Three different datasets have been compiled for the evaluation of both stages (audio segment classifier and meta-classification scheme for the activity recognition). The first two of the datasets are associated with the training and the evaluation stage of the low-level audio event classifier, while the third dataset is used to train and evaluate the meta-classifier. In particular the three datasets are described below:

1. *Audio segment classification dataset (Training)*: This dataset consists of one uninterrupted recording for each class, recorded through the mobile calibration GUI. The total duration of each recording is 200 seconds, giving 20 minutes in total for all six classes. This is the total time required to calibrate the audio recognition model, plus the time required for the SVM model to be trained.
2. *Audio segment classification dataset (Evaluation)*. This is used to evaluate the audio segment classifier. It consists of 600 audio segments, i.e. 100 from each audio class. Each audio segment is 1-sec long (the adopted decision window duration). Obviously the segments of this dataset are totally independent to the segments of the previous dataset to avoid bias in evaluation.
3. *High-level event dataset* This dataset is used to evaluate the proposed high-level activity recognition method. It consists of 20 long-term sessions for each activity class, i.e. 120 sessions in total. Each long-term session corresponds to a recording of varying duration, 20 seconds to 2 minutes in particular. The raw recordings are fed as input to the trained low-level audio segment classifier and that process has led to a series of short-term, low-level audio events. This processed dataset is openly provided.[2]. Each recording-session corresponds to another file. All files follow the JSON format, with the following fields:
   - winner class probability
   - (normalized) signal energy
   - timestamp
   - winner class

   The time resolution of all fields is 1 second, so each file has several rows that correspond to several audio segments of the same recording-session. Training and classification performance was measured using repeated random sub-sampling of this dataset. In particular, in each sub-sampling 10% of the data is used for testing.

3.3.2 Experimental Results

Low-level audio event recognition evaluation Table 8 presents the confusion matrix, along with the respective class recall, class precision and class F1 values for the audio segment classification task. The second dataset has been used to this end. The overall F1 measure was found to be equal to 83.5%. This actually means that more than 8 out of 10 audio segments (1 second long) are correctly classified to any of the 6 audio classes, on average.

High-level activity recognition First, we present the F1 measures for all meta-classifiers applied on:

- meta-features 1 to 6
- meta-features 1 to 7
- all meta features

The best performance is achieved for the Support Vector Machine classifier, when all features are used. In addition, the 7th meta-feature (the average, per second, transitions from silence to other audio classes) adds 2%. Furthermore, if the energy-related features are also combined, the overall performance boosting reaches 4%. Note that for the SVM classifier, a linear kernel has been adopted. Probably this is the reason that the SVM classifier overperforms the more sophisticated ensemble-based classifiers, since it

[2]Available at `https://zenodo.org/record/376480`

Table 8: Audio segment classification results: Row-wise normalized confusion matrix, recall precision and F1 measures. Overall F1 measure: 83.5%

Confusion Matrix (%)

	Predicted					
True ⇓	activity	boiler	music	silence	speech	wash basin
activity	79.3	0.0	0.0	5.2	6.9	8.6
boiler	32.4	51.4	2.7	13.5	0.0	0.0
music	0.0	0.0	88.0	0.0	12.0	0.0
silence	0.0	0.0	0.0	100.0	0.0	0.0
speech	0.0	0.0	12.9	0.0	87.1	0.0
wash basin	0.0	0.0	0.0	0.0	0.0	100.0

Performance Measurements (%, per class)

Recall:	79.3	51.4	88.0	100.0	87.1	100.0
Precision:	71.0	100.0	85.0	84.3	82.2	92.1
F1:	74.9	67.9	86.5	91.5	84.6	95.9

is proven to be more robust to overfitting due to low training sample size. Second, Table 10 presents the detailed evaluation metrics for the best high-level event recognition method, i.e. the SVM classifier for all adopted high-level features. According to the results above, the proposed methodology achieves a very high recognition accuracy for the six activities, based on simple low-level audio decisions and signal energy statistics, especially when the SVM classifier is used. The highest confusion is between the activities that include speech in their signal ('watching TV' and 'talking'–'music'), but these three as a whole are confidently separable from 'kitchen cleanup' and 'other activity'.

Finally, for comparison purposes, we have evaluated the ability of a one-level audio classifier that directly maps mid-term audio feature statistics to high-level activities. Using the same dataset, the F1 measure for this method was significantly lower (80%). Additionally, such an approach would require more training data, since it needs to much a high-dimensional audio feature space to semantically-high activity classes. This is not only computationally demanding, but impractical, since acquiring annota-

Table 9: Comparison between classification methods and feature sets for the high-level activity recognition task. Performance is quantified based on the F1 measure

	Features		
Method	1-6	1-7	1-11 (all)
kNN	84	88	84
SVM	88	90	**92**
Random Forests	89	89	90
Extra Trees	88	89	88
Gradient Boosting	88	88	89

tions for high-level activities is a laborious and intense task. On the other hand, the proposed approach has made training low-level, short-term multidimensional audio classifiers an easy task (through the calibration procedure described above), while the training of the meta-classifier requires only a few training samples, since it is based on audio segment classification decisions, not on very low-level and multi-dimensional audio features.

Table 10: High-level event recognition based on all meta-features for the best classifier (SVM)

Confusion Matrix (%)						
True ⇓ Predicted ⇒	no activity	talking	watching TV	music	cleaning up kitchen	other
no activity	100	0	0	0	0	0
talking	0	96	4	0	0	0
watching TV	7	11	71	5	0	5
music	0	5	5	89	0	0
cleaning up kitchen	0	1	0	0	95	4
other	0.5	0	0	0	0.5	99
Performance Measurements (%, per class)						
Recall:	100	96	71	89	95	99
Precision:	93	85	89	95	99	91
F1:	96	90	79	92	97	95
Overall F1 measure: 92%						

4 Activity Recognition using Home Automation Sensors

The context-aware system infrastructure described in D5.1 was extended for monitoring activities of daily living by setting a rule-based approach for both offline and real-time recognition. Novel aspects of the approach include the ability to recognise arbitrary scenarios of complex activities using bottom-up multi-level reasoning based on rules through the use of sensor events at the lowest level. Four ADLs were selected for recognition. We developed ad-hoc ADL classifiers based on a rule-based reasoning system (RBR). The rule engine finds facts in data and matches them against the rules. A rule looks as follows:

```
IF (TV_energy_consumption is ON) and (Pressure_sensor_chair is ON)
THEN Start_calculation_watchingTV_time.
```

This system is combined with an accurate indoor localisation approach that fuses location information with events. The RBR can be extended to include fuzzy in the input at the beginning and in the output at the end by using classes. For example:

```
IF (TV_energy_consumption is ON) and (Pressure_sensor_chair is ON)
for Long_Time
THEN Possible_Emergency.
```

Fuzzy logic provides a means of dealing with uncertainty depending on user's habits. For example, for one user, two hours spent watching TV can be long time while for another user, four hours can be short time.

This chapter gives the ad-hoc ADL rules implemented for the FHAG pilot.

4.1 Meal preparation

Name: Meal Preparation Start Time
Description: This rule captures the start time of the meal preparation. This considers any meal, breakfast, lunch, or supper.
Conditions: If [(cooktop is on) or (kettle is on) or (microwave is on) or (toaster is on)] and [(upper cabinet is opened) or (right cabinet is opened) or (left cabinet is opened) or (fridge is opened)] then
Actions: Set Meal Preparation Start Time

Name: Meal Preparation End Time
Description: This rule captures the end time of the meal preparation. It is considered meal that food served and eaten as breakfast, lunch, or supper.
Conditions: If [(cooktop is off) and (kettle is off) and (microwave is off) and (toaster is off)] and [(upper cabinet is closed) and (right cabinet is closed) and (left cabinet is closed) and (fridge is closed)] during 5 minutes then
Actions: Set Meal Preparation End Time

Name: Meal Preparation Duration Time
Description: This rule calculates the duration of the meal preparation. It is considered meal that food served and eaten as breakfast, lunch, or supper.
Conditions: If (Start Time is not 00:00 and End Time is not 00:00) then
Actions: Calculate the difference between End and Start Times

Note: The upper cabinet stores dishes, glasses, coffee mugs; the right cabinet stores food; and the left

cabinet stores the cooking pots and frying pan.

4.2 Watching TV

Name: Watching TV Start Time
Description: This rule captures the start time of watching TV.
Conditions: If [(TV energy consumption is on) and (pressure_sensor_chair is on) and (BLE is inside TV area) and (patient_identification is true)] or [(TV energy consumption is on) and (pressure_sensor_sofa is on) and (patient_identification is true)] or [(TV energy consumption is on) and (pressure_sensor_under_bed is on)] then
Actions: Set Watching TV Start Time

Name: Watching TV End Time
Description: This rule captures the end time of watching TV.
Conditions: If (TV energy consumption is off) or [(pressure_sensor_chair is off for 10 minutes) or (pressure_sensor_sofa is off for 10 minutes) or (pressure_sensor_under_bed is off for 10 minutes)] **Actions:** Set Watching TV End Time

Name: Time Spent Watching TV
Description: This rule calculates the total time spent watching TV.
Conditions: If (Start Time is not 00:00 and End Time is not 00:00) then
Actions: Calculate the difference between End and Start Times

4.3 Going out

Name: Frequency of Going Out
Description: This rule captures the number of times that the patient goings out on a daily basis
Conditions: If (main_door is open) and (motion_sensor1 and motion-sensor2 and motion_main_door are off for 5 minutes) then
Actions: Increment Going Out and Set Timestamp

4.4 Getting up from bed

Name: Frequency of Getting Up from Bed
Description: This rule captures getting up from bed during the night, to be fused with other methods.
Conditions: If [(current time is between 22:00 and 07:00) and (pressure_sensor_under_bed is off)] then
Actions: Increment Getting Up From Bed and Set Timestamp

5 INDOORS LOCALIZATION USING BLE BEACONS

Indoors positioning is becoming an important feature of IoT, Smart Home, and Wearable devices. Especially for devices that can change their location:

- Autonomously moving devices (robots, drones, etc.)
- Carried devices (wearables, smartphones, tablets, toys, etc.)
- Movable devices (voice command assistants, computers, furniture, etc.)

There are two generic approaches to indoor positioning, most of the times a combination of both its used to achieve highest accuracy:

- Self-contained, usually based on image or radar data. In these approaches, the device does not depend on any infrastructure. Instead, it tries to figure out the space on its own, and understand where it is located looking at pre-mapped surroundings. Most of the times, creating the map of the surroundings and understanding location, are done concurrently, gradually improving map and localization accuracy; this is referred to as SLAM (simultaneous localization and mapping)
- Infrastructure based, where the indoors space is equipped with fixtures that have known location. The moving device uses information provided by these fixtures to understand its location.

In this discussion, we focus only on the second approach. There can be different technologies applied, like acoustic or RF beaconing, visual markers etc. We are only concerned with RF-based solutions.

5.1 Background

5.1.1 Trilateration based on RSSI

Using similar principles as the ones used in outdoor positioning (GPS), this technique requires a number of radio transmitters, which send precisely synchronized timestamp-enhanced packets. The moving device is usually also synchronized so that when it reads the timestamp it can calculate the TOF (time of flight) of the packet, and thus the distance.

Trilateration has the great advantage of being similar (and in some implementation also compatible) with outdoors positioning systems. So for devices that are expected to be also outdoors like smartphones, it looks an obvious choice. However, it presents some non-trivial technical challenges:

- It needs extremely fine grain and accurate synchronization between the transmitters. Actually,

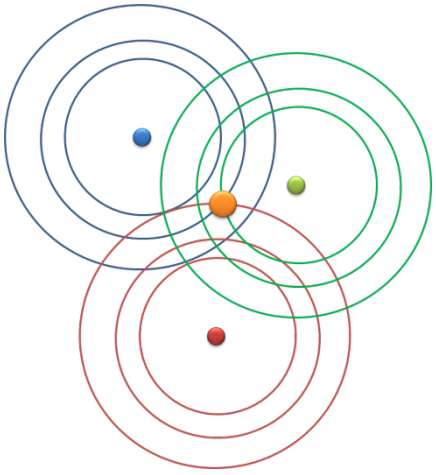

Figure 8: RSSI based trilateration

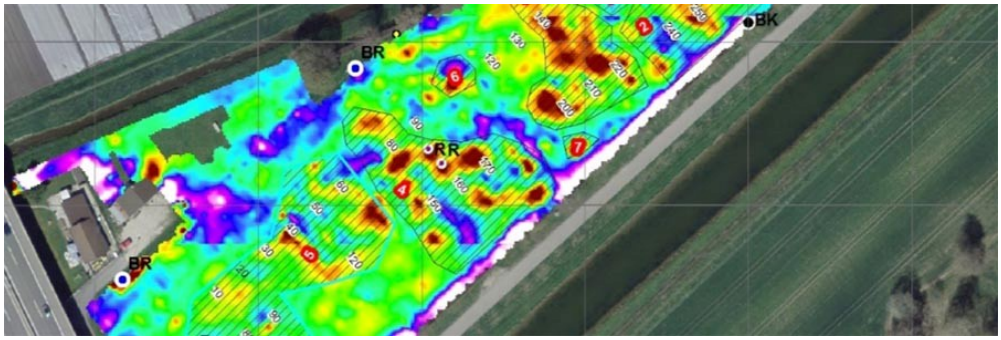

Figure 9: Fingerprinting based localization

localization accuracy depends almost entirely on this synchronization. This makes it very difficult to deploy and very vulnerable to temperature or other environmental factors which affect clock frequencies.

- It needs fine grain and accurate synchronization also for the receiver. This can be alleviated however if the system does not reply on three transmitters but has four or more. In that case, the receiver can self-synchronize; but this comes with a higher infrastructure cost.
- It is affected by reflections of the signal. If all is in the same room, the receiver can simply read only the highest-power packet in terms of multiple receptions (not that trivial as it sounds). But in multi-room cases, the trilateration system may get completely confused. To minimize this risk, sometimes a combination of simple beacons (to indicate which room we are in) is used, so that the device simply ignores packets from transmitters located outside.

5.1.2 Fingerprinting

Fingerprinting is actually a SLAM technique that does rely on RF transmitters. The idea is that the transmitters send continuous signals, which create — at any given location — a specific combination. The map of all these combinations is the fingerprint of the indoor region that is covered.

A device can create this fingerprint by moving around and logging how the various signals combine in each spot. This first 'moving around' has to be guided, so that the fingerprint data is registered on the floorplan of the region.

The fingerprinting technique is simple to deploy and generic, and it's becoming more and more popular. However, it needs careful handling of a couple of challenges:

- Since a huge number of spots will have the same fingerprint value, it is not enough to match values to locations. Instead, the moving device has to combine values from the previous locations (move history) to find a unique set of data that can be used for localization. This makes the techniques not suitable for more static devices that might be occasionally relocated, but ideal for robots and drones which are constantly on the move.
- The need for initial creation of the fingerprinting map. In some use cases this is not possible (e.g. when someone enters a previously-unvisited building). Lowest accuracy results can still be feasible, by employing ready-made maps or maps from another similar device.

5.1.3 Proximity-based

This is the simplest, lowest cost and less accurate approach. It simply replies on a number of radio beacons spread in the building. The moving device can listen to them, and understand where it is located by simply checking which device is within its vicinity.

The obvious benefit is low cost and easy deployment. The main drawback is accuracy: You can only tell you are close to one or more devices you cannot understand how close so that trilateration methods are applied. This is greatly improved however by:

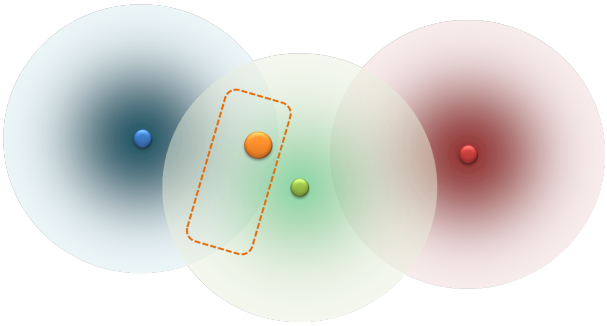

Figure 10: Proximity based localization

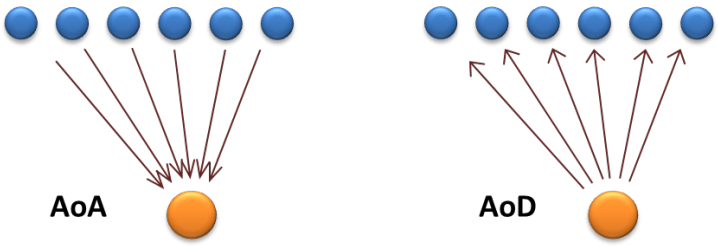

Figure 11: Angle of Arrival/Departure based localization

- Having a large number of short range beacons
- Using radio strength (RSSI) as a very rough way to estimate distance.

5.1.4 Using future BLE 5.x

Bluetooth radio can be used with any of the above techniques. Since Bluetooth is wide spread (in all smartphones and tablets and in a growing number of wearable and IoT devices), much research has gone into finding the boundaries and developing products to enable Bluetooth-based localization. The proliferation of BLE beacons goes a great step towards enabling proximity based or fingerprinting methods. This led the Bluetooth SIG to discuss further enhancements, which are expected to be standardized by the start of 2018, the enablement of AoA, AoD (angle of arrival, angle of departure) information in BLE packets.

AoA and AoD are based on principles similar to the trilateration approach, since they also use accurate timing comparison from multiple transmitters. But — instead of time of flight — they determine TDOA (Time Difference of Arrival / Departure) from/to an antenna array. These differences can be used to calculate a good estimate of the angle between the antenna array and the moving device.

The benefit of AoA/AoD based approaches is that they are easier to deploy, since the antenna array comes as a single appliance. However, the standard specification is not yet released, which means that products are not expected on the market for at least one or two years from now.

So, although long-term it looks like the most promising technology, we still have to rely on custom implementations or apply less accurate proximity-based techniques.

5.2 RADIO Absolute Localization Methods

Indoor wireless localization using Bluetooth Low Energy (BLE) beacons has attracted considerable attention after the release of the BLE protocol. BLE beacons have the following advantages: small size, light weight, low cost, power saving and are widely supported by smart devices, which make them a dominant wireless localization technology. As indicated in previous sections there are three main localization techniques, proximity-based, trilateration and fingerprinting. Our localization system, depicted in Figure 12, uses the trilateration technique with additional filters in order to reduce the measurements noise.

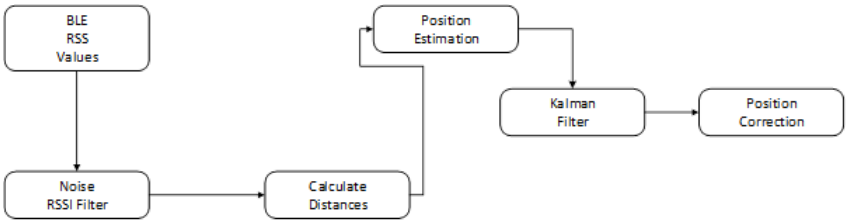

Figure 12: RADIO localization system

Table 11: Path Loss Exponents for Different Environments

Environment	Path Loss Exponent γ
Free Space	2
Urban Area cellular radio	2.7 to 3.5
Shadowed urban cellular radio	3 to 5
In building (Line of Sight)	1.6 to 1.8
In building (Obstructed)	4 to 6
In Factories (Obstructed)	2 to 3
Home Environment	4.5
Office Environment	3.5

Trilateration is accepted as the most appropriate way to determine the location of a sensor node based on locations of beacons (Abdullah et al., 2013). The procedure attempts to estimate the position of a node by minimizing the error and discrepancies between the measured values. Once the distances between a target node and all reference nodes based on the RSSI of the received packets are found, the position of the target node is calculated using the trilateration method. In order to calculate the distances between the beacons and the observation mote, we collect RSSI values from the beacons. The RSSI is a measurement of the power of a radio signal. A main challenge with RSSI ranging is that the effect of reflecting and attenuating objects in the environment can radically distort the received RSSI, making it difficult to infer distance.

In order to mitigate the RSSI noise we use an *autoregressive moving average (ARMA)* model (Haykin, 2002), using the following equation:

$$RSSI_t = RSSI_{t-1} - c\,(RSSI_{t-1} - RSSI_t)$$

where c is a coefficient that denotes the smoothness, lower value, means smoother the moving average. The determination of the coefficient, it is done empirically with measurements in different locations. After the mitigation of the RSSI value, we calculate the distance using the log-normal shadowing model, as follows:

$$P(d) = P(d_0) - 10\gamma \log_{10}\left(\frac{d}{d_0}\right) + X_\sigma$$

where γ represents the path-loss exponent. Typical path loss exponent (γ) (Rappaport, 1996) are shown in Table 11. $P(d_0)$ represents the RSS at the reference distance, d_0, $P(d)$ represents the RSS at the distance between the observation node and the receiver, d and X_σ represents a Gaussian random variable, with zero mean, caused by shadow fading.

Finally, after the first estimation of the position using the trilateration technique, we adapt the Kalman Filter (KF) in order to estimate the target's current location by fusing current and historical information. The KF addresses the general problem of trying to estimate the state $x \in \mathcal{R}^n$ of a discrete-time controlled process that is governed by the linear stochastic difference equation.

$$x_k = Ax_{k-1} + Bu_{k-1} + W_{k-1}$$

with a measurement $z \in \mathcal{R}^n$ that is

$$z_k = Hx_k + v_k$$

The random variables W_k and v_k represent the process and measurement noise, which are white Gaussian noises. These matrices represent the covariance matrices of the process (or input), respectively of the measurement noise.

The state vector (x) of the KF is given by

$$x = [\ x\ y\ u_x\ u_y\ ]$$

where x and y are 2D position in the horizontal plane and v_x and v_y are their corresponding 2D velocities.

The state transition matrix (A) relates the step $k-1$ to the current state k. The Δ_t variable represents the sampling period.

$$A = \begin{bmatrix} 1 & 0 & \Delta_t & 0 \\ 0 & 1 & 0 & \Delta_t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The input control matrix (B) is given by

$$B = \begin{bmatrix} \Delta_t^2/2 \\ \Delta_t^2/2 \\ \Delta_t \\ \Delta_t \end{bmatrix}$$

The final estimated position is correlated to the current RSSI measurements, and the position these values give, but also taking account the previous positions of the observation object.

5.3 RADIO Relative Localization Methods

The RADIO relative localization method is based on the proximity approach as already briefly described. This approach overcomes the drawbacks and the peculiarities that the RSSI values and the respective distance models impose (due to signal distortion). In that context, RSSI is used as a relative measure that indicates how close a BLE enabled device is to another BLE device and not its absolute distance.

Similar implementations are given by the proximity profile, as defined by the Bluetooth 4.0 specification (SIG, 2014). Particularly, the proximity profile defines the behaviour when a device moves away from a peer device so that the connection is dropped or the path loss increases above a predefined level. The

profile operates on the connectable mode of BLE and defines two roles for the devices, the proximity monitor and proximity reporter. The proximity monitor maintains a connection with the proximity reporter and monitors the RSSI of this connection. When the connection is lost the proximity reporter alerts to the level specified in the alert level characteristic as defined by the link loss service specification.

While the proximity service can be used as an indication of whether a device (reporter) is near to another BLE device (monitor), it cannot be used as a service that locates the position of an object in the space. The relative localization method designed and implemented in the RADIO project enables objects that are equipped with BLE devices to be trackable in the space they operate. The implemented solution is based on a three-tier architecture. Particularly, the wireless network formed by the BLE devices consists of the reporters, monitors and the gateway.

Reporters are BLE devices attached to objects to report their presence in the space. The data unit used for the RADIO relative localization service (as in the absolute localization service) is based on the beacon format. As already mentioned for the RADIO absolute localization approach, the beacon format has been selected due its compact size and its wide adoption by BLE applications. The additional fields needed have been injected in the payload segment of the packet.

Monitors are devices that comprise the static infrastructure of the space and they are responsible to scan for relative localization beacons and report them to the gateway. The monitor BLE devices are installed to predefined locations and their position is known. The monitor nodes scan for surrounding BLE devices that advertise themselves. The advertisements are scanned and the RSSI values are recorded by the static nodes to the report packet. The work cycle of the monitor device for every captured presence packet is completed when the monitor forwards the report packet to its neighbours until it reaches the gateway. The presence and report packets are beacon advertisement packets which carry in their payload information such as the node id of the monitor node that scanned the report packet and the respective RSSI value of the wireless link among the two devices (monitor and reporter).

Finally, the *gateway* collects the report packets sent from the monitors and processes the information that the report packets carry to track the objects' location. The gateway has stored the network structure and locations of the monitor nodes. From every received report the gateway extracts the monitor node id that listens the reported node, and the RSSI value. The location of a reporter node is assigned to the location of the monitor node that listened the reporter with the stronger RSSI.

However, the distortion of the RSSI signal caused by object reflections and interference imposed some inaccuracies in the relative localization service. Therefore, the RSSI noise was mitigated through the ARMA model described earlier.

5.4 Smart Home – BLE Gateway Cooperative Isolated Methods

5.4.1 Identification of furniture close to TV through relative localization and interface with the main controller

The Relative Localization service has been applied and evaluated in the RADIO environment by monitoring and identifying the furniture that are close to the TV. The static infrastructure as described in the previous section has been in installed in a smart room and the fixed positions of the monitor devices where recorded.

The next step of the deployment was to define the object that was going to be tracked in the smart room. For that purpose, a BLE node operating in the reporting mode was attached on a chair. The BLE node was configured to report its presence every second.

The monitor nodes are scanning for presence packets that are transmitted by the reporter node. By the time such a packet is recorded, the monitor nodes construct the report packet and switch to the advertisement mode to forward it to the gateway. The gateway functionality is implemented as a Java application component following the programming model described in Section 5.2, D3.7. The observations extracted by the gateway relative localization application are exposed to the main controller through the

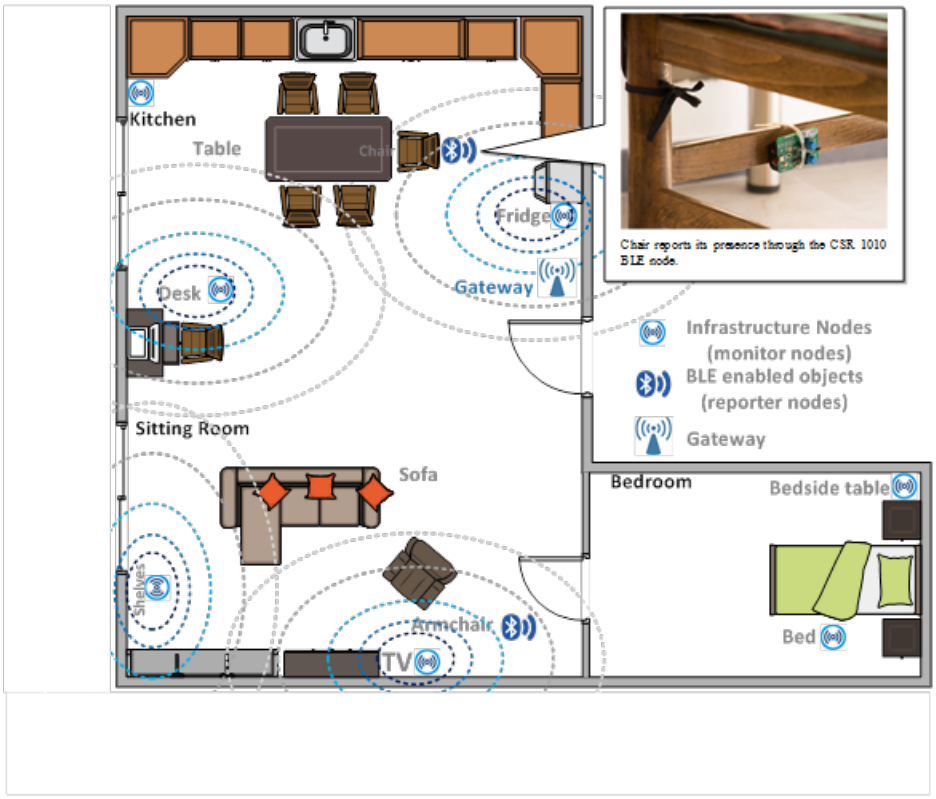

Figure 13: Indicative network constellation that delivers relative localization services

MQTT protocol (Banks and Gupta, 2015).

A typical constellation of a BLE based network that delivers the relative localization service is depicted in Figure 13. As presented, the home has preinstalled BLE devices with known locations responsible to monitor surrounding objects (TV, fridge, desk, bed, etc.). As presented, two objects (chair and armchair) located in different places in the house are reporting their presence. The chair located near the fridge is reported by the monitor attached on the fridge device, and then the report packet is transmitted to the gateway. At the same time, the armchair is reporting its presence, and the presence packets are captured by the monitor node attached to the TV. The TV node constructs the report packet and forwards through its neighbors to the gateway. The report packet transmitted by the TV node, reaches the gateway through three hops forwarding (travels through the shelves node to the desk node and finally to the gateway). The gateway publishes the updated location of an object only if it has changed from the previous time it was reported.

The object locations published by the BLE gateway to the main controller can be described by the following JSON model:

```
{
  "vicinity": [{ // array of points of interest , e.g. TV (id and name)
    "id": Number, "name": String,
    "things": [{  // array of things in the vicinity of poi (id and name)
      "id": Number, "name": String
    }]
  }]
}
```

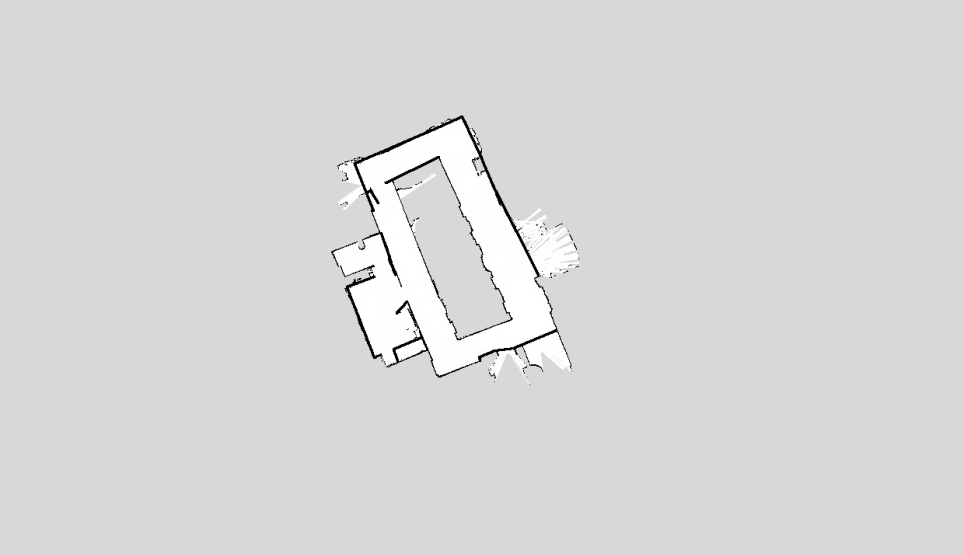

Figure 14: The FHAG trials area map as mapped by the RADIO robot.

5.5 Robot Tracking through RADIO Map Convergence

A required service identified through the progress of the project is to give the ability to authorized users (such as a technician) to locate and track the location of the robot through a user friendly graphical interface. In that context, a robot localization method has been designed and implemented. The Radio robot tracking service utilizes the robot's navigation and localization system. An application running on the NUC that controls the robot gets every second the position of the robot as given by the ROS environment. The developed application runs the following command:

```
rostopic echo /global_pose
```

subscribes to the `global_pose` ROS topic, and gets the following output:

```
x: 1.01669911569
y: 0.110426031014
theta: 0.0
---
```

The fields x and y denote the position of the robot on the x and y axes of its map, where theta is the orientation of its movement. Figure 14 presents an indicative map that the robot uses for its navigation.

The x and y values extracted from ROS are sent to the backend and the UI through the `/position` MQTT topic. In the backend takes part the second step of the robot localization process which is the convergence

of the robot's location on its own map to the UI map coordinates. The x and y values sent by the ROS to the global pose topic are relative to a $(0, 0)$ point that is marked manually on the UI map at deployment time as part of the mapping process. The resolution metric, given by the pose configuration, is used to transform distance to pixels on the bitmap. These definitions are used by the UI to converge the robot map and the UI map.

5.6 Security through Theft Detection Methods

Anti-theft systems are designed to raise the difficulty of theft to an infeasible level. Our theft detection method uses accelerometer and gyroscope data in order to identify theft situations. The system architecture is shown in the Figure 15. As shown, the system contains four main components: the *Data Parser* which is responsible to gather the required data, the *Data Calibrator* which is used to calibrate the data, and the *Antitheft Detector* which is the component that checks if a theft attempt occurred. Finally, the

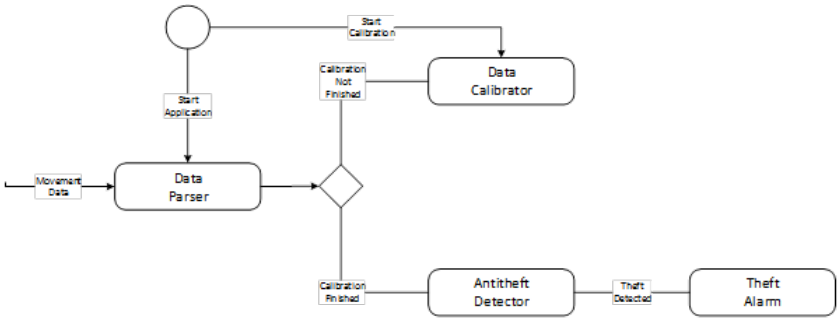

Figure 15: Antitheft System Architecture

Theft Alarm component is the broadcaster, which notifies that a theft situation occurred.

The system is divided in two phases, the Calibration phase and the Theft detection phase. When the system starts, it is in Calibration phase, which lasts 10 seconds. In that time the Data Calibrator collects accelerometer and gyroscope data, and determine the minimum and maximums thresholds for the three axis for both accelerometer and gyroscope sensors. Due to the accelerometer and gyroscope sensor do not have steady values, when in absolute stillness, when the calibration phase finish, we need to add a gap to the minimum and maximum values observed, to avoid small deviations from the thresholds when the system is on theft detection phase. This gap is a constant value, configured empirical observations.

After the calibration phase finishes, the system switches to the theft detection phase, where it is rtying to determine if an abnormal situation occurred to the observation object, in our case the RADIO robot. More specifically, in the theft detection phase the Theft Detector component tries to determine if the current accelerometer and gyroscope data are inside the thresholds defined during the calibration phase. Again due to non-steady values from the sensors, there is a possibility to identify a theft attempt, which is not correct. To avoid this problem, we define a threshold, using empirical observations, which determines how many theft detections need to be observed at least, in order to have a theft situation. Finally, when a theft situation is identified a theft alarm is sent to the system administrator.

REFERENCES

H. Abdullah, I. Khan, and M.S. Zainal. Proposed wireless sensor network in vehicles for road surveillance. In *Proceedings of the 2013 IEEE Conference on Systems, Process and Control (ICSPC 2013)*, 2013.

Andrew Banks and Rahul Gupta. MQTT protocol specification. Technical report, OASIS, December 2015. URL `http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html`.

Ross Bencina and Phil Burk. PortAudio–an open source cross platform audio API. In *Proceedings of the International Computer Music Conference, Havana*, pages 263–266, 2001.

Leo Breiman. Arcing the edge. Technical Report 486, Statistics Department, University of California at Berkeley, 1997.

Jerome H Friedman. Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4): 367–378, 2002.

Juergen Gall, Angela Yao, Nima Razavi, Luc Van Gool, and Victor Lempitsky. Hough forests for object detection, tracking, and action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 33(11):2188–2202, 2011.

Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine learning*, 63 (1):3–42, 2006.

Theodoros Giannakopoulos. pyAudioAnalysis: An open-source Python library for audio signal analysis. *PloS One*, 10(12), 2015.

Theodoros Giannakopoulos and Aggelos Pikrakis. *Introduction to Audio Analysis: A MATLAB® Approach*. Academic Press, 2014.

S. Haykin. *Adaptive Filter Theory*. Prentice Hall, New Jersey, 4 edition, 2002.

Tin Kam Ho. Random decision forests. In *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*, volume 1, pages 278–282. IEEE, 1995.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin. A practical guide to support vector classification. Technical report, Department of Computer Science, National Taiwan University, July 2003.

Kim Hyoung-Gook, Moreau Nicolas, and Thomas Sikora. *MPEG-7 Audio and Beyond: Audio Content Indexing and Retrieval*. John Wiley & Sons, 2005. ISBN 047009334X.

Mahesh Pal. Random forest classifier for remote sensing classification. *International Journal of Remote Sensing*, 26(1):217–222, 2005.

Michalis Papakostas, Fillia Makedon, Theodoros Giannakopoulos, and Vangelis Karkaletsis. Short-term recognition of human activities using convolutional neural networks. In *Proceedings of the 2016 International Conference on Signal-Image Technology and Internet Based Systems (SITIS 2016)*, 2016. doi: https://zenodo.org/record/376482.

Hamed Pirsiavash and Deva Ramanan. Detecting activities of daily living in first-person camera views. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2847–2854. IEEE, 2012.

John C Platt. Probabilistic outputs for support vector machines and comparisons to regularized likeli-

hood methods. In *Advances in Large Margin Classifiers*. Citeseer, 1999.

T. S. Rappaport. *Wireless Communications: Principles and Practice*. Prentice Hall, New Jersey, 1996.

Bluetooth SIG. Bluetooth specification version 4.0. Technical report, Bluetooth SIG, 2014.

S. Theodoridis and K. Koutroumbas. *Pattern Recognition, Fourth Edition*. Academic Press, Inc., 2008.

Pavan Turaga, Rama Chellappa, Venkatramana S Subrahmanian, and Octavian Udrea. Machine recognition of human activities: A survey. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(11):1473–1488, 2008.