



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 3.4

ADL and mood recognition methods I

Dissemination Level	Public
Due Date of Deliverable	Project Month 15, June 2016
Actual Submission Date	7 April 2017
Work Package	WP3, <i>Modular conceptual home architecture design and ICT method development for efficient, robust and flexible elder monitoring and caring</i>
Task	T3.2, <i>ADL and emotion recognition method development</i>
Lead Beneficiary	NCSR-D
Contributing Beneficiaries	TWG, AVN
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 their technical evaluation. The deliverable also comprises the prototype implementations of these methods.

History

Version	Date	Reason	Revised by
01	18 Jan 2016	Document structure	NCSR-D
02	24 Jun 2016	Human walking tracking in range data, method description (Section 2) and experiments on cluster alignment (Section 2.4)	NCSR-D
03	27 Jun 2016	Visual recognition of bed transfer and pill intake events (Section 4)	AVN
04	2 Jul 2016	Walking tracking experiments using the URG-04LX laser scanner (Section 2.5)	NCSR-D
05	2 Aug 2016	Evaluation of bed transfer and pill intake recognition (Section 4.3).	TWG, AVN
06	4 Oct 2016	Moving object tracking in RGB-D signal (Section 3)	NCSR-D
07	3 Nov 2016	Walking tracking experiments using the UST-10LX laser scanner (Section 2.5)	NCSR-D
08	29 Mar 2017	Evaluation Moving object tracking in RGB-D signal (Section 3.7)	NCSR-D
09	3 Apr 2017	Final editorial editing	NCSR-D
10	5 Apr 2017	Internal review	ROBOTNIK
Fin	7 Apr 2017	Addressing internal review comments, final document preparation and submission	NCSR-D

Executive Summary

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

These methods implement the components foreseen by the ADL recognition architecture (D3.1 *Conceptual Architecture I*). Development and testing is driven by realistic experimental datasets, which will be publicly released as part of D3.5, the upcoming second deliverable for this task. The ADL recognition methods developed so far are:

- A human walking pattern recognition method that tracks the movement of individuals in the robot's vicinity by analysing range data from a laser scanner.
- Machine vision methods that recognize and track motion and that identify the onset and ending of a bed transfer activity. These are then used to measure the duration of bed transfer activities.

Range data from a laser scanner is used to recognize human motion patterns and track the movement of people. A second-level analysis of these tracks measures the time it takes to walk 4m, which is one of the measurements required from the RADIO system. Detecting human walking in range data is considerably harder than using colour and depth images, but is inherently privacy-preserving because the original data stream does not contain any visual information. Work during RADIO methodologically improves a method previously developed by NCSR-D, also porting the implementation to Python so that it can be integrated with the rest of the system.

The machine vision methods integrated in the RADIO system detect motion by comparing the current image frame with some frame from the past. Comparison is performed by accumulating the differences between current and past pixel values in small square blocks. The motion of the same person is tracked between frames, and heuristic rules are applied about what constitutes motions characteristic of getting out of bed or using a medication cup. Further development in D3.5 will complement this line of work with classifiers for further activities.

Abbreviations and Acronyms

ADL	Activities of Daily Living
HPR	Human Pattern Recognition, a method for tracking walking humans by clustering and recognizing human walking patterns in range data
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

CONTENTS

Contents	iv
List of Figures	vi
List of Tables	vii
1 Introduction	1
1.1 Purpose and Scope	1
1.2 Approach	1
1.3 Relation to other Work Packages and Deliverables	2
2 Human Pattern Recognition in Range Data	3
2.1 Overview	3
2.2 Background	3
2.3 The HPR Method	4
2.4 Cluster Alignment	5
2.4.1 Method	5
2.4.2 Evaluation	7
2.5 Cluster Tracking	9
2.5.1 Method	9
2.5.2 Evaluation	9
3 Moving Object Tracking in RGB-D	12
3.1 Overview	12
3.2 Undefined areas elimination and smoothing	12
3.3 Lighting conditions preprocessing	12
3.4 Change detection	14
3.5 Bounding box formation	15
3.6 Tracking bounding boxes through frames	15
3.7 Evaluation	19
4 Visual Event Recognition	22
4.1 Overview	22
4.2 Method Description	22
4.2.1 Time to stand up from bed and start walking	22
4.2.2 Handling of the medication cup	24
4.3 Evaluation	24
4.3.1 Evaluation approach	24
4.3.2 Experiments to characterize detection of bed transfer	25
4.3.3 Experiments to characterize detection of medication cup handling	25
4.3.4 Motion detection Algorithms' Categorizes	26
4.3.5 Description of Motion detection Algorithm's main approach	26
4.3.6 The method of Motion detection Algorithm's trials	27
4.3.7 Evaluation scenarios for human activities detection	28
4.3.8 Evaluation scenarios for object movement detection	33
4.3.9 Summarization	37
References	38

LIST OF FIGURES

1	Dependencies between this deliverable and other deliverables	2
2	3D point-cloud of a pair of walking legs.	4
3	3D point-cloud of somebody walking away from the robot.	6
4	3D point-cloud of non-walking movement	6
5	Grid images of the three clusters.	7
6	Walking trajectory across several frames	9
7	Characteristic example of occlusion.	10
8	Tracked clusters in a crossed human situation	11
9	Closing operation example	13
10	Depth sensor raw image	13
11	Depth sensor image after Closing operation	13
12	Region grow operation example	14
13	Depth sensor raw image	14
14	After Region grow	14
15	Depth sensor processed image	15
16	Gamma chart	17
17	Grayscale raw image	17
18	After gamma	17
19	Clahe-histogram selection example	19
20	Clahe-histogram clipping and redistribution	19
21	Grayscale raw image	20
22	After Clahe	20
23	Grayscale processed image	20
24	Reference image	21
25	Difference image	21
26	Difference image	21
27	Bounding box formation	21
28	Typical sequence of events	23
29	Alternative testing settings	25
30	Alternative testing settings	26
31	Human activities detection 1st Scenario Startup Phase	28
32	Human's Activities Detection 1st Scenario Initialization Phase	29
33	Human's Activities Detection 1st Scenario Standing Event Identification	29
34	Human's Activities Detection 1st Scenario Walking Away Event Identification	30
35	Human activities detection 2nd Scenario Startup Phase	30
36	Human's Activities Detection 2nd Scenario Initialization Phase	31
37	Human's Activities Detection 2nd Scenario Standing Event Identification	31
38	Human's Activities Detection 2nd Scenario Walking Away Event Identification	32
39	Human's Activities Detection 3rd Scenario Startup Phase	32
40	Human's Activities Detection 3rd Scenario Initialization Phase	33
41	Human's Activities Detection 3rd Scenario Standing Event Identification Failure	33
42	Human's Activities Detection 3rd Scenario Walking Away Event Identification Failure	34
43	Human's Activities Detection 4th Scenario Startup Phase	34
44	Human's Activities Detection 4th Scenario Initialization Phase	35
45	Human's Activities Detection 4th Scenario Standing Event Identification Failure	35
46	Human's Activities Detection 4th Scenario Walking Away Event Identification Failure	36
47	Cup Movement Detection Algorithm Initialization Phase, 1st Scenario	36
48	Cup Movement Detection Algorithm Test of not moving the Cup, 1st Scenario	37

49	Cup Movement Detection Algorithm Test of Expected Sequences of Using the Cup, 1st Scenario	37
50	Cup Movement Detection Algorithm Initialization Phase, 2nd Scenario	38
51	Cup Movement Detection Algorithm Test with respect to various final positions, 2nd Scenario	38

LIST OF TABLES

1	List of prototypes of the methods described in D3.4	1
2	Evaluation of classification performance with and without alignment	8
3	MAE and MRE for the walking analysis	11
4	Evaluation results for 75% overlap	19

1 INTRODUCTION

1.1 Purpose and Scope

This report documents the sensor data analysis methods developed during M7-M15 of Task 3.2. These methods recognize the *Activities of Daily Living (ADL)* identified in D2.6 *Guidelines for balancing between medical requirements and obtrusiveness I* as the objective for ADL recognition for the first phase of method development.

This document reports on the methods developed in Task 3.2 during M7-M15 and their technical evaluation. The prototype implementations of these methods (also developed in Task 3.2) are available as public software repositories (Table 1).

The integrated analysis system, including these method implementations and other auxiliary software, is delivered as D3.9 *Integrated data analysis system*.

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 first iteration of the system list of components and architecture (D3.1 *Conceptual architecture I*). We then proceeded by assuming this background as a starting point for the development of the 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, which will also be released at the end of this task.

Upon achieving satisfactory results on these datasets, the components are then tested by users within the target group in the formative evaluation trials conducted in controlled environments at FSL (D6.6 *Controlled pilot trials report II*). These latter datasets cannot be released, but are used internally to adapt the ADL recognition methods and to refine the scenarios we record in the Roboskel Lab/AAL House datasets so that the technical validation datasets are more realistic.

Table 1: List of prototypes of the methods described in D3.4

Method		Repository
Sect. 2: range data	Walking pattern recog- nition	https://github.com/radio-project-eu/ HumanPatternRecognition
Sect. 3: image, depth	Moving object tracking	https://github.com/radio-project-eu/ros_visual
Sect. 4: image	Bed transfer and pill intake	Full computer: https://github.com/radio-project-eu/ motion_analysis FPGA (without CV library): https://github.com/ radio-project-eu/simple-visual-events

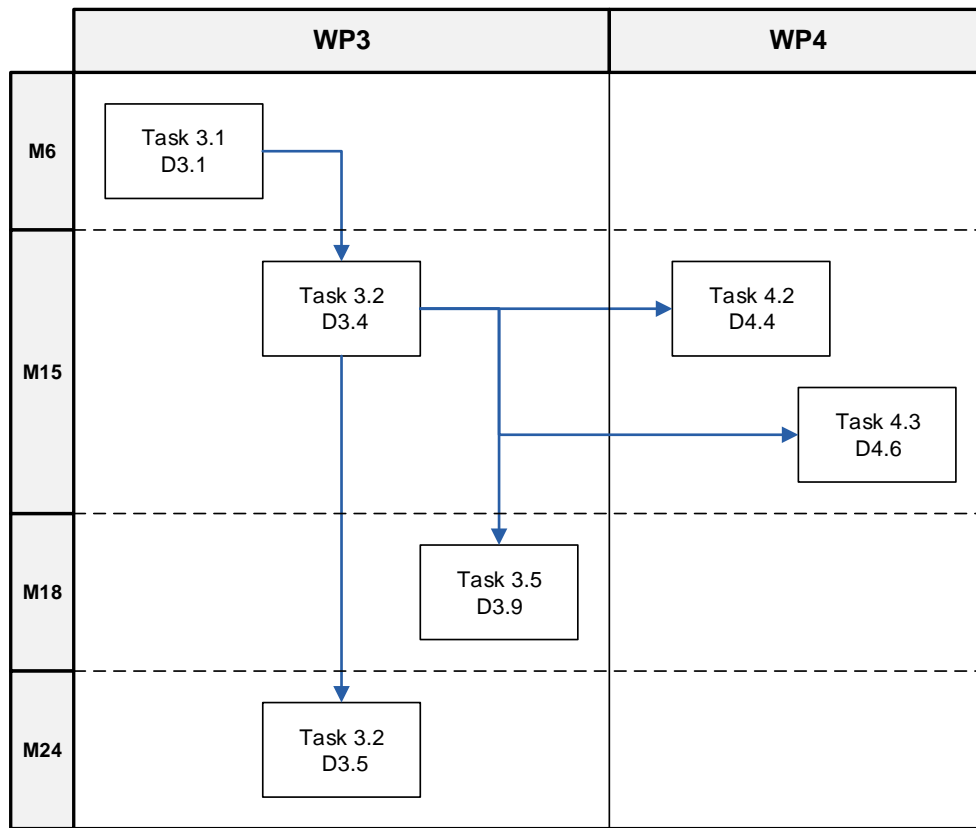


Figure 1: Dependencies between this deliverable and other deliverables

1.3 Relation to other Work Packages and Deliverables

This deliverable depends on *D3.1 Conceptual architecture for sensing methods and sensor data sharing I* which sets technical requirements on the ADLs that need to be recognized and also defines the architecture that combines the various components that make up D3.4.

The software prototypes developed in Task 3.2 (also part of D3.4) are used by Task 3.5 *Privacy and resource-sensitive integrated data analysis* to prepare *D3.9 Integrated data analysis system*, noting that D3.9 also includes other components that are not directly relevant to ADL recognition. The methods in D3.4 are also used by Task 4.2 *Embedded device design and development* to carry out research on which of these components can be more efficiently implemented as hardware components.

Finally, the methods in D3.4 will be complemented by the methods in *D3.5 ADL and mood recognition methods II*, also developed in Task 3.2, to form together the overall RADIO suite of perception methods.

2 HUMAN PATTERN RECOGNITION IN RANGE DATA

2.1 Overview

The advantages of using range data to detect and track movement are that reliable and accurate measurements can be made, especially in indoors applications, and that the range data (and especially the planar laser scanner data discussed here) can be unobtrusively collected by comparison to wearable motion sensors. What is also interesting is that planar range data carry, by its nature, very little information. This is an advantage in avoiding the privacy issues around collecting and analysing visual or even 3D data, but also makes it practically impossible to extract characteristic features from individual scans. As a consequence, the community has drawn its attention to detecting *moving* objects so that characteristic movement patterns can be extracted from richer object representations that span multiple frames.

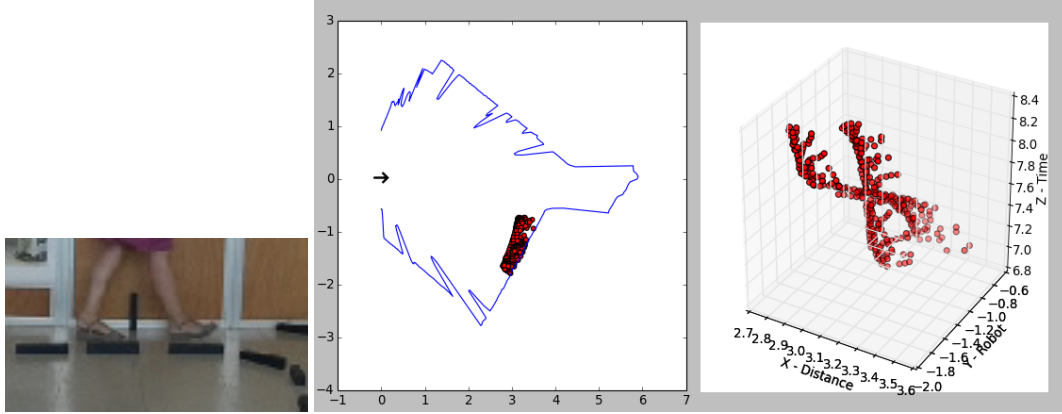
The core idea of our HPR method is to stack consecutive 2D scans into a 3D space-temporal representation, where X,Y is the planar data and Z is the time dimension. This 3D representation combines information about the shape and size of objects with information about their motion in time. This provides richer information wherein clustering and pattern recognition can be applied to recognize human walking patterns and to track the different individuals in the scene.

The RADIO component assumes as a starting point previous relevant work by NCSR-D (Section 2.3). Work during RADIO (a) improved the classification of movement patterns as human walking or not, by developing a novel pre-processing step which aligns the spatio-temporal objects, so that information about the direction and speed of movement is factored out of the representation (Section 2.4); (b) extended the method with tracking objects across frames using overlapping frame windows (Section 2.5).

2.2 Background

In order to track movement across scans, we need to associate scan points in one scan with the scan points of the same person in a subsequent scan. This is a challenging problem in situations such as occlusion, data sparsity, and physical proximity of objects. There are two lines of research: The first is based on *Kalman filters* that serve as *motion models* and track objects by estimating the future track position from past observations (Arras et al., 2008). Although successful, Kalman filters face the key issue of defining the motion model. To address this, Spinello et al. (2008) predefined three different motion models and in each step chose the one with the highest probability. Bennewitz et al. (2005) proposed an unsupervised algorithm in which motion patterns were learnt automatically using *expectation-maximization estimation* and Hidden Markov Models. Other approaches are based on assumptions about the environment to simplify the problem. Nemati and Åstrand (2014), for example, use hard limits on object size to separate moving humans from moving forklifts in an automated industrial environment and limits on maximum speed and maximum proximity of different objects to segment scan points into objects and to associate objects across scans. More recent methods fuse multiple modalities to improve robustness. Ristić-Durrant et al. (2016), for example, fused range data with 3D depth data. Although successful in increasing robustness, the fusion of different modalities voids the privacy argument in favour of planar range data.

An alternative approach (and the one assumed as the basis for the work in RADIO) is to stack multiple planar scans into a 3D *frame* where time serves as the third dimension (Varvadoukas et al., 2012). This representation simultaneously informs about the size and shape of the objects in the scene and their movement, so that no explicit motion models are necessary. These 3D objects are then clustered based on this spatio-temporal proximity, so that ‘clearer’ scans inside the frame can help solve occlusions and sparsity in more cluttered or distant instances of the same object’s track through the frame. After segmentation, these spatio-temporal representations are treated as 3D objects and classified as ‘human walking’ instances using methods from machine vision (surface modelling, HOG feature extraction, and



The 3D spatio-temporal representation (right) and the projection on the map (middle) of somebody walking across the front view of the robot (left). The robot is at $(X, Y) = (0, 0)$ facing towards the positive half of the horizontal axis (X), as marked by the black arrow in the middle image.

Figure 2: Characteristic example of the 3D point-cloud of a pair of walking legs.

classification). The fundamental assumption is the same as by Nemati and Åstrand (2014) that object segmentation and object tracking through time should be based on geometric proximity. However, in the method by Varvadoukas et al. (2012) these assumptions are manifested as unsupervised clustering in the Euclidean space whereas Nemati and Åstrand (2014) hardwire the definition of ‘human leg’ and ‘forklift’ as limits on the width of these objects in the scan.

2.3 The HPR Method

As already mentioned, the core idea of the *HPR* method is to approach walking pattern recognition as the task of classifying the 3D objects created by stacking consecutive 2D scans into a 3D spatio-temporal representation, where X, Y is the planar data and Z is the time dimension (Varvadoukas et al., 2012).

More specifically, the first step is to remove the points that correspond to the static map created by Simultaneous Localization and Mapping and used to localize the robot in the environment. The remaining scan points correspond to dynamic objects in the environment. These are used to construct 3D frames by translating the polar scans into the 2D Cartesian space, and then buffering multiple such 2D planes for a period of time. The time dimension is translated into space by multiplying with 5 kmph, the average speed of human walking. In this representation, a pair of walking legs creates the characteristic helix-like shape curves shown in Fig. 2.

The 3D data points are separated and grouped into 3D objects using the DBSCAN clustering algorithm (Ester et al., 1996). DBSCAN is an established density-based clustering technique which builds clusters of arbitrary shape, and is efficient for low-dimensional data. DBSCAN fits our spatial data very well, because it takes advantage of the proximity of the data points in the plane. Another advantage is DBSCAN’s non-linear separation of clusters, facilitating crossing paths resulting in non-linear disjunctions. Finally, DBSCAN is robust to noise due to the triggering and the accuracy of laser scans. In our experiments we use Euclidean distance as the distance metric, with 40 set as the minimum number of points in a cluster.

The next step is to apply *surface modelling* in order to compose a surface that fits the point cloud. The fitting defines the function $y = f(z, x)$, where z is the scaled time dimension and x, y are the Cartesian coordinates with the robot located at $(0, 0)$ facing towards the positive half of the y axis. In this manner, (a) occlusion guarantees that $f(\cdot)$ is a function; and (b) we fit the most informative surface, the one facing the scanner, and not the ‘ceiling’ view that would only give us movement patterns without showing the motion of the legs.¹

¹The implementation used in our experiments is a Python port for ROS of the MAT-

The next step is to extract *Histograms of Oriented Gradients (HOG)*, which are descriptors of the gradients in the image. In our experiments we use the sk-image implementation² of the HOG descriptors proposed by Dalal and Triggs (2005). The extracted features are then used by classification models, where we have experimented with a *Naive Bayes* classifier with PCA pre-processing for reducing dimensionality to independent features, with *Support Vector Machines* under the *Radial Basis Function (RBF)* kernel, and with *Linear Discriminant Analysis (LDA)*.³

2.4 Cluster Alignment

2.4.1 Method

The clusters formed by human walking give the characteristic shape shown in Figure 2. Each of the two traces in that figure tracks a leg and the overall shape results from the human walking pattern where legs alternate between being almost stationary during the *stance* (the parts of the track that rise in the figure) and then moving rapidly during the *swing* (the almost horizontal part of the track in the figure).

The inclination of the imaginary centerline between the two traces gives the walking speed and the projection of this centerline on the spatial plane gives the walking direction. The core idea of our method is that we can improve the classification models by *aligning* movement traces so that their imaginary centerlines have the same pose. If we can do this, we factor speed and orientation out of the point cloud and, subsequently, also out of the resulting surface model and HOG descriptors.

To achieve this, we observe that the data points of each cluster can be approximated by a Gaussian distribution, thus they can be described by a mean value and a covariance matrix. The Gaussian distribution has a direction in the space that can be approached by an ellipsoid. Our aim is to find the main direction of this ellipsoid in order to rotate the data points of the cluster. To do this, we use *Singular Value Decomposition (SVD)*, which decomposes a matrix into three matrices U, Λ, V where Λ is a diagonal matrix and U, V are orthogonal matrices.

SVD fits our approach because we can achieve rotation through the use of the eigenvectors. Specifically, we apply SVD in the covariance matrix of each cluster in order to rotate it in the direction that has the maximum variance. The SVD method is used as a tool for the decomposition of the covariance matrix, to get the eigenvalues and eigenvectors that are necessary for the declaration of the main direction of each cluster.

With the use of SVD method we achieve the alignment of the eigenvalues and the corresponding eigenvectors. By sorting the eigenvalues in a descending order, we use the respective eigenvectors to form the transformation matrix. The maximum eigenvalue represents the main direction of the cluster, thus for its normal distribution too. As the covariance matrix is symmetric, V and U are same. Therefore, U is the linear transformation matrix that we will use for the alignment of the cluster points:

$$X' = U \cdot (X - \mu) \quad (1)$$

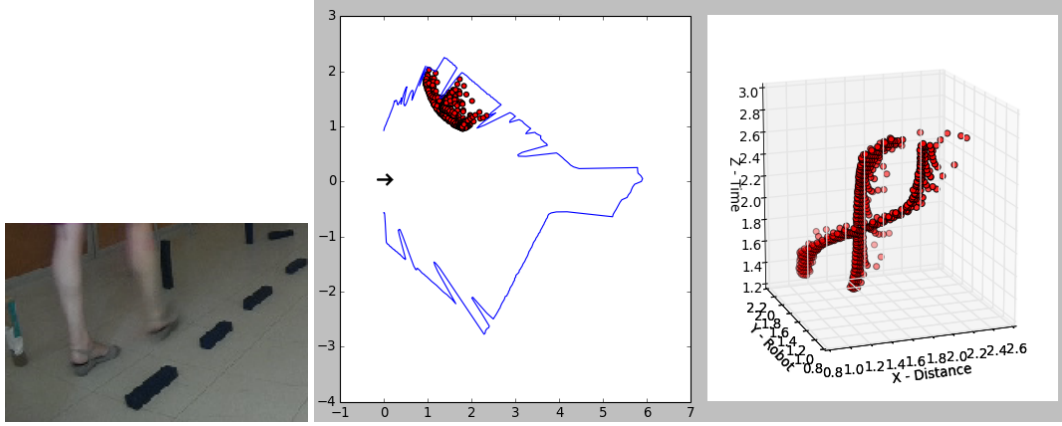
where X, X' are the $l \times N$ initial and transformed data matrices respectively and U is the $l \times l$ transformation matrix. The attribute μ is the mean value of X , thus $(X - \mu)$ specifies a transferring preprocessing step.

As Equation 1 specifies, each transformed data point of the cluster is derived from the projection of the initial data point in the space, where it is declared by the eigenvectors of U matrix. The final alignment of each cluster is computed by transferring its data points to the beginning of the axes and rotating them with the use of the U transformation matrix. The align stage leads to the alignment of the clusters by the same direction, regarding the corresponding variability.

LAB implementation at <http://www.mathworks.com/matlabcentral/fileexchange/8998-surface-fitting-using-gridfit>

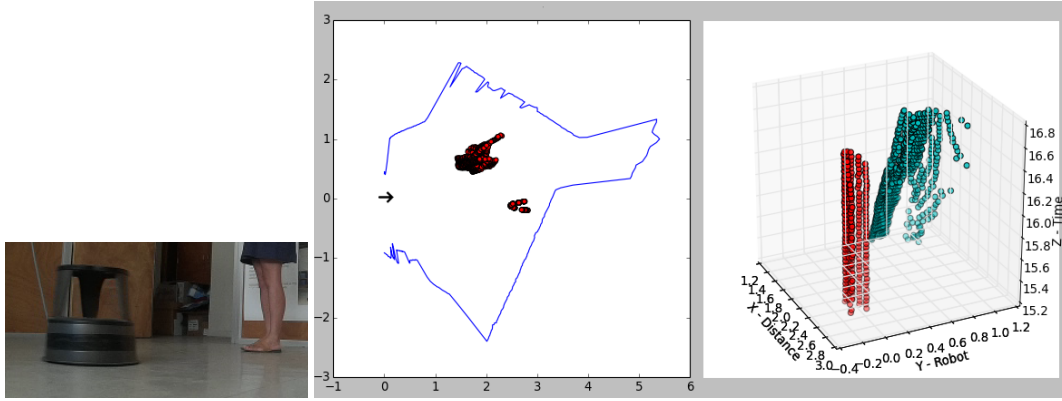
²See <http://scikit-image.org>

³All three, as implemented in Python in the sk-learn library, see <http://scikit-learn.org>



The 3D spatio-temporal representation (right) and the projection on the map (middle) of somebody away from the robot (left). The robot is at $(X, Y) = (0, 0)$ facing towards the positive half of the horizontal axis (X), as marked by the black arrow in the middle image.

Figure 3: Characteristic example of the 3D point-cloud of somebody walking away from the robot.

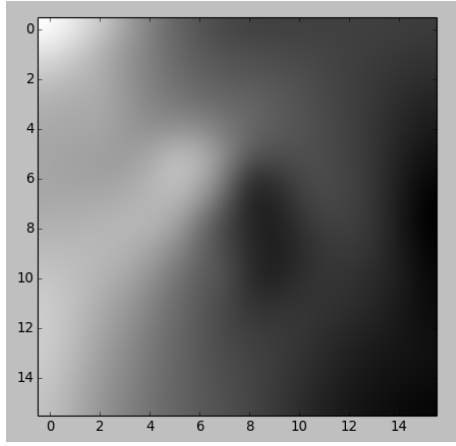


The 3D spatio-temporal representation (right) and the projection on the map (middle) of a stool rolling near a person standing still (left). The robot is at $(X, Y) = (0, 0)$ facing towards the positive half of the horizontal axis (X), as marked by the black arrow in the middle image.

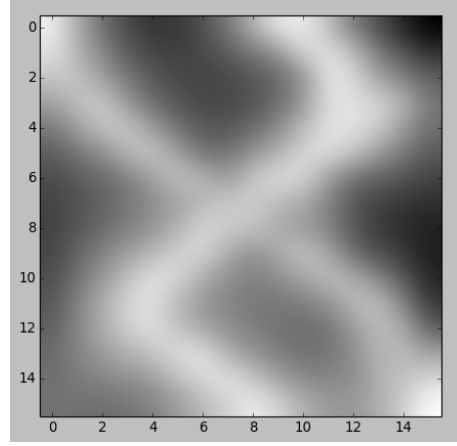
Figure 4: Characteristic example of the 3D point-cloud of non-walking movement.

To demonstrate the effect of alignment on the surface fitted to clusters, consider Figure 3 showing a scene where a person is walking in a different direction relative to the robot than in Figure 2. Although the resulting clusters look visually similar, their different direction makes their similarity less pronounced in their grid images (Fig. 5a and 5c). Alignment emphasises their similarities and transforms the original problem into one that is more suited for classifying using HOG descriptors (Fig. 5b and 5d). What should be noted is that using the direction of the Gaussian ellipsoid of the point cloud to find the direction of the transformation we are expressing a property of natural, uninhibited human gait. In other movement patterns, such as stepping sideways, the largest variance in the point cloud will not necessarily correspond with the direction of movement and will not maintain direction invariance in the HOG descriptors. This is, however, a positive quality of the approach for our use case, as it removes unnatural and special circumstances from our walking speed measurements.

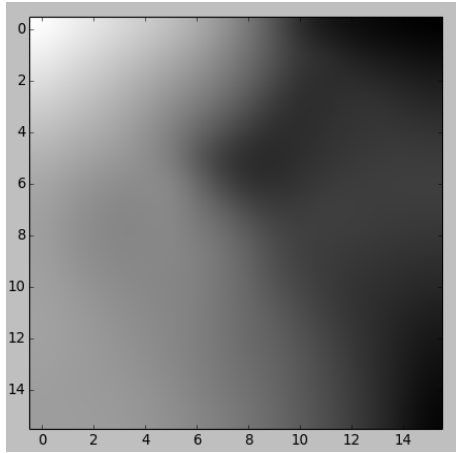
Finally, it should be noted that alignment does not obscure the distinction between human walking and other movement. As an example, Figure 4 shows a cluster from a stool pushed to roll on its wheels. As can be seen in Figure 5e and 5f, the aligned grid image remains separable from the walking pattern grid images.



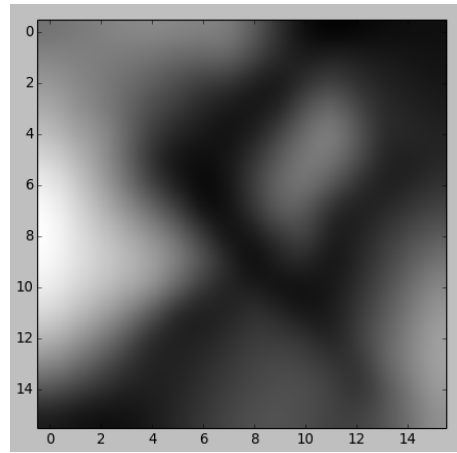
(a) Raw cluster from Figure 2



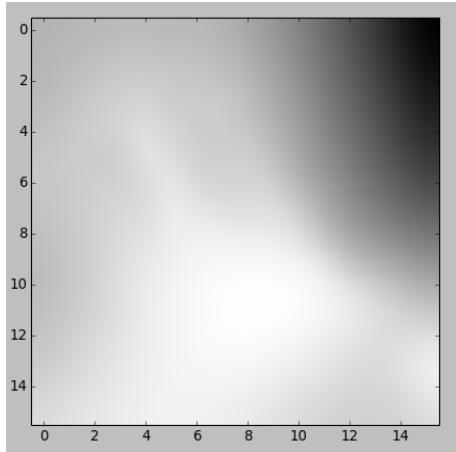
(b) Aligned cluster from Figure 2



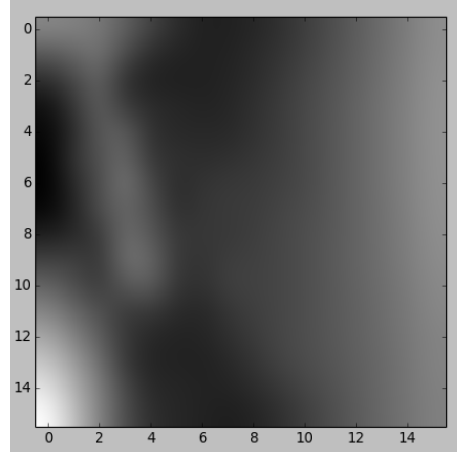
(c) Raw cluster from Figure 3



(d) Aligned cluster from Figure 3



(e) Raw cluster from Figure 4



(f) Aligned cluster from Figure 4

Figure 5: Grid images of the three clusters.

2.4.2 Evaluation

To test our alignment method, we will compare the classification accuracy of the pipeline described in Section 2.3 with and without the alignment pre-processing stage. We have collected scans where people walk with speed and direction changes, periods of standing still, and interacting with furniture in the room (sitting in chairs, picking and putting down boxes, walking between furniture).

Table 2: Evaluation of classification performance with and without alignment

Classifier	Metrics	Without alignment	With alignment
NB + PCA	Precision	100.00%	81.25%
	Recall	22.76%	55.91%
	Accuracy	23.32%	81.27%
SVM	Precision	6.25%	82.81%
	Recall	40.00%	55.79%
	Accuracy	76.67%	81.27%
LDA	Precision	12.50%	76.56%
	Recall	28.57%	80.33%
	Accuracy	73.14%	90.46%

The range finder is a Hokuyo UST-10LX mounted on a robot that is stationary throughout data collection. The following parameters were used:

- The range finder is mounted 12cm above the ground, collecting scans just above the human's ankle
- One scan is obtained every 25msec
- Each clustering frame includes 40 scans, and lasts 1 sec
- Clustering is done on Euclidean distance as the distance metric, with a minimum of 40 points in a cluster
- 36D HOG features are extracted from 16x16 surfaces with the use of 6 histogram bins, 8x8 cell size in pixels and with un-normalisation in the histogram's blocks

We collected 811 such frames, with a duration of 1 sec each, split into 12 scenes. These were randomly split into a training set (70% of the data) and a testing set (the rest of the data).

We used the resulting feature vectors to evaluate binary classification between human walking vs. anything else. We experimented with three different classification methods:

- *Naive Bayes* with *Principal Component Analysis (PCA)* pre-processing for reducing dimensionality to independent features.
- *Linear Discriminant Analysis (LDA)*, a linear classifier that is closely related to Naive Bayes. LDA uses Bayes' rule and the model is generated by fitting class conditional densities to the data.
- *Support Vector Machine (SVM)* under the *Radial Basis Function (RBF)* kernel.

The metrics that we use in order to evaluate our classification experiments are: Precision, Recall and Accuracy. Table 2 presents the experimental results with and without the alignment stage. We can observe that the alignment stage improves performance for all three classification methods. Moreover, we can conclude that LDA classifier is better suited to this task, achieving an accuracy of 90.46%, which is the highest among all our experiments with and without alignment.

What we can also observe in Table 2 is that SVM massively overgeneralizes when presented with non-aligned feature vectors and accepts too many false positives (i.e., low precision). The precision increase under alignment is consistent with our hypothesis that alignment homogenizes and makes separable the HOG feature vectors.

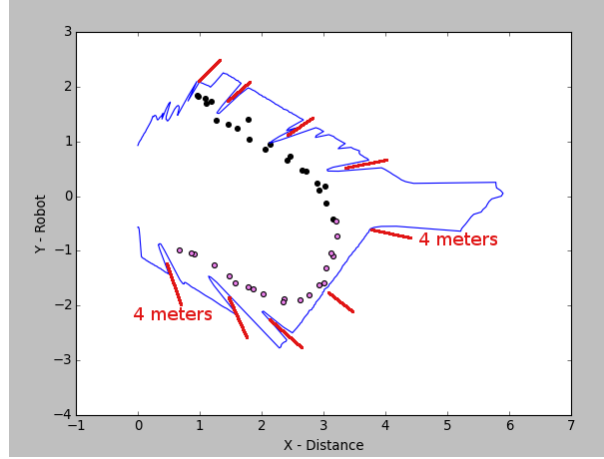


Figure 6: Median points in the plane that show the trajectory of a walking human tracked across several frames. Different colors in the points represent the two 4m spans recognized by the system. The ground truth is derived from markers on the ground set 1m appart with their position given here by the red lines.

2.5 Cluster Tracking

2.5.1 Method

In the original HPR method, the objective was the categorization of objects in laser scans as human walking or other movement, without the requirement to associate the scans across frames. In RADIO, we had the new requirement to measure walking speed over a distance of 4m. Since frames are shorter than the time it takes to walk 4m, we extended the method in order to track the movement of each human in the scene and take measurements from longer sequences of frames.

To achieve this, we re-defined frames so that they overlap in time, by placing two (2) scans in both the previous and the next consecutive frame. In this manner, we have a partial overlap that helps us link each 3D object classified as ‘human walking’ in one frame with the most-overlapping human-walking 3D object in the next frame. We use the Euclidean distance metric to compute proximity. The tracking algorithm is executed in a slide window mode for efficiency, i.e it keeps the k newest clusters for each tracked object.

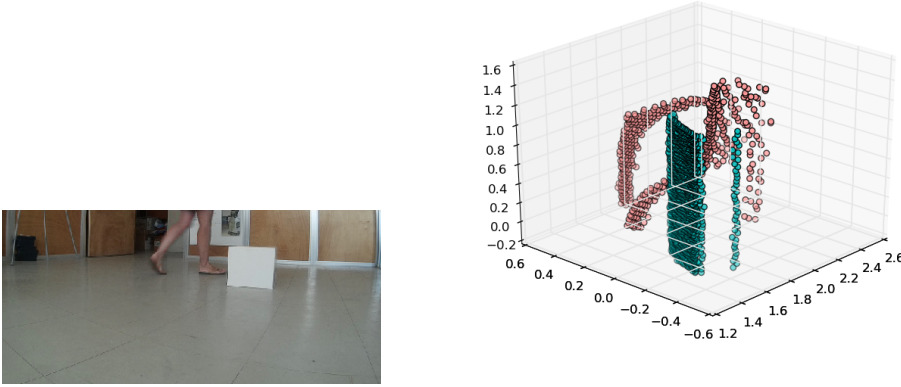
Besides tracking the same person across frames, we also need to reduce these point clouds to a position on the planar map, in order to make the walking speed measurement. To do this, we slice the clusters to segments (thinner than complete frames) along the time dimation. Each segment is reduced to its median 3D point and projected to the spatial plane, so that we get a series of timestamped positions of the person on the planar map. Walking speed measurements are made on this final representation (Figure 6).

The advantage of measuring walking speed as described here (as opposed to directly using the raw scan) is tracking over longer sequences and also making measurements on moving objects that match the normal human gait patterns. If a person stands still in the middle of a 4m span or needs to walk sideways or in general needs for whatever reason to assume a gait that does not match normal walking, those sequences are disregeraded.

2.5.2 Evaluation

To test our proposed methods, we use two different scanning laser range finders (Hokuyo UST-10LX and Hokuyo URG-04LX) in order to collect the range data. The UST-10LX laser scans every 25msec with a maximum range of 10m, while the URG-04LX scans 4 times slower with maximum range 5m. Both are located 12cm above the ground, thus they collect data points above the human’s ankle.

Evaluation is based on recordings of scenarios with challenging proximity situations and occlusions, sudden and smooth changes in walking speed and direction or completely stopping and resuming walk-



Instance of a scenario where the human is walking around the back of a box and the clusters produced by this scene.

Figure 7: Characteristic example of occlusion.

ing, two people crossing each other while walking, and people walking behind or picking up and relocating a box. Figure 7 has a characteristic instance.

The range finders used were the Hokuyo URG-04LX mounted on the original RADIO Robot prototype and the Hokuyo UST-10LX mounted on the current prototype. The robot is stationary throughout data collection. The following parameters were used:

- The average speed of human walk is set to 5 km/h.
- The time scale of UST-10LX is set to 25 msec and of URG-04LX to 50 msec.
- The clustering window is set to 40 scans for UST-10LX and to 20 scans for URG-04LX.
- The minimum distance walked to accept a measurement is 4m.
- Based on the experiments presented in Section 2.4, the LDA classifier performs better, the experiments in this section use the LDA-trained model.

The evaluation metrics are the *Mean Absolute Error (MAE)* and the *Mean Relative Error (MRE)*. MAE gives the mean absolute error between the system-computed and the ground-truth time to walk 4 meters:

$$MAE = \frac{1}{n} \cdot \sum_{i=1}^n |f_i - y_i| \quad (2)$$

where n is the number of the measurements, where f_i are the system-computed values, and y_i the ground-truth values. MRE indicates how good a measurement is relatively to the size of the measured quantities:

$$MRE = \frac{1}{n} \cdot \sum_{i=1}^n \frac{|f_i - y_i|}{y_i} \quad (3)$$

Table 3 presents, for both scanners, the MAE and MRE of the whole dataset and of the dataset excluding the sessions with crossing trajectories. The considerably higher error rate in the crossing scenarios emphasises the importance of correctly handling occlusions and proximity, erroneously associating objects between frames leads to erroneous measurements. Fig. 8 presents the tracked clusters of a crossed human session. We can notice that erroroneous clustering of the points in the circle as ‘red’ radically changes the median point of the ‘green’ cluster and thus of the walking speed computations.

This is to a large extent unavoidable in the laser scanner modality, as clustering can only be based on physical proximity. Subsequent work in WP4 is aim to apply fusion between laser range data and data

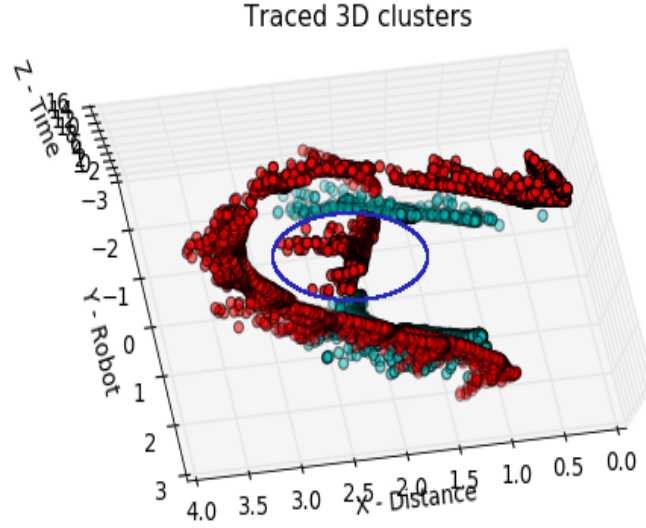


Figure 8: Tracked clusters in a crossed human situation. The different colors represent the different humans. The cluster inside the circle specifies the error in overlapping.

from other sensors. For example, the results from the laser range finder can be combined with visual information to solve difficult proximity cases.

Table 3: MAE and MRE for the walking analysis. The error measures are computed by taking into consideration the whole Dataset 2 and by excluding the crossed human sessions.

Error Metrics	Whole Dataset		Without crossed humans Dataset	
	UST-10LX	URG-04LX	UST-10LX	URG-04LX
MAE	0.89	2.29	0.44	2.75
MRE	0.33	0.59	0.11	0.68

3 MOVING OBJECT TRACKING IN RGB-D

3.1 Overview

In the context of object tracking and identification we used two information channels, the RGB and depth video streams. Our original plan was to apply a *late fusion* methodology, where the RGB and depth modalities are processed independently and their final results are compared and combined to produce a fused movement tracking stream. Experimental practice, however, revealed that the depth modality is too noisy and unstable to be useful in isolation, as it outputs large *undefined areas* under strong lighting conditions or for highly reflective materials.

That led us to experiment with various preprocessing methods for eliminating and smoothing out the undefined areas ('holes' in the image) caused by strong light. Although these preprocessing steps have dramatically improved the quality of the depth data, tracking in depth only was still not robust enough. As a consequence, the architecture changed to one where recognizing and tracking bounding boxes of moving humans is done in the RGB modality only, and depth features are appended to the RGB features of the RGB bounding boxes. The combined RGB and depth features will be used in D3.5 to classify movement into different ADLs.

3.2 Undefined areas elimination and smoothing

For the first part we used a morphological Closing operation that removes small black holes as seen in Figure 9, which was the form that the sensor noise appeared. The result of this method in our data can be seen in Figure 11.

For the second part we used *region growing* techniques to eliminate large undefined areas caused by reflectivity, low resolution, and distance. An example of the method and the results we obtained can be seen in Figures 12 and 14, respectively. On the Region Grow methods we experimented with many variations to get optimal results. Those included different filling patterns (lines, squares, etc.) as well as every possible combination of growing direction (top-bottom, left-right, diagonal, etc.) While the closing operation achieved the desired results, the Region Growing methods introduced another form of error that we did not initially anticipate, merging error areas with humans in the view. That resulted in even more imbalance in our measurements and could not be sustained. In addition in certain cases where the erroneous areas were caused by large distance and were very uniform, the algorithm would not fill the area correctly merging nearby objects with them. The final processed depth image is shown in Figure 15.

3.3 Lighting conditions preprocessing

The initial step in our processing pipeline is to ascertain that our method is independent of the lighting conditions. Of course in all aspects of image processing the change of lighting poses a significant problem.

Initially we applied a standard gamma correction method. In general gamma correction is defined by the following power-law expression

$$I_{out} = I_{in}^{\gamma}$$

where

$$\gamma = \frac{d\log(I_{in})}{d\log(I_{out})}$$

For $\gamma > 1$ we have application of expansive power-law nonlinearity called gamma expansion. In Figure 16 the correlation between the perceived and displayed color intensity is shown. Without correction the

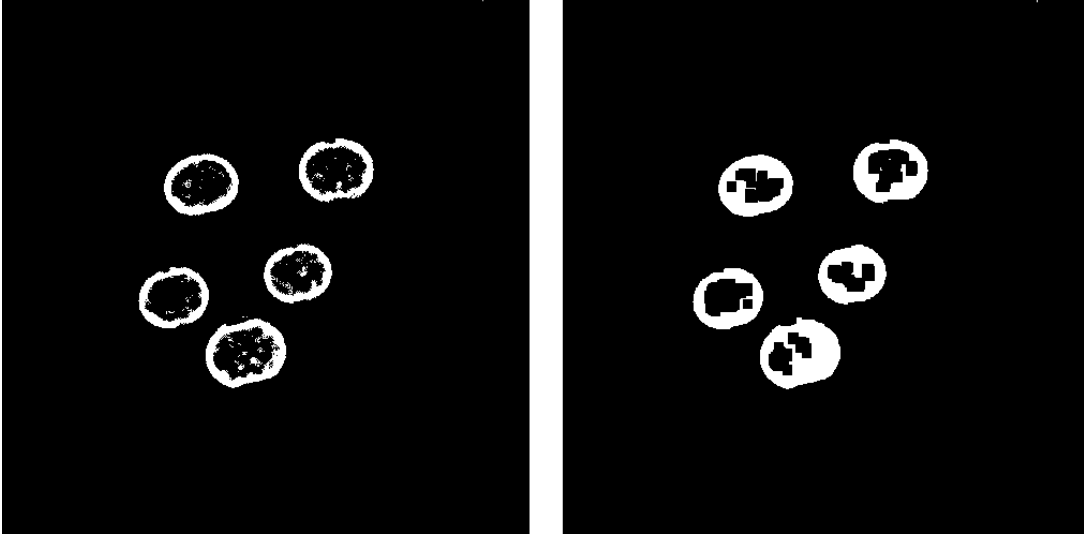


Figure 9: Closing operation example

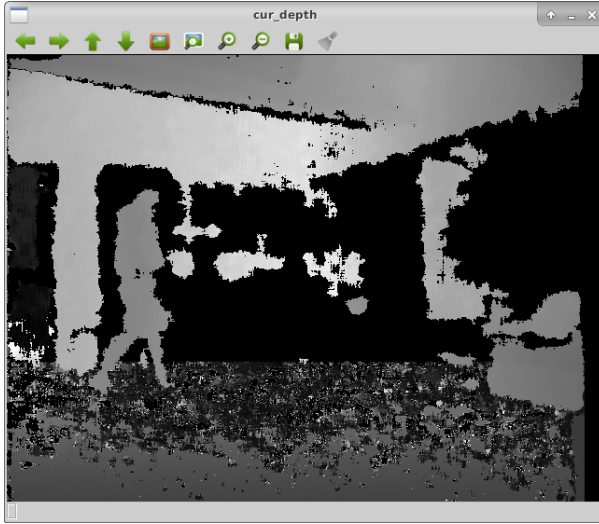


Figure 10: Depth sensor raw image, where undefined areas are given in black. Notice how the human figure is dotted with multiple miniature undefined areas.

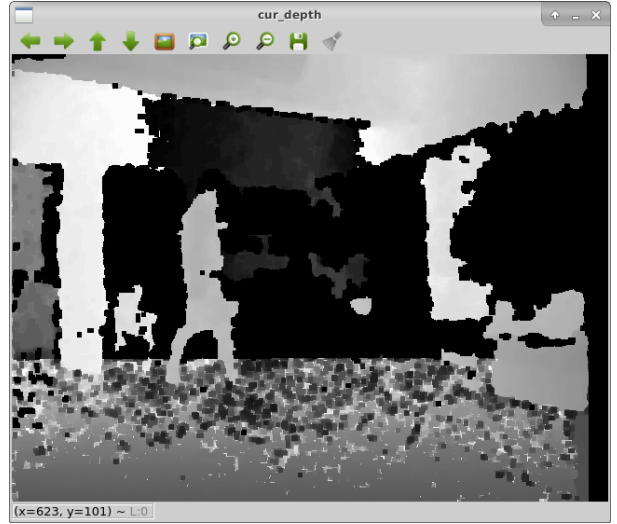


Figure 11: Depth sensor image after Closing operation, where the many small dots in the human figure have been reduced to a single hole.

increase in color intensity is not represented accurately. That leads to a drop in algorithmic performance because color distance operations fail to depict color difference accurately.

After correcting the gamma as seen in Figure 18 we had to balance the lighting on cases of low/high luminance. Also to achieve object detection and tracking we need to be able to perform differential operations between moving objects and the background. Towards that goal it is crucial to have a relatively high level of contrast with as little noise as possible. High luminance creates areas that we have limited visibility while on low luminance the whole image is obscure. In such cases we would get substantially subpar results. We experimented with various methods and the best results were given by Contrast Limited Adaptive Histogram Equalization (CLAHE) to improve the contrast. It calculates a number of histograms as it can be seen in Figure 19 of the image and redistributes the lightness values, which in our case are not on a separate channel.

To avoid noise amplification a limit is selected to reduce the contrast enhancement. As it is shown in Figure 20 the histogram is clipped before computing the cumulative distribution function and values exceeding it are distributed over all histogram bins.

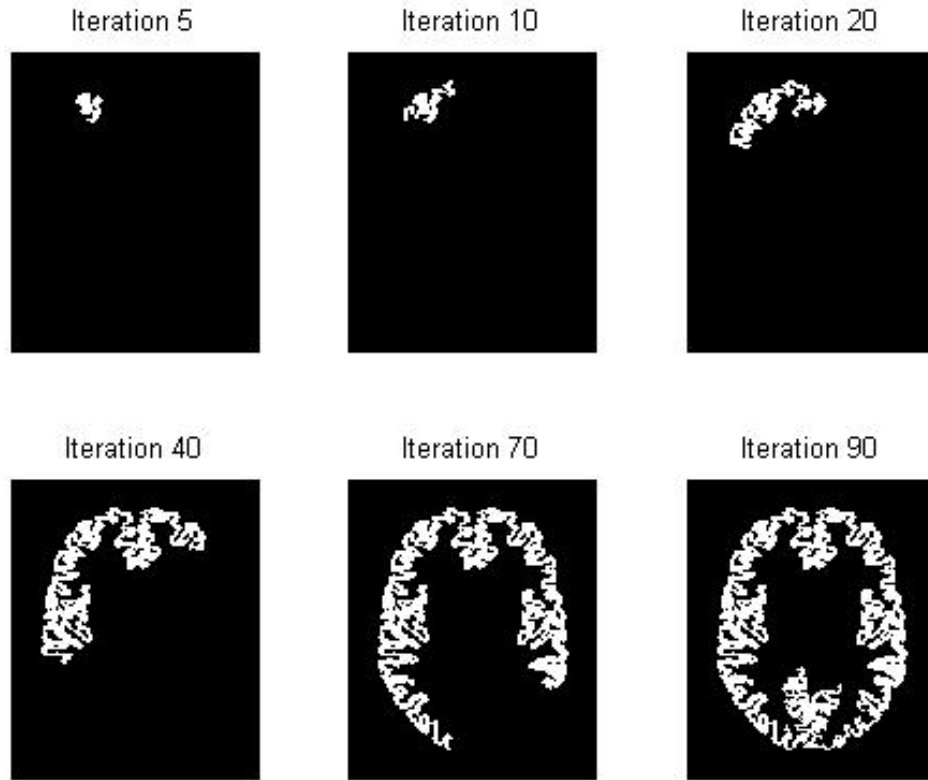


Figure 12: Region grow operation example

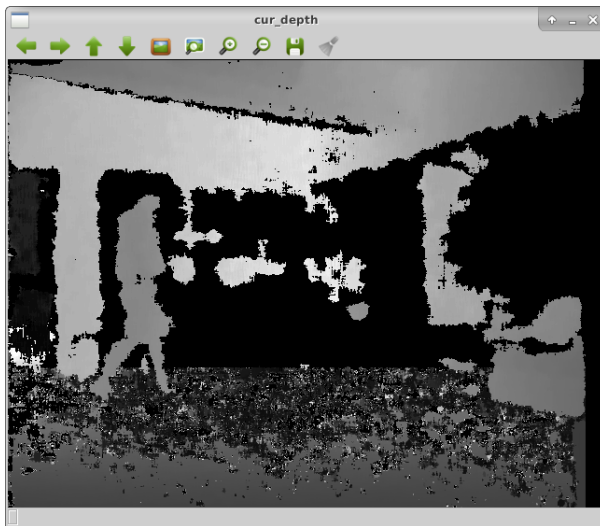


Figure 13: Depth sensor raw image

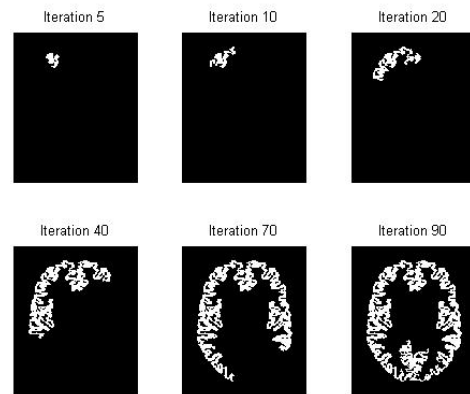


Figure 14: After Region grow

The result of the CLAHE method can be seen in Figure 22 and the final processed image in Figure 23. While the result might look unrealistic to the naked eye, it provides a basis for the execution of our algorithms and the production of stable results.

3.4 Change detection

To achieve change detection we used only the RGB channel for the reasons discussed previously. The main idea was to obtain the differences between successive image frames and use them to detect moving

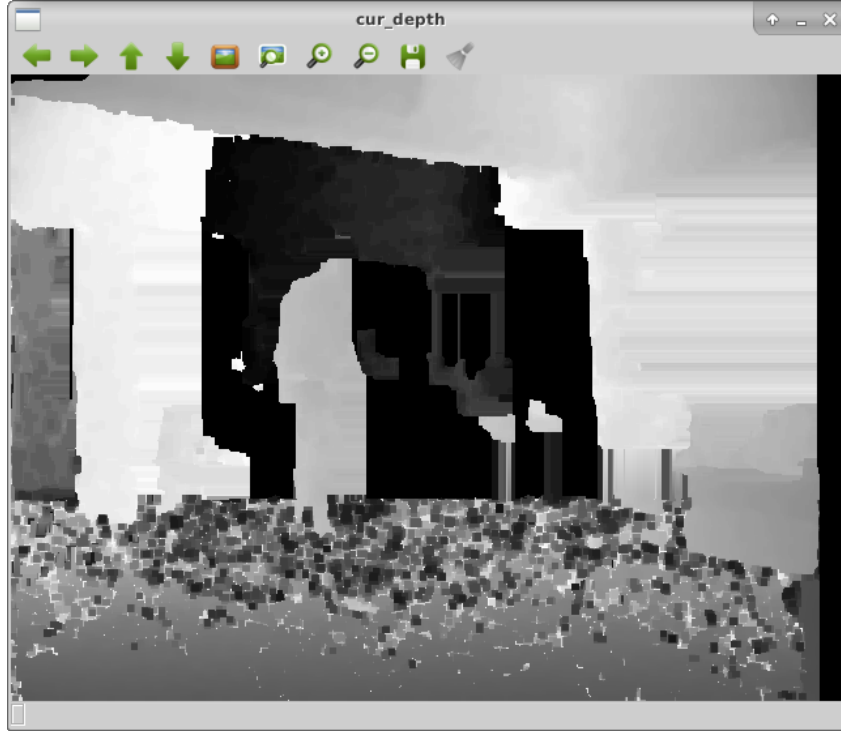


Figure 15: Depth sensor processed image

objects. In practice we saw that we did not require 3-channel RGB images but grayscale since the results were the same and we would save computational power. To detect the differences we would calculate the absolute difference the current and reference image array. Then we would convert the image to binary format using a threshold operation. The reference image array would be a weighted array of images that was updated using the following expression:

$$reference = current * (1 - \alpha) + reference * \alpha$$

where alpha was a predefined weight stemming from experimental results.

That weighted reference image would give more stable results with less noise. An example of a reference array and its respective difference from the current image in binary format can be seen in Figures 24 and 25.

3.5 Bounding box formation

The change detection operation is applied on the frame difference, seen in Figure 25, produced by the preprocessing module and its purpose is to create ROIs (Regions of Interest). This is achieved by iterating through its pixels and defining rectangle areas, for each non-zero pixel, that progressively grow by merging them. The merge criterion for each couple of candidate rectangles depends on the size of their intersection area normalized by the area of the smaller. In particular, we adopt the requirement for the merging procedure, that the area covered by the intersection in the small ROI is positive or their euclidean distance is below a certain minimum threshold. As a final step, an inner merge is applied with the criteria being the existence of positive intersection between the boxes and a limit on the total area of their union. Lastly we filter out ROIs that are too small. The described process in pseudocode is presented in Algorithm 1 and the result on our image can be seen in Figure 27.

3.6 Tracking bounding boxes through frames

The tracking mechanism uses two lists, the stored ROIs list that has all the tracked boxes and the newly detected that has all the boxes detected in this frame. It utilizes the ROIs that are stored and attempts

Algorithm 1 Change Detection

```

1: procedure
2:    $box\_list \leftarrow empty$ 
3:    $n \leftarrow \text{small int}$ 
4:    $distance\_threshold \leftarrow \text{small int}$ 
5:   for  $pixel$  in  $difference\_image$  do
6:     if  $pixel$  is not zero then
7:        $flag \leftarrow True$ 
8:        $roi \leftarrow box(pixel.x, pixel.y, n, n)$ 
9:       for  $box$  in  $box\_list$  do
10:         $distance \leftarrow |box - roi|$ 
11:        if  $roi \cap box > 0$  or  $distance < distance\_threshold$  then
12:           $box \leftarrow roi \cup box\_list$ 
13:           $flag \leftarrow False$ 
14:        end if
15:      end for
16:      if  $flag$  is  $True$  then
17:        insert  $box$  in  $box\_list$ 
18:      end if
19:    end if
20:  end for
21:  for  $box1$  in  $box\_list$  do
22:    for  $box2 \neq box1$  in  $box\_list$  do
23:      if  $box1 \cap box2 > 0$  and  $(box1 \cup box2).area < 2 * area\_threshold$  then
24:        insert  $box\_list \leftarrow box1 \cup box2$ 
25:        delete  $box1, box2$ 
26:      end if
27:    end for
28:  end for
29:  for  $box$  in  $box\_list$  do
30:    if  $(box < min)$  then
31:      delete  $box$ 
32:    end if
33:  end for
34: end procedure

```

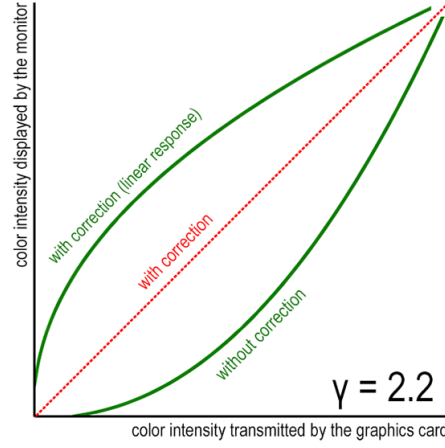


Figure 16: Gamma chart

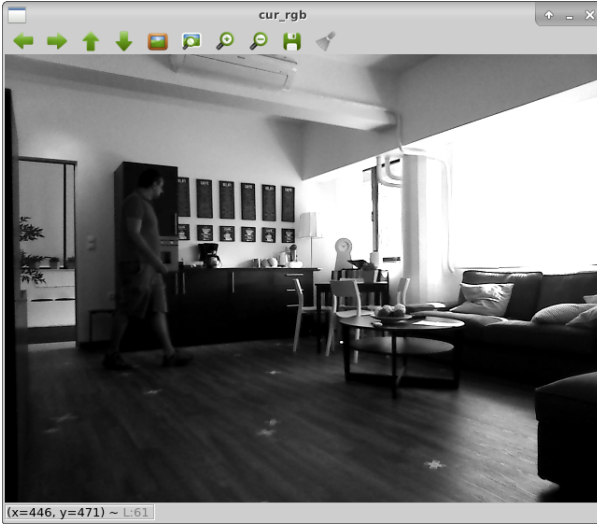


Figure 17: Grayscale raw image

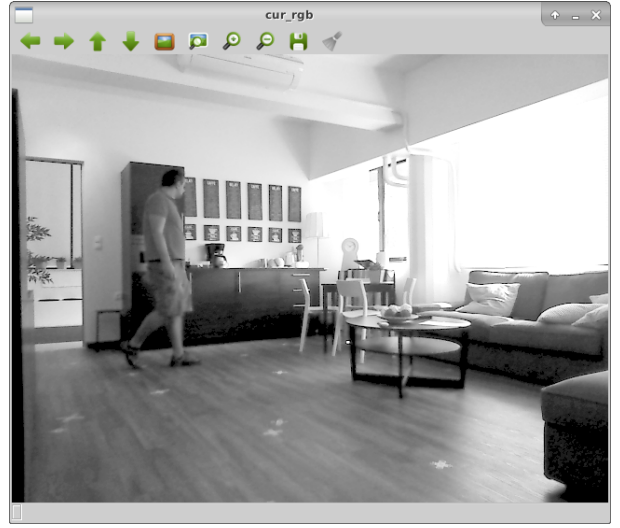


Figure 18: After gamma

to redetect them in each frame. For each stored ROI, it finds those from the newly detected ones who have a positive intersection area and calculates the union between them. If the intersection area is within 10% of the box with the maximum area (between the two) then it is considered a match and that box is removed from the newly detected list. All the detected ROIs that did not match with the stored ones, are inserted into our tracked boxes list and given a default rank. The stored ROIs that were a match are repositioned according to the ROI produced by the union. The reposition rules dictate the relation between the position and size of the two ROIs. Those rules have been developed to model the tracking needs of our human detection scenario. More specifically they become increasingly flexible when the box moves and more rigid as it stays in place. Lastly a merge between the stored ROIs is conducted to avoid false positives. The criteria for the merge are positive intersection area and a rank threshold. That threshold stops the algorithm from merging ROIs with similar ranks as the sum of their ranks gets larger. That aims to avoid merging ROIs with high ranks that probably belong to people in our view that are in contact or overlapping. For every ROI that was redetected we incremented its rank which denotes the redetection. The rank increment is limited by the framerate to be able to adapt in situations where we have low/high framerate due to change of equipment or heavy algorithmic load. Then we proceed to decrement the rank of every stored ROI and delete those whose rank falls below zero. Every ROI that is below a predefined threshold is not used, until we its rank increases through redetection. The described procedure can be seen in pseudocode at Algorithm 2.

Algorithm 2 Tracking

```

1: procedure
2:   for stored in stored_box_list do
3:     for current in box_list do
4:        $temp \leftarrow current \cup stored$ 
5:        $max\_area \leftarrow \max(current.area, stored.area)$ 
6:       if  $(current \cap stored > 0)$  and  $max\_area * 1.1 > temp\_area$  then
7:          $union \leftarrow temp \cup union$ 
8:         remove current from box_list
9:       end if
10:      if  $union\_area > stored\_area$  then
11:         $stored \leftarrow (stored + union)/2$ 
12:      else
13:         $power \leftarrow stored\_area/union\_area$ 
14:         $x\_dif \leftarrow union\_x - stored\_x$ 
15:         $y\_dif \leftarrow union\_y - stored\_y$ 
16:         $w\_dif \leftarrow union\_w - stored\_w$ 
17:         $h\_dif \leftarrow union\_h - stored\_h$ 
18:         $stored\_x \leftarrow x\_dif/power$ 
19:         $stored\_y \leftarrow y\_dif/power$ 
20:         $factor \leftarrow \frac{abs(x\_dif) + abs(y\_dif) + 1}{L * (power + 1)}$ 
21:         $stored\_width \leftarrow abs(w\_dif * factor)$ 
22:         $stored\_height \leftarrow abs(h\_dif * factor)$ 
23:      end if
24:    end for
25:  end for
26:  for current in box_list do
27:    insert current in stored_box_list
28:    initialize current_rank
29:  end for
30:   $max\_rank \leftarrow framerate$ 
31:  for box1 in stored_box_list do
32:    for box2  $\neq box1$  in stored_box_list do
33:       $rank\_ratio \leftarrow (box1\_rank + box2\_rank)/max\_rank$ 
34:      if  $box1 \cap box2 > 0$  and  $rank\_ratio < 1.1$  then
35:         $insert\_box\_list \leftarrow box1 \cup box2$ 
36:        delete box1, box2
37:      end if
38:    end for
39:  end for
40:  for stored in stored_box_list do
41:    decrement stored_rank
42:    if  $stored\_rank < 0$  then
43:      remove stored
44:    end if
45:  end for
46: end procedure

```

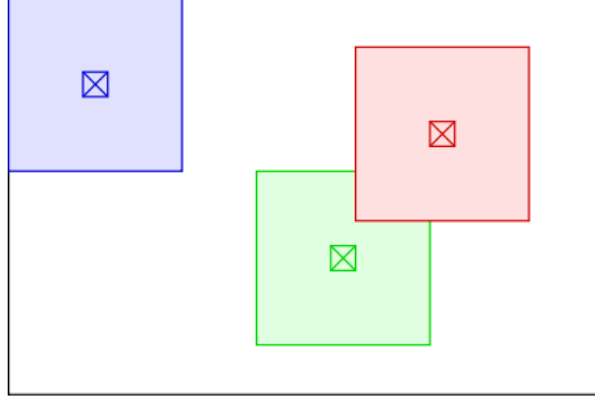


Figure 19: Clahe-histogram selection example

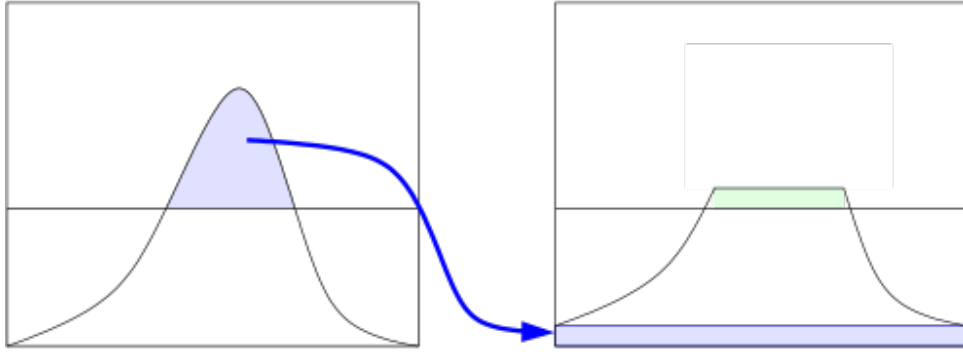


Figure 20: Clahe-histogram clipping and redistribution

3.7 Evaluation

To evaluate the performance of our methods on detecting and tracking moving objects, we manually annotated five videos from the RADIO dataset. The dataset is presented in D3.5.

Figure 4 gives the results on the specific task of recognizing and tracking moving objects, as classifying into specific ADLs will be reported in D3.5. The evaluation methodology is to measure the total area overlap between the manually annotated ground truth and the system-produced ROIs and only consider the system-produced ROIs correct if the overlap exceeds 75%. In that context the performance was almost independent of the environment and the results were correct quick consistently (over 90% of the frames).

As a next step we will try to achieve similar results in a environment with multiple people, fusing (in the context of WP4) multiple channels of information including the laser scanner and smart home information about the number of people in the scene.

Table 4: Evaluation results for 75% overlap

Recall	99.0	97.5	94.9	97.9	96.9
Precision	96.2	97.8	92.5	95.7	93.7

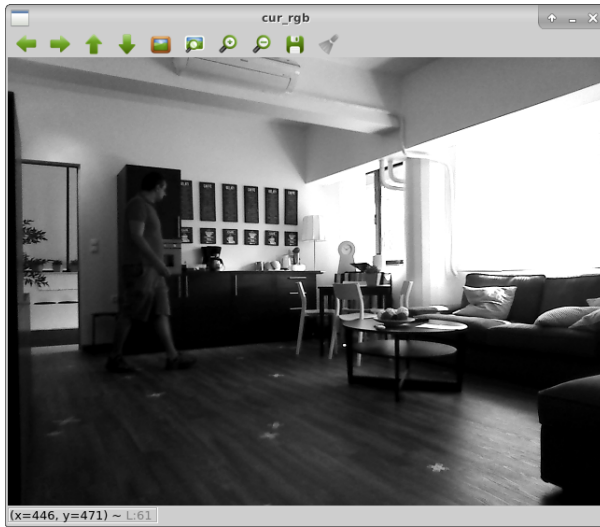


Figure 21: Grayscale raw image

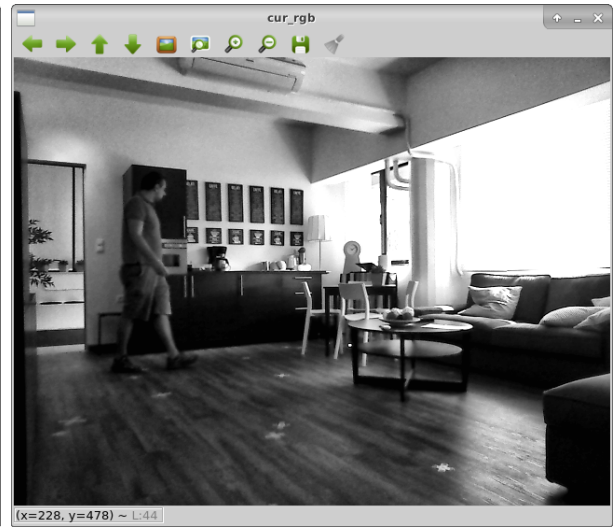


Figure 22: After Clahe

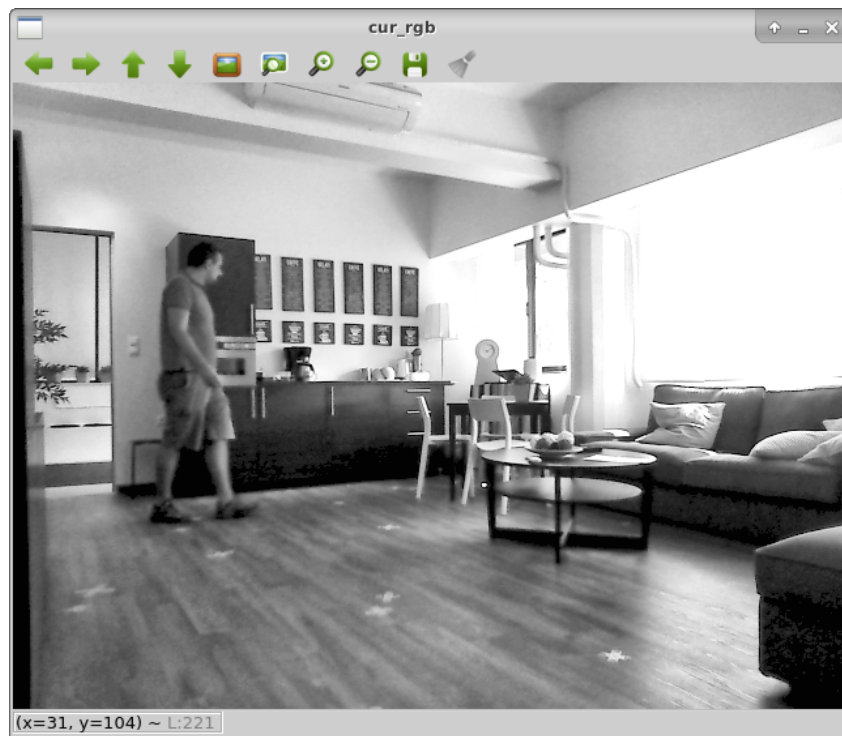


Figure 23: Grayscale processed image

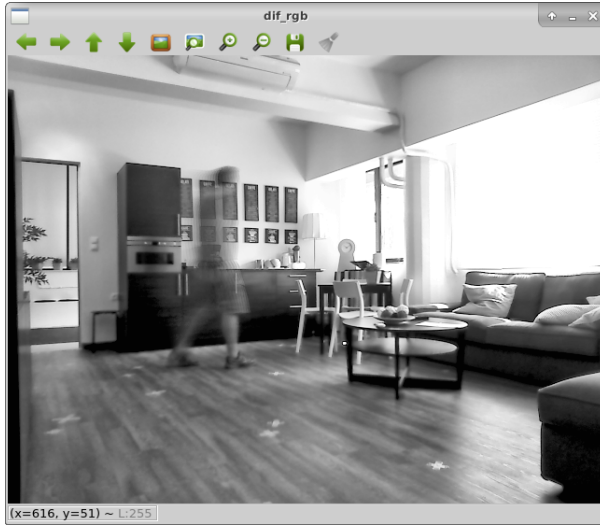


Figure 24: Reference image

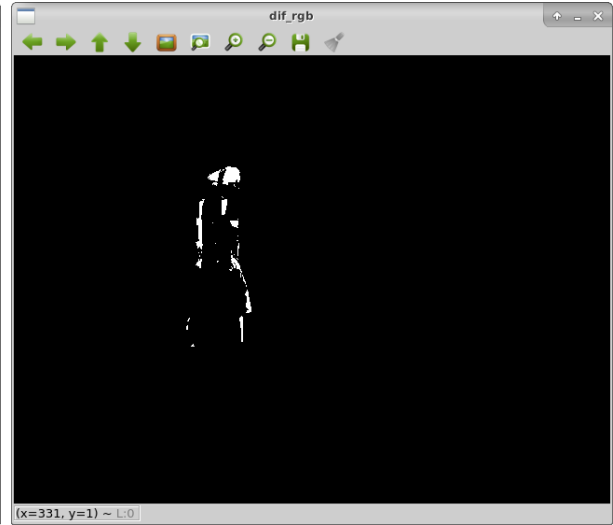


Figure 25: Difference image

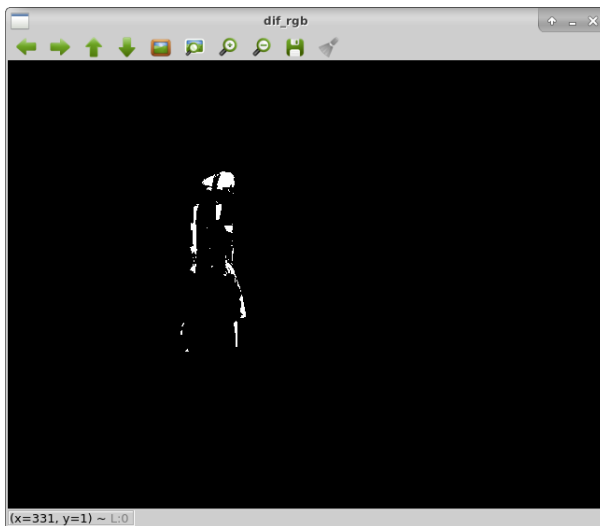


Figure 26: Difference image

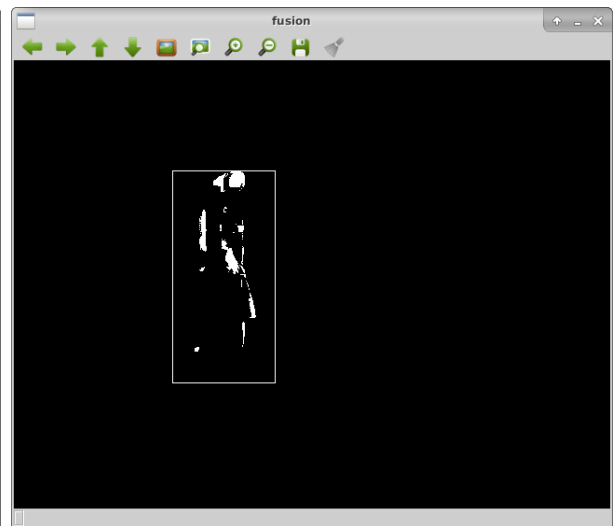


Figure 27: Bounding box formation

4 VISUAL EVENT RECOGNITION

4.1 Overview

In this section, we describe the method proposed for detection of two ADLs:

- Time to stand up from bed and start walking
- Handling of the medication cup

The methods are based only on analysis of visual images as they are provided by the robot camera. In order for both methods to operate reliably, some conditions have to apply:

1. The robot is standing still, in a location defined using installation
2. There is some trigger mechanism to initiate visual recognition. Latency of this trigger mechanism should be as small as possible and fixed

Both methods are based on comparison of the current image frame with some frame from the past. Comparison is performed by accumulating the differences between current and past pixel values in small square blocks.

4.2 Method Description

4.2.1 Time to stand up from bed and start walking

In this method, each frame is compared to its previous frame. Therefore, any block which views a changing part of the image will be marked as changed.

The algorithm finds all changed blocks and calculates:

- Their bounding rectangle, i.e. the region defined by the top, bottom left and right-most changed blocks. The topmost block is the *height* of the bounding rectangle
- Their center, i.e. the average x,y of all changed blocks

The method makes two assumptions:

- That the only changed blocks are caused by the movement of the person. This means that we do not expect strange reflections or shadows, neither do we expect changing light conditions
- That the center corresponds to the center of the person's body. This is not true if the person does not move all his body but just a part of it, i.e. the hands. The algorithm may be tricked.

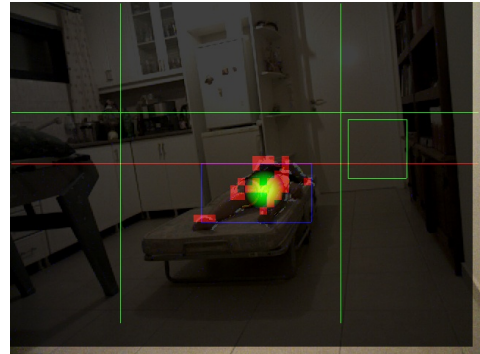
To recognize an event, a region is predefined—during installation—which should bound the bed and set a height limit which corresponds to a standing person's height.

It is important to have the robot camera at a height as close as possible to a standing person's chest (approx. 1m 30 cm). If this is not possible, the algorithm might be tricked by the fact that a person closer to the camera will look taller than a person away from the camera. Experimentation during installation will help define the optimum height limit setting.

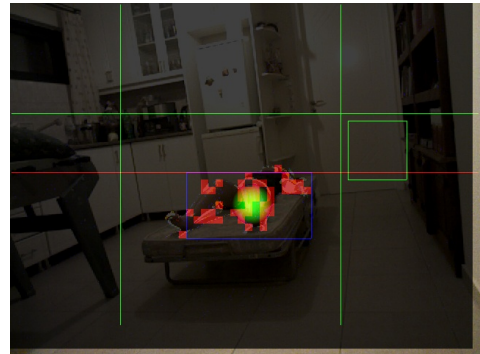
An expected sequence of events is shown in Figure 28.

The time difference between (d) and (a) is the 'Time to start walking' measurement. If the time from (c) to (a) is larger than a predefined timeout, the system should abandon the algorithm; it is probably a person who is moving on the bed, and not one who is going to stand up and walk. This timeout is person-specific, or could be set at a generic default value.

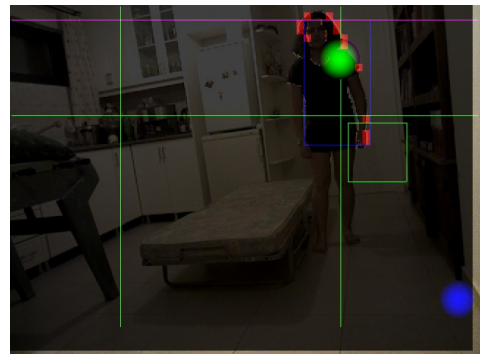
(a) A sensor detects that a person starts moving on the bed and triggers the algorithm.



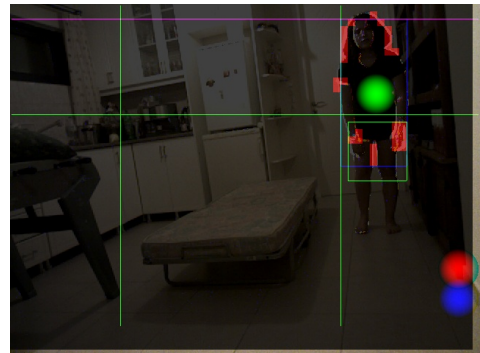
(b) The person moves on the bed and starts standing up from one side of the bed.



(c) When the person is standing up but is still next to the bed, the algorithm reports 'Standing'



(d) When he moves out of the (left or right) boundaries, the algorithm reports 'Walking'



(x) An extra detection is when the person is moving outside the (left or right) boundaries but is not standing up. In that case, a warning is generated by the method.

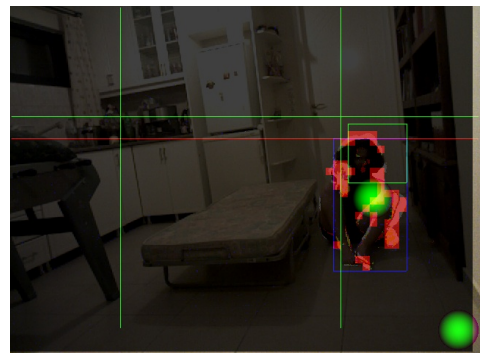


Figure 28: Typical sequence of events

4.2.2 Handling of the medication cup

The medication cup is a cup with pills which is left by the nurse on a small table next to the person's bed. The method assumes that:

- The location where the medication cup is left is more or less fixed. This can be easily enforced by e.g. a clear marking on the surface of the table.
- The person who takes the pills will not leave the cup at exactly the same location.

Therefore, the algorithm simply compares this small region of the image at two time points:

- The first time point is when the nurse who placed the cup moves away. A motion sensor could be used to detect this, but preferably some button press or other confirmative action by the nurse should be used.
- The second time point is a time after e.g. 30 minutes or so, before the nurse gets out collecting the cups. This should be passed as an event to the robot through the smart home infrastructure.

4.3 Evaluation

The main objective of this section is to present the utilization experience of the motion detection based ADL algorithms' first version. In that respect, an evaluation is attempted pertaining to, on one hand, the validation of the algorithms' operation in anticipated movement patterns considering real human subjects and, on the hand, the identification of cases where the algorithm doesn't perform as expected due to conditions and/or scenarios characteristics not taken into consideration during the design/implementation phase. The former aspects is very important since it will signify the current version as the first valid version of the algorithms upon which enhancements and extensions will be made. The latter aspect is also very important since it will serve as roadmap as to how the algorithms need to be extended in next versions.

In all cases the verification will be made through the utilization of two cameras. Firstly, a 3D camera which effectively feeds the ROS based motion detection algorithms. Secondly an independent RGB recording the exact same frame sequences. The second camera is used only as a verification tool in order to be as objective as possible as to what the algorithm identifies as a specific 'event'. In this way we can be sure that cases (although unlikely) will be detected concerning any malfunctions and any imaging delay of the algorithm, in relation to the actual image taken by the simple RGB camera.

4.3.1 Evaluation approach

The two methods need to be evaluated by carefully prepared experiments, which will allow to:

- Measure the reliability of the methods
- Identify possible systematic shortcomings

The reliability is the percentage of properly detected ADLs and is defined by two numbers:

$$R_{total} = N_{correct} / N_{total}$$

$$R_{used} = N_{correct} / N_{percieved}$$

Where:

N_{total} is the total number of experiments executed

$N_{percieved}$ is the number of experiments which the method believes it has correctly detected

$N_{correct}$ is the number of experiments that the method has correctly detected as confirmed by a human operator who is witnessing the same experiments

Although R_{total} seems to be a proper measure of reliability, we prefer to use R_{used} (hence the name) since this allows the method to exclude any experiments that are known to be outside its capabilities.

Variant	Typical	Alternatives
Person	Male, 165cm, 60kgr	Height (150–185cm), Weight (45–120kgr), Male/Female
Position	Foetus	
Movement	Sit on bed, then stand up	
Speed	30 sec	
Direction	Straight away from bed	
Ambient Light	Normal, 'Lights On'	
External events	None	
		Move around on bed, then stand up. Stand up immediately. Get up from the other side of the bed.
		5–120 sec
		Towards camera: Straight away from bed: Diagonal:
		-50% – +120%
		Person walking buy. Lights switch on / off.

Figure 29: Alternative testing settings

In each experiment, a human operator is needed to keep notes. The following paragraphs will give more detail on the experiments for each ADL method.

4.3.2 Experiments to characterize detection of bed transfer

The variants of the experiments are:

- The Person, i.e. a person who is acting as if he/she was the RADIO user
- The Position, i.e. the location and body shape on the bed
- The Movement, i.e. a specific pattern on how the person gets up from the bed
- The Speed, defined on how much total time it takes for the person to stand up
- The Direction at which the person walks away from the bed
- The ambient Light
- External events like e.g. other people passing buy

Each experiment should consist of the following steps:

1. Have the *person* lying on the bed in the defined *position*
2. The person starts stand-up *movement* at the defined *speed*
3. The person stands up and walks away at the defined *direction*

The experiment should be carried out in the 'typical' scenario. Alternative tests should also be executed where all variants have their typical value, besides a 'tested' variant which is also tried at other values (Figure 29).

4.3.3 Experiments to characterize detection of medication cup handling

The variants of the experiments are:

- The Initial Location, i.e. the location where the cup was initially placed by the nurse



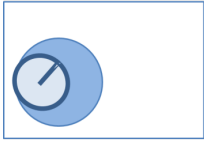
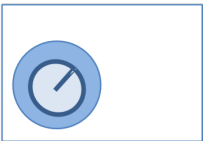
Variant	Typical	Alternatives
Initial Location	Centered	 Centered  Off center
Final Location	Off Center – Rotated	 Off center - Rotated  Centered - Rotated
Ambient Light	Normal, ‘Lights On’	-50% – +120%
External events	None	Person walking buy. Lights switch on / off.

Figure 30: Alternative testing settings

- The Final Location, i.e. the location of the cup after taking the pills
- The ambient Light
- External events like e.g. other people passing buy

Each experiment should consist of the following steps:

1. The nurse leaves the cup at the defined initial location on the table
2. A trigger instructs the robot to take a snapshot
3. The person picks up the cup and leaves it back
4. Another trigger instructs the robot to compare initial and final location

The experiment should be carried out in the ‘typical’ scenario. Alternative tests should also be executed where all variants have their typical value, besides a ‘tested’ variant which is also tried at other values (Figure 30).

4.3.4 Motion detection Algorithms’ Categorizes

In this phase our evaluation considers two types of ADL algorithms.

The first one aims to detect the occurrence of specific movements made by a person. Specifically, as it will be analysed later on, we are interested in detecting specific phases during a person’s rising from the bed and walking away from it.

The second one aims to identify that a person takes the medication assigned to it. To achieve this objective the motion detection algorithm focuses, not on the person itself, but on the cup through which the medication is served to the person. So the algorithm effectively detects whether the cup has been moved or not by the person. Without a doubt such an identification does not guarantee that the person has actually taken its medication. However, identifying that a cup has not moved at all from the time it placed in a predefined position is a pretty accurate indication that the person has not taken the medication which is equally important and useful.

4.3.5 Description of Motion detection Algorithm’s main approach

In all cases the algorithm is applied on frame sequences captured by the camera, with aspect ratios 640 x 480 px and in RGB. In each frame, the algorithm demarcates six main monitoring areas for body

activity, and a seventh area specifically for object detection i.e. the cup previously mentioned.

During algorithm's operation the six areas related to body movement are visualized through a green horizontal and two vertical lateral lines. Respectively, the seventh area regarding the cup movement identification is depicted by a rectangular.

In order for the algorithm to identify specific event it effectively tries to identify the crossing of the body (or the cup respectively) from one area to the other. To do that it identifies the centre of gravity of the moving pixels taking into consideration a window of frames. In order to help us with the evaluation, the algorithm visualizes the centre of gravity as a circular green coloured signal, facilitating the optical verification of an event identification. When the image's changing pixels gravity centre crosses a control area's predefined limits, then it can displays via of the analogue optical signal, the type of event like STANDING, WALKING, OUT OF BED BUT NOT STANDING, etc.

4.3.6 The method of Motion detection Algorithm's trials

As previously mentioned the experiments were based on two cameras including an ASTRA Orbbec mounted on the RADIO Robot providing the frames to the actual motion detection algorithm and a Logitech USB cam independently recording the same frame sequence.

After each scenario was completed the frame sequences were analysed so as to visually compare specific frames related to event identification.

Both ADL algorithms, (1st is Time to stand up from bed and start walking and 2nd is the Handling of the medication cup), are based on simple comparison of the current image frame with previous ones. Comparison is performed by accumulating the differences between current and past pixel values in small square blocks.

Time to stand up from bed and start walking In this method, each frame is compared to its previous frame. Therefore, any block corresponding to a changing part of the image will be marked as changed.

The algorithm identifies all changed blocks and calculates the followings:

- Their bounding rectangle, i.e. the region defined by the top, bottom left and right-most changed blocks. The topmost block is the 'height' of the bounding rectangle
- Their center, i.e. the average x,y of all changed blocks The method makes two assumptions:
- That the only changed blocks are caused by the movement of the person. This means that we do not expect strange reflections or shadows, neither do we expect changing light conditions
- That the center corresponds to the center of the person's body. This is not true if the person does not move all his body but just a part of it, i.e. the hands. The algorithm may be tricked in such a case.

To recognize an event, a region is predefined—during installation—which should bound the bed and set a height limit which corresponds to a standing person's height.

The time difference between the moment the person moves out of the (left or right) boundaries and the moment the sensor detects that a person starts moving on the bed (thus triggers the algorithm) is the 'Time to start walking' measurement. On the other hand, if the time from the moment the person is standing up but is still next to the bed to moment the sensor detects that a person starts moving on the bed (thus triggers the algorithm) is larger than a predefined timeout, the system should abandon the algorithm; it is probably a person who is moving on the bed, and not one who is going to stand up and walk. This timeout is person-specific, or could be set at a generic default value.

Handling of the medication cup The medication cup is a cup with pills which is left by the nurse on a small table next to the person's bed. The method assumes that:

- The location where the medication cup is left is more or less fixed. This can be enforced by green

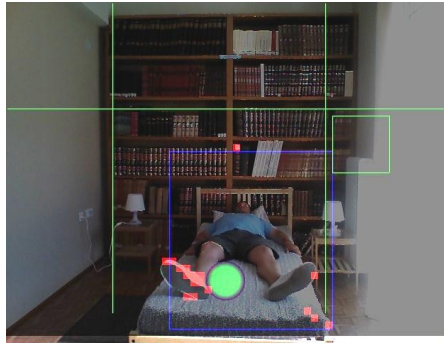


Figure 31: Human activities detection 1st Scenario Startup Phase

napkin on the surface of the table.

- The person who takes the pills will not leave the cup at exactly the same position. Therefore, the algorithm simply compares the seventh region of the image at two time points:
 1. The first time point is when the nurse who placed the cup moves away.
 2. The second time point is a time after e.g. 10 seconds, before the nurse gets out collecting the cups.

Consequently it is possible to identify if the cup has been moved or (and probably more indicative) that the cup has not been moved.

4.3.7 Evaluation scenarios for human activities detection

In this section we present the most indicative cases highlighting how the algorithms behaves in various scenarios, how the algorithms visualizes the events identified and how specific frames and respective indications compare to the second camera recordings.

In all cases both cameras' height is set at 130cm, approximately the level of a standing person's chest.

Default movement sequence Figure 31 indicates the case where a person slightly moves on the bed without, however, not moving towards getting up from it. Firstly the green lines are depicted effectively representing the boundaries the crossing of which signifies a movement event (the green rectangular is not relative for the specific set of experiments). Then with red indication the algorithm indicates all the points that change between two consequent frames. Corresponding to these red indication the green circle indicates the centre of all points changing inside a frame. Here it is interesting to note (and this frame was deliberately selected) specific points in the frames that the algorithm assumes that they have changes but in reality they have not. Such an example is shown in Figure 31 by a red point directly above the person on the bookcase. Apparently this is a point that has not changed but the algorithm identifies it such most probably due to some light condition change. The same applies for some points on the low-right corner of the matrix. The effect of such false identification is that the bounding box (i.e. the blue rectangular) is much wider that it should and consequently there is a deviation of the centre of changed points in the frame.

In Figure 32 the case is captured where the person is moving towards getting out of bed. It is noticed that in this case (a) the only points that the algorithm identified as moving are those of the person's leg (i.e. there are not misidentifications due to lighting effects etc.) therefore both the bounding box and the centre visualization are much more well focused on the person's moving leg. This is the triggering event for the algorithm to identify a person might be starting its 'movement towards getting out of bed initialization timer' as previously mentioned. For verification purposes in Figure 32b the same frame taken by the second camera (offering much brighter frames) is presented. As clearly shown there is indeed a clear movement of the persons' left leg (as we are seen the frame) towards getting out of the bed.



Figure 32: Human's Activities Detection 1st Scenario Initialization Phase

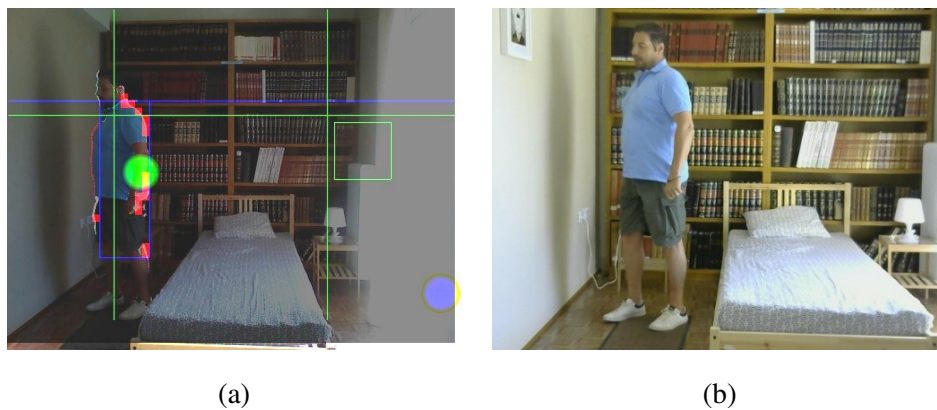


Figure 33: Human's Activities Detection 1st Scenario Standing Event Identification

Now we move to the case where the person is standing up but has not yet walked away from the bed. This is shown in Figure 33a. On one hand the upper points in frame identified as changed have crossed the horizontal green line which is perceived by the algorithms as a person getting in upwards body position. On the other hand the centre of the changed points of the frame remains in between the two vertical green lines corresponding to a person standing right next to the bed and not yet walked away. This case is correctly identified by the algorithm as the blue indication appears which corresponds to the 'STANDING' event. Once again the second camera verifies this event identification since the person has clearly stood up from the bed and resides next to the bed, not yet walked away.

Then we move on to the next event which is the 'WALKING' event as depicted in Figure 34. Assuming that the person has stood up (as presented in the previous phase) the algorithm monitors if the centre of changing points in the frame crosses either of the vertical green lines corresponding to the area indicated as 'close to the bed'. As shown in Figure 34a, the centre of changed points has crossed the left vertical line and conjunction to the previously identified 'STANDING' event the new 'WALKING' event is signalled, visualized through the red circular indication. As verified by the second camera, indeed the person monitored has started to walk away from the bed so the event identification is considered a successful one.

Analysis of default movement sequence The main objective of this scenario is to highlight all the expected phases of a person's body movement and validate that indeed they are successfully and timely captured leading to correct event identification. Following numerous repetitions it is shown that when the limits (i.e. green lines) are correctly setup and the person performs the movements as he/she is anticipated, the algorithm always follows the correct sequence of phases which also means the delay measurements are also correctly captured. A point of attention worth noting has to do with the influence

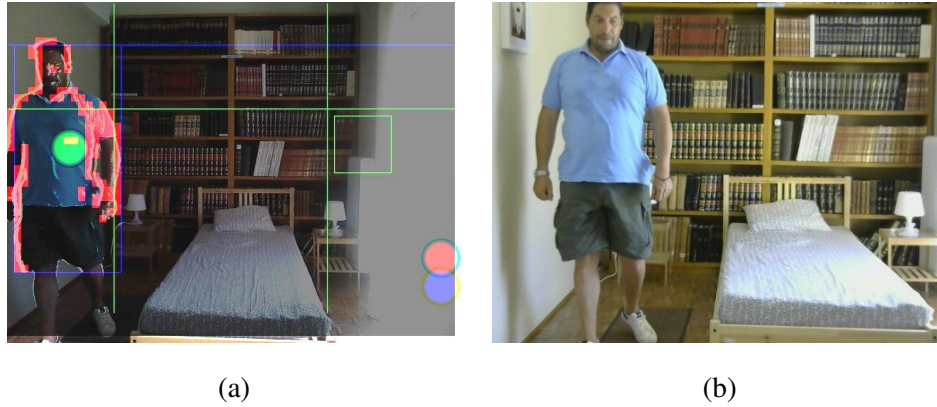


Figure 34: Human's Activities Detection 1st Scenario Walking Away Event Identification

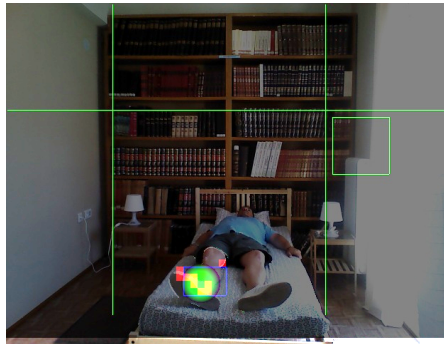


Figure 35: Human activities detection 2nd Scenario Startup Phase

of environmental conditions (i.e. light conditions and/or other objects and persons moving in front of the camera) on the algorithms efficiency since such conditions affect the points in frame identified as 'changed' and this the bounding box and centre of changed points upon which all following events identification are based.

Default movement sequence with increasing speed In this 1st scenario the persons is moving at normal to slow speed as expected to encounter in an elderly people facility. However we tested the algorithm multiple times with the person performing the expected movement sequence but with increasing speed to evaluate if the algorithm in any case did not go through the expected operational phases as they are identified in the 1st scenario. Of course for reasons of space we present frames only from the measurement where the persons moved with maximum speed we think is beyond anything expected to encounter in real everyday life scenarios.

As in the previous case we start by the persons slightly moving on the bed with no clear indication that he will get up from the bed. When there are no environmental conditions influencing the algorithm bounding box and center of changed points are well focused on moving part (Figure 35).

Then in Figure 36 the phase is depicted where the persons has clearly started to stand up from the bed. The algorithm has been triggered and respective delay measurement timer initiated. For verification purposes of the person's moving speed we can see in Figure 6b that the camera is not well focused on the persons arm due to his moving speed.

Then in Figure 37 we see the case where the person has stood up from the bed and is right on the edge to cross the left vertical line separating the area close to the bed to the area away from bed. Once again the algorithm successfully identifies this phase since only the 'STANDING' signal is triggered as expected.

Then immediately after the center green circle of changed points of the frame cross over the 'away from bed' area the 'WALKING' signal appears as expected.

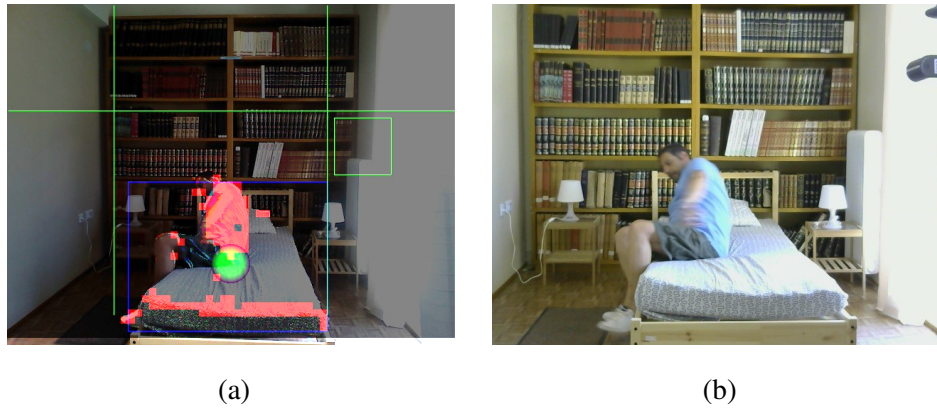


Figure 36: Human's Activities Detection 2nd Scenario Initialization Phase

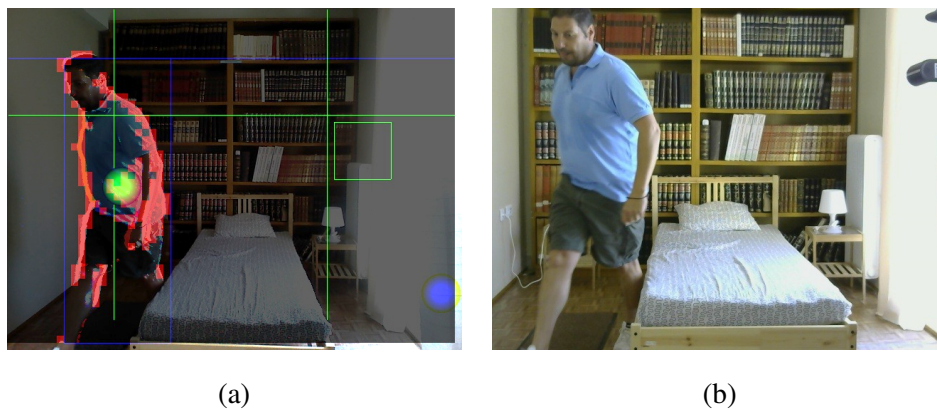


Figure 37: Human's Activities Detection 2nd Scenario Standing Event Identification

Analysis of default movement sequence with increasing speed The main conclusion from 2nd scenario is that the algorithm was able to identify all expected operation phases regardless of the speed the persons moves. Specifically considering that in the 1st scenario the transition from phase a) to d) required 380 frames, in the 2nd scenario we tested cases where the same process required approximately 150, 130 and 100 frames i.e. the person moved faster by 2.5, 2.9 and 3.8 times respectively.

Movement sequence not crossing the upper horizontal limit Both the introductory analysis and the previous scenarios clearly indicate that for the motion detection algorithm to efficiently operate, a specific sequence of events related to the predefined areas (denoted by the horizontal and vertical green line) must apply. Consequently, a question arises regarding the algorithms behaviour when these requirements are not met. In the specific scenario we assume the case where the person when standing up does not cross the upper horizontal line. Such case can arise for numerous reasons including poor horizontal line configuration with respect to the person's high (i.e. the persons is not as tall as anticipated) as well as the case where the persons is getting up from the bed but not in a full standing up position of his/her body due to some condition which is to be expected especially regarding elderly people.

Once again the test starts with the persons moving on the bed. In this case there is strong influence from environmental parameters actually tricking the algorithm to identify points that are not moving as changed. This leads to the case where the bounding box even surpasses the right vertical green line. The center of changed points, however, is still in expected limits so there is no false indication (Figure 39).

In the next phase (Figure 40) the algorithm identifies that the persons is starting to get up from the bed as in previous cases.

In phase indicated in Figure 41 the first false indication of the algorithm is observed. In this case although the persons has clearly gotten out of the bed (Figure 41b) the algorithm fails to identify it and triggers

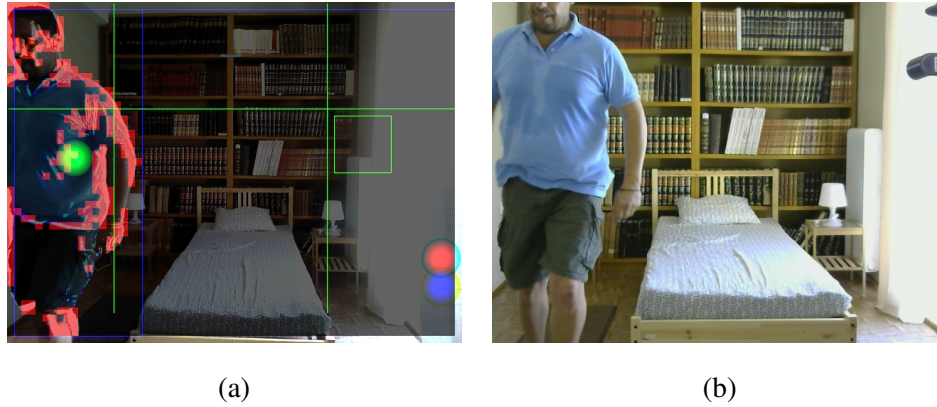


Figure 38: Human's Activities Detection 2nd Scenario Walking Away Event Identification

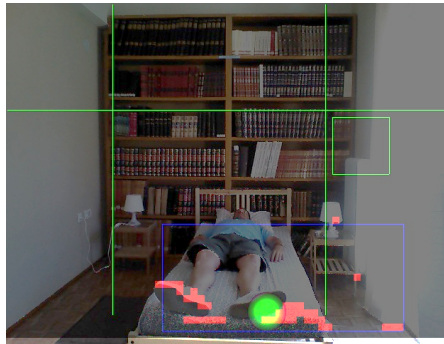


Figure 39: Human's Activities Detection 3rd Scenario Startup Phase

the appearance of the expected blue signal. Of course the reason is that the upper changing points of the frame do not cross the limit indicated by the green horizontal line.

Then we move on to 'walking away' phase (Figure 42) where once motion detection algorithm fails. In this case center signal crosses the left vertical green line which should trigger the 'WALKING' event. However, since there wasn't a previous 'STANDING' event the algorithms fails to identify the sequence and issues a warning signal corresponding to an algorithm failure.

Analysis of movement sequence not crossing the upper horizontal limit In this case it is highlighted that correct configuration of the horizontal limit corresponding to the 'STANDING' event is of paramount importance. Consequently, this is a personalized parameter specifically tailored to the specific user considering his/her body dimensions or/and specific conditions that could prevent him/her from standing to a full upwards body position. In any case, based on how the algorithms are designed and developed if this condition is not met the algorithm fails to successfully identify all event.

Inadequate configuration of vertical limit Another aspect we aimed to test in this set of experiments concerns the case where the configuration of vertical limit is such that there is no real space for the algorithm to distinguish between the 'STANDING' phase and the 'WALKING' phase. Practically this can happen if the vertical limitation is placed too close to the bed prohibiting the algorithm to separate the two distinct phases. Thus this is the case explored in this scenario.

Once again the sequence starts with the phase of slight movement on the bed leading the center of changed points in the frame to indicate the moving part of the person (Figure 43).

As shown in Figure 44, now the person moves towards getting up from the bed where the respective area vertical threshold is deliberately placed right next to the bed. The movement direction is clearly verified by the second camera.

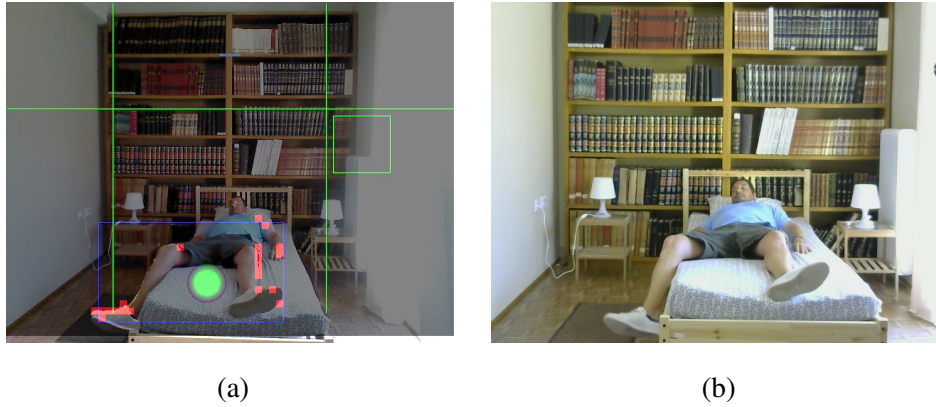


Figure 40: Human's Activities Detection 3rd Scenario Initialization Phase

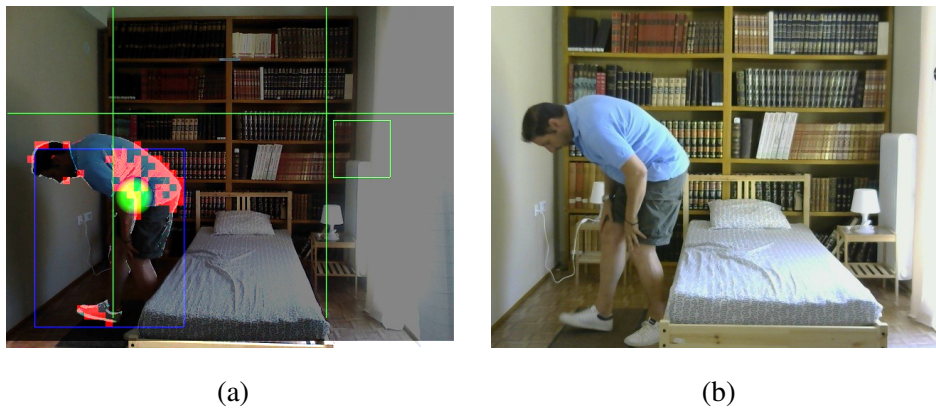


Figure 41: Human's Activities Detection 3rd Scenario Standing Event Identification Failure

The main difference is depicted in Figure 45a and verified in Figure 45b where the person's movement pattern triggers effectively concurrently both the 'STANDIND' and 'WALKING' events. However, in this particular case the triggering of 'WALKING' can be considered misleading since as depicted in Figure 15b the person has not started to walk away from the bed and may not start walk away from the bed. Also cases where recorded where the center signal crossed the vertical limit before the person crossed the horizontal line signaling the standing event. In this case the behavior followed the pattern presented in the 3rd scenario and the algorithm effectively failed.

In Figure 46 the person starts walking away from the bed the triggered events appear to be correct since both 'STANDING' and 'WALKING' are triggered. But as mentioned in step c) if the person did not start to walk away the indication could be erroneous.

Analysis od vertical limit configuration The fourth scenario effectively validated that the appropriate definition of vertical and horizontal are threshold probably comprise the most critical factor of the algorithms considered. In this particular case it is noticed that when the space between the bed and the vertical threshold indicating the 'away from bed' area the two main events are triggered virtually simultaneously which can lead to misleading visualizations by the algorithm tested.

4.3.8 Evaluation scenarios for object movement detection

For the particular algorithm evaluation the specific object consider is a medication cup. The medication cup is a cup with pills which is left by the nurse on a small table next to the person's bed. The method assumes that:

- The location where the medication cup is left is more or less fixed. This can be enforced by e.g. a clear marking on the surface of the table. In our case we used a small green napkin.

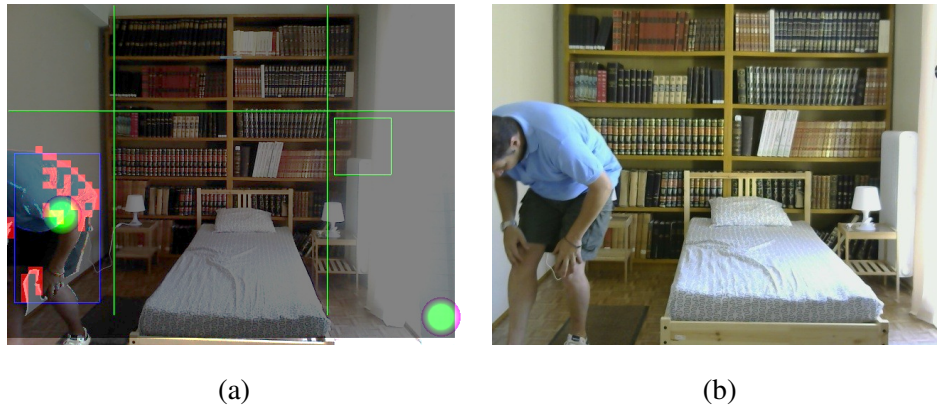


Figure 42: Human's Activities Detection 3rd Scenario Walking Away Event Identification Failure

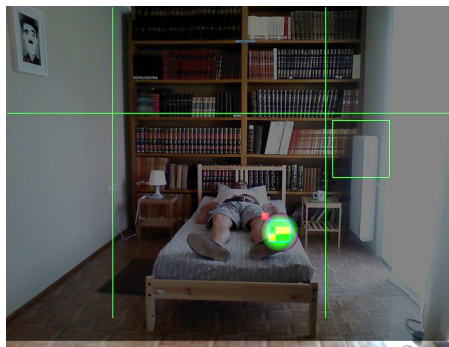


Figure 43: Human's Activities Detection 4th Scenario Startup Phase

- The person who takes the pills will, most probably, not leave the cup at exactly the same location. Therefore, the algorithm simply compares the seventh region of the image at two time points:

(c) The first time point is when the nurse who placed the cup moves away.

(d) The second time point is a time after e.g. 10 seconds, before the nurse gets out collecting the cups.

The variants of the experiments are as follows:

- The Initial Location, (for our scenario the cup placed on the table beside of the bed)
- The Final Location, the location of the cup after taking the pills. In the context of the presented evaluation various assumptions were and placed the cup in different positions from initial position)
- The ambient Light: In our scenarios the ambient light was stable without changes.
- External events: On the first scenario we placed our hand in the control area and we got out this, without handling the cup.

Each experiment should consist of the following steps:

- The cup is left at the defined initial location on the table
- A trigger instructs the robot to take a snapshot
- The cup is picked up and left back
- Another trigger instructs the robot to compare initial and final location

In the specific scenario all indications related to person's movement identification (i.e. green threshold lines and light visual indications) are out of scope and not considered.

1st Scenario (The effect of different final locations)

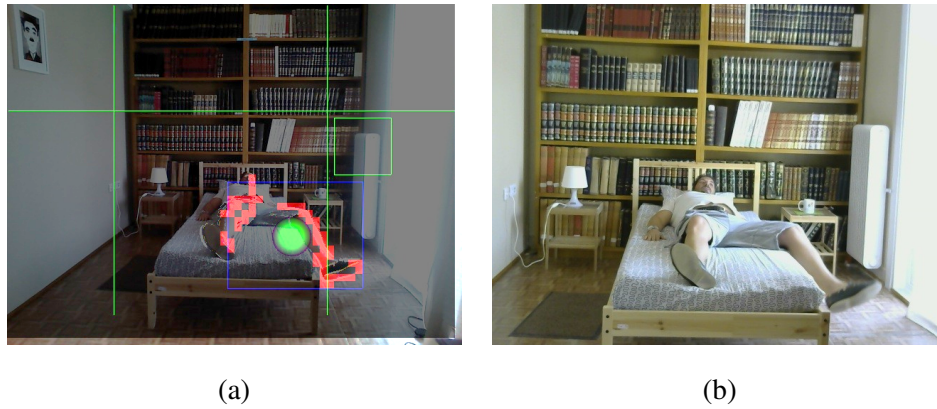


Figure 44: Human's Activities Detection 4th Scenario Initialization Phase

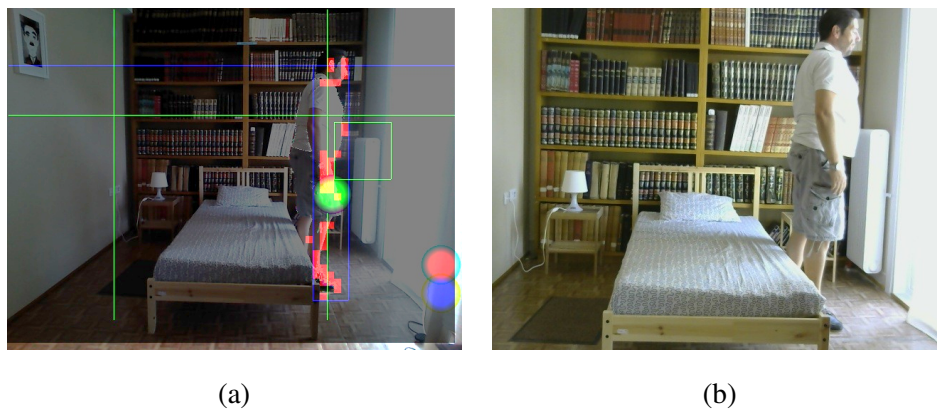


Figure 45: Human's Activities Detection 4th Scenario Standing Event Identification Failure

This is the most basic test in the context of which we trigger the algorithm identifying that the cup has moved and try different ‘final’ positions of the cup until we can find a place where the algorithm identifies that the cup has not moved.

(a) In Figure 47 the initialization phases is depicted where, as instructed, we placed the cup on the table and we start the algorithm.

b) As a first test case depicted in Figure 48 (a and b) we placed our hand in the control area (a) and we got out without actually moving the cup (b).

When the hand was inside the green rectangular the algorithm identified that something was changed but when the hand was removed from the control area the algorithm identified that nothing has changed i.e. the cup has not been moved, which is a correct reaction.

c) In the second test the objective is to evaluate how accurately would the algorithm identify that the cup has been removed and not placed at exactly the same position. Firstly we placed the cup on the table and activated the algorithm to capture the contour of the cup (Figure 49a). If we move the cup outside of the controlled area Figure 19b and reposition back, then the algorithm performs a comparison between the initial and final position. We make a test with the reposition of cup in three different positions (Figure 49c, d, e). Finally, we were able to place the cup at exactly the same position as the initial one and the algorithm identified that the cup has not been moved (Figure 49f).

Scenario Analysis

After the cup was placed in the control area defined by the green rectangular the algorithm was able to identify if a change is made inside this control area. Therefore, if the only thing changed in the control area is related to the cup (or any object for that matter) the algorithm can deduce whether the cap has

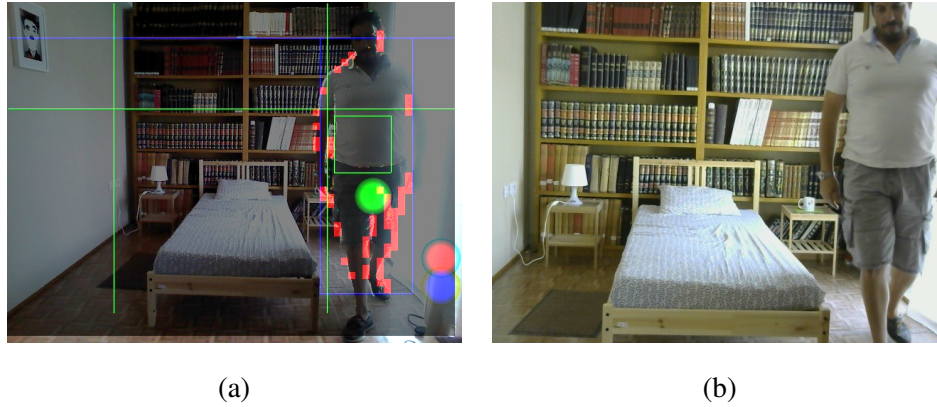


Figure 46: Human's Activities Detection 4th Scenario Walking Away Event Identification Failure

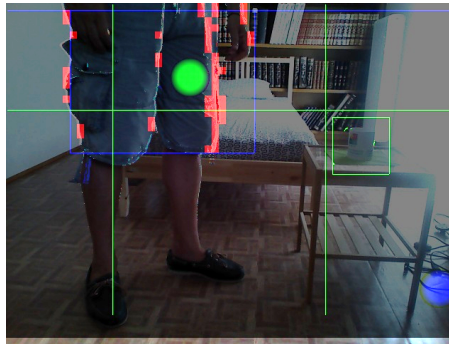


Figure 47: Cup Movement Detection Algorithm Initialization Phase, 1st Scenario

been moved or not.

However there are two cases worth noting. The first concerns the scenarios where the change made is not related to the cup. Specifically, we noticed that when the person's hand enters the control area the algorithm identified that something has changed. So if the final moment is selected while the hand is in the control area the algorithm will deduce that the cup has been moved, whereas in reality it has not. Another interesting observation is that we were able, relatively easily, to place the cup at a position that the algorithm assumed is the same as the initial one. Therefore, the possibility exists that the user has actually used the cup but the algorithm doesn't identify it because the cup was placed at exactly the same position. Respective probability is however expected to be low in real-life use cases.

2nd Scenario (The effect of distinct characteristics of the cup)

Driven by the fact that we were able to place the cup at a position that the algorithm assumed to be exactly the same as the initial one this test aims to evaluate whether specific characteristics of a cup (specifically the cup handle) decreases the possibility of placing the cup at exactly the same position. The idea here is that the cup's contour recorded by the algorithm at initial position is more complex compared to the previous case so it increases the probability that the contour at final position will not be exactly the same.

We placed the cup on the table with handle to be visible and we start the algorithm (Figure 50).

When the cup is moved outside the controlled area the algorithm identifies the change (Figure 51a). Then the cup is placed inside the controlled area and the algorithms continuously compare the snapshot taken from the initialization phase to the last one. The fact that during the initialisation phase the handle of the cup was taken into consideration for the contour formation increased the probability of the algorithm identifying the cup movement, although the cup was placed in the same position as the initial one (b). After numerous attempts the 'user' managed to place the cup at such a position and angle so that the

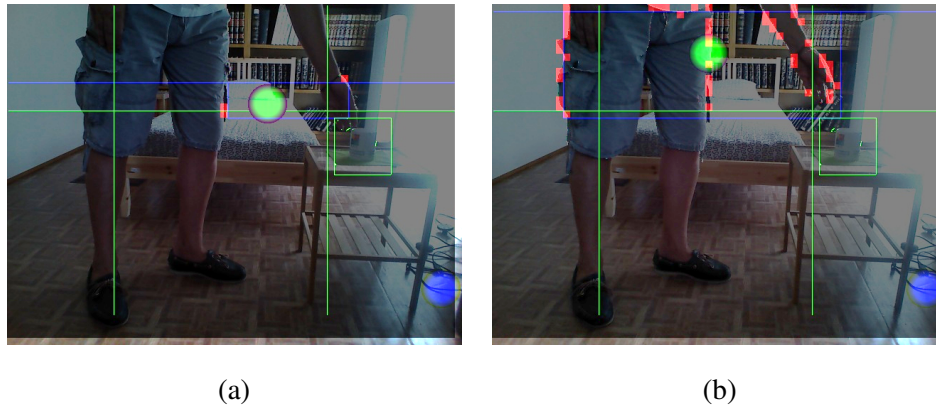


Figure 48: Cup Movement Detection Algorithm Test of not moving the Cup, 1st Scenario

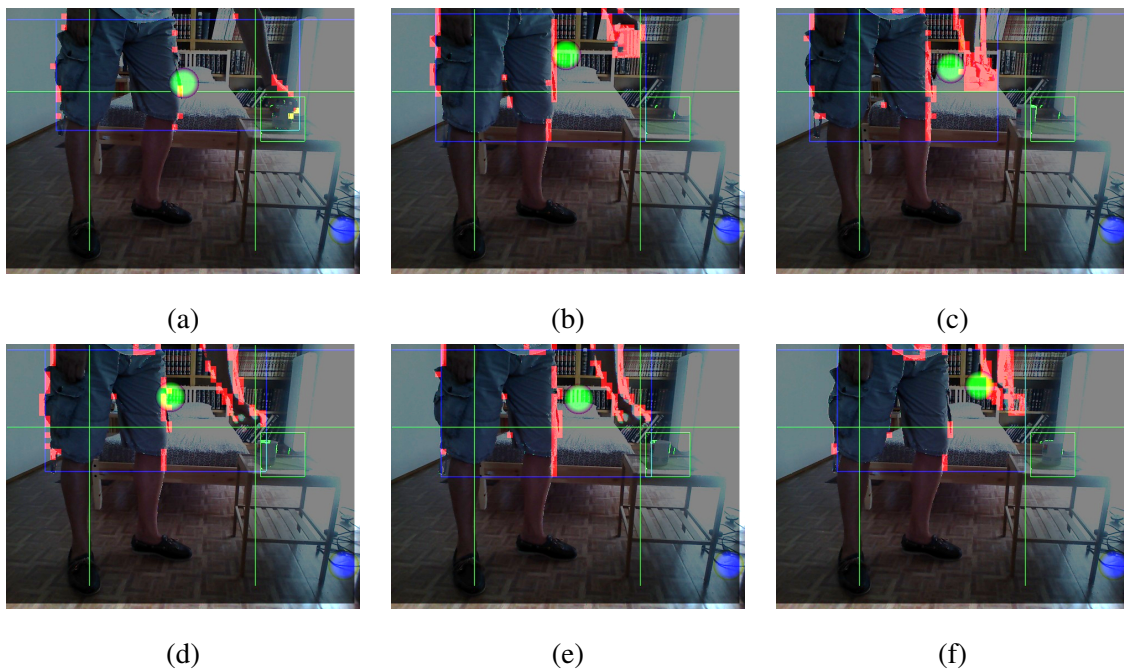


Figure 49: Cup Movement Detection Algorithm Test of Expected Sequences of Using the Cup, 1st Scenario

algorithm was not able to identify the cup displacement i.e. the pills intake by the cup ‘user’ (c).

Scenario’s Analysis

As with the 1st scenario the algorithm after being initialized by forming the contour of the cup was able to identify the removal of the cup from the predefined area (depicted by the green rectangular). The most interesting observation in this 2nd scenario was that the increased contour complexity (in this case complexity was added by making the cup handle visible) indeed decreases the probability of false negative regarding to the cup removal identification due to incidentally placing the cup at the exact same position.

4.3.9 Summarization

In this chapter we presented the user experience gained by using the first version of the motion detection based ADL algorithms and the main results extracted by an exploration effort regarding the algorithms’ validity as well as regarding aspects that can lead to underperformance of failure. The evaluation focused on two ADL aspects. Firstly, the algorithm focused specifically on the user trying to identify specific phases with respect to the activity of getting up from the bed and walking away from it. Secondly, the algorithm focused on the utilization of a medication cup by the user as an indication that the user has

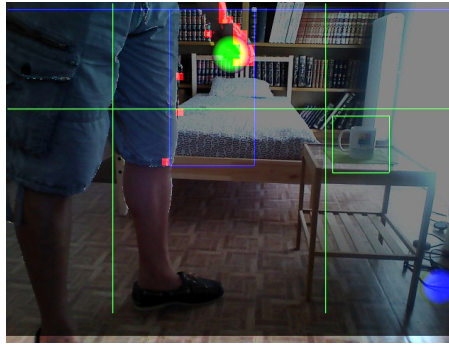


Figure 50: Cup Movement Detection Algorithm Initialization Phase, 2nd Scenario

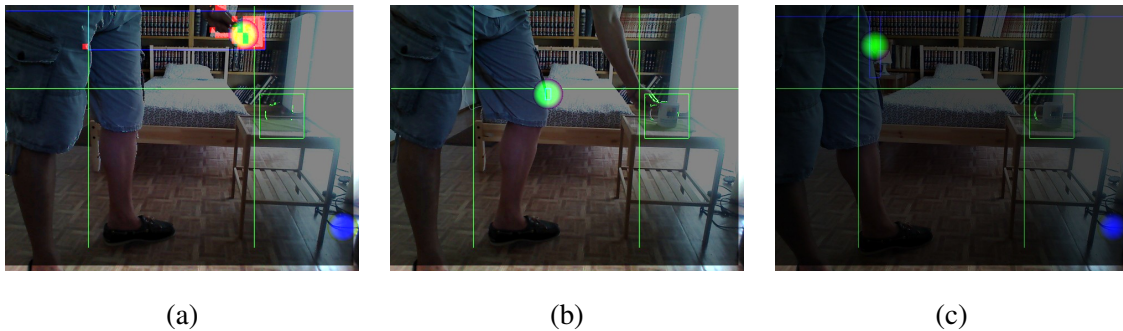


Figure 51: Cup Movement Detection Algorithm Test with respect to various final positions, 2nd Scenario

taken his/hers pills or that the user has not taken his/hers pills. Avoiding too much technical details the algorithms have been presented and their functionality analysed. Also the visualization of the different operational phases of the algorithms have also been highlighted.

It is noted that from the user point of view there is not interaction with the robot or the algorithms i.e. the user doesn't have to do anything else but go on with his/hers daily activities. This is very important since it covers the zero obtrusiveness requirement being one of the cornerstones of RADIO platform overall. Also from a technician point of view the configuration of the algorithms and adjustment to the specific requirements of the user of/and the place is quite easily done by configuring three lines (i.e. segmenting the frame in the six areas) and moving a rectangular indication (corresponding the control are of the medication cup).

Overall and after numerous experiments the algorithms performed quite well and as expected, correctly identifying all the human body's stages in the context of the aforementioned scenario as well as the event of cup displacement. A general comment concerns the fact that all event identifications are based on changes recorded inside the whole frame. That poses the requirement that in the frame nothing should change apart from the body or objective in question, in order for the algorithm to performance reliably. This could cause underperformance in case where there are more than one person or in case where apart from the body that moves something else also is moved like a chair or a table. The same issue may arise when there are sudden and significant changes in environmental conditions and especially the lighting conditions. In all such cases the algorithm identifies more points in frame that are changed than it should which effectively lead to the calculation of an erroneous gravity centre of those points which can lead to false event identifications. Further work in WP4 will look for ways to improve robustness through fusion.

REFERENCES

- Kai Oliver Arras, Slawomir Grzonka, Matthias Luber, and Wolfram Burgard. Efficient people tracking in laser range data using a multi-hypothesis leg-tracker with adaptive occlusion probabilities. In *Proc. of the 2008 IEEE Intl Conf. on Robotics and Automation, (ICRA 2008), May 19-23, Pasadena, CA, USA*, pages 1710–1715, 2008. doi: <http://dx.doi.org/10.1109/ROBOT.2008.4543447>.
- Maren Bennewitz, Wolfram Burgard, Grzegorz Cielniak, and Sebastian Thrun. Learning motion patterns of people for compliant robot motion. *I. J. Robotic Res.*, 24(1):31–48, 2005.
- Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2005)*. IEEE, 2005. doi: 10.1109/CVPR.2005.177.
- Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei et al. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD 1996)*, pages 226–231, 1996.
- Hassan Nemati and Björn Åstrand. Tracking of people in paper mill warehouse using laser range sensor. In *Proceedings of the 2014 European Modelling Symposium (EMS 2014)*. IEEE, 2014.
- Danijela Ristić-Durrant, Ge Gao, and Adrian Leu. Low-level sensor fusion-based human tracking for mobile robot. *Facta Universitatis, Series: Automatic Control and Robotics*, 1(1), 2016.
- Luciano Spinello, Rudolph Triebel, and Roland Siegwart. Multimodal people detection and tracking in crowded scenes. In *Proc. 23rd AAAI Conf. on Artificial Intelligence (AAAI 2008), Chicago, IL, 13–17 July 2008*, pages 1409–1414, 2008.
- Theodoros Varvadoukas, Ioannis Giotis, and Stasinios Konstantopoulos. Detecting human patterns in laser range data. In *Proceedings of the 20th European Conference on Artificial Intelligence (ECAI 2012)*, volume 242 of *Frontiers in Artificial Intelligence and Applications*, August 2012.