Noname manuscript No.
(will be inserted by the editor)

Action detection fusing multiple Kinects and a WIMU:
an application to in-home assistive technology for the elderly

Albert Clapés[1,2] · Àlex Pardo[1] · Oriol Pujol[1] · Sergio Escalera[1,2]

Received: date / Accepted: date

Abstract We present a vision-inertial system which combines two RGB-Depth devices together with a wearable inertial movement unit (WIMU) in order to detect activities of the daily living. From multi-view videos, we extract dense trajectories enriched with a Histogram of Normals description computed from the depth cue and bag them into multi-view codebooks. During the later classification step a multi-class Support Vector Machine (SVM) with a RBF-$\mathcal{X}^2$ kernel combines the descriptions at kernel level. In order to perform action detection from the videos, a sliding window approach is utilized. On the other hand, we extract accelerations, rotation angles, and jerk features from the inertial data collected by the wearable placed on the user's dominant wrist. During gesture spotting, a Dynamic Time Warping (DTW) is applied and the aligning costs to a set of pre-selected gesture sub-classes are thresholded to determine possible detections. The outputs of the two modules are combined in a late-fusion fashion. The system is validated in a real-case scenario with elderly from an elder home. Learning-based fusion results improve the ones from the single modalities, demonstrating the success of such multi-modal approach.

A. Clapés
E-mail: aclapes@cvc.uab.cat
À. Pardo
E-mail: alexpardo.5@gmail.com
O. Pujol
E-mail: oriol_pujol@ub.edu
S. Escalera
E-mail: sergio@maia.ub.es
· [1] Dept. Matemàtica Aplicada i Anàlisi, UB, Gran Via de les Corts Catalanes 585, 08007, Barcelona, Spain
· [2] Computer Vision Center, Campus UAB, Edifici O, 08193, Cerdanyola del Vallès, Spain

Keywords Multi-modal activity detection · Computer vision · Inertial sensors · Dense trajectories · Dynamic Time Warping · Assistive Technology

1 Introduction

In the developed countries, more and more people are to live longer; people that will inevitably be potential sufferers from age-related conditions, e.g. mild cognitive impairments or even the more severe dementia syndrome, as stated by [75]. The inversion of the population pyramids in these countries will cause the number of caregivers to be relatively small and thus hardly accessible for the elderly people. In this context, many institutions and companies are investing in advanced and cost-effective technological solutions that could complement the in-home assistance provided by caregivers and allow affected people to stay longer living independently. Such solutions can be provided by intelligent in-home assistive systems, sometimes also referred to as Ambient-assisted Living (AAL) technologies.

The AAL research is often conducted from the point of view of the Ambient Intelligence (AmI) [17,94,106,67]. The paradigm proposes the use of networks of electronic devices and sensors that seamlessly inter-operate to perceive, understand, and adapt to the user needs. Pressure sensors, RFID tags, pyroelectric (presence) sensors, video cameras or other visual sensors, and wearable inertial measurement units (WIMU or simply IMU) among others, are often found in AmI systems. These kind of systems are thus *multi-modal*: data is collected in the nodes by sensors of different nature. Data that are somehow combined to improve results over unimodal approaches, or even to enable analyses that would

not be possible given either time or computation constraints.

Researchers have exploited the video cue in order to enable perception capabilities in AmI systems and hence be able to carry out the demanded complex human analysis [43,101,81]. Nonetheless, most of the video-based action recognition algorithms have been demanding in terms of computational resources, especially those performing the highest level analyses. The apparition of Microsoft Kinect, a cheap and reliable sensor integrating both a RGB video camera and a depth infrared sensor, supposed the rapid emergence of new approaches that largely outperformed the already existing ones in many tasks. For instance, it allowed background subtraction algorithms to be robust to illumination conditions [34] or enabled the body pose estimation from depth maps in real-time and location of the joints' positions [88]. Today, depth imagery combined with the color information – namely RGB-Depth (or RGBD) – is a successful multi-modal approach being applied to a large list of human analysis-related tasks: gesture recognition [31,100,41], more complex activity recognition [72,58,90,105], re-/identification [59,73,10,68], gait analysis [23], and subject-object interaction analysis [25], etc.

Inertial sensors have been also widely considered by the authors to carry out human analysis tasks such as gesture recognition [62,2,78], gait analysis [103], or event detection, e.g., fall detection [6], especially prior to the apparition of Kinect. These devices provide a higher degree of precision over cameras determining accelerations and orientations. Despite their lack of contextual awareness, when used in combination with cameras these provide a powerful modality to exploit for the task of action/gesture recognition.

In this work, we propose a two-module system combining two Kinect devices together with a WIMU in order to recognize activities of the daily living in a real-world scenario with elderly. The Kinect devices face to each other, so to have a complete occlusion-free view of the scenario. The streams are processed to compute multi-modal dense trajectories. Our trajectories are enriched with a Histogram of Oriented Normals (HON) computed from the depth maps, complementing the relative displacement, the Histogram of Oriented Gradients (HOG), Histogram of Oriented Flows (HOF), and Motion Boundary Histogram (MBH) descriptors. Trajectories are then bagged into multi-view codebooks. Following the approach of [97], a codebook is generated for each kind of description. Then, in order to perform the classification, a multi-class Support Vector Machine (SVM) with $\mathcal{X}^2$-kernel combines the descriptions at kernel level. For action detection, a sliding window approach is followed, so that Bag-of-Words (BOW) representations for the windows are built from the extracted trajectories in an efficient "integral" way.

In parallel, an egocentric module is in charge of performing gesture recognition. In particular, the WIMU used is a Shimmer sensor placed in the elderly's dominant wrist. In order to recognize the gesture we first preprocess the data in order to extract relevant information such as accelerometer, rotation angles, and jerk. Then, we select a set of models from the sequences, which are used to obtain alignment distances (costs) by means of a Dynamic Time Warping (DTW) algorithm. During the process of detecting the gestures, a DTW performs the alignments respect to the models and determines if new gestures being performed by comparing alignment-to-model costs to a set of learnt thresholds. Gestures are defined at a different level of annotation from activities/actions, so as to be more atomic and palliate the inherent noisiness to these kind of devices.

The system is validated in a real-case dataset with elderly people from the SARQuavie Claret elder home. We guided the elders on different scripted scenarios involving gestures while interacting with objects: "taking a pill", "drinking from a glass", "eating from a plate", and "reading a book". We recorded a total of 31 different sequences of 1-3 minutes of duration each, with 14 elderly people appearing in it. The obtained results show the effectiveness of the system. Moreover, the learning-based fusion results improve the ones from the single modalities, demonstrating the success of such multi-modal approach.

Next, we summarize the main contributions of this work:

- Extension of the dense trajectories framework with the HON descriptor and its application to action detection in low frame-rate multi-view videos.
- A novel approach for considering multiple sub-models per gesture in a dynamic time warping setting.
- The integration of the two former for predicting activities in a novel real-case scenario dataset (with multiple RGBD views and inertial sensors). In contrast to many works that use in-lab recorded datasets, we recorded actual elder people with mobility problems using the settings one could find in a real home environment.

The remainder of the paper is organized as follows: Section 2 reviews the state-of-the-art on assistive technology for in-home elderly people monitoring with special emphasis on works utilizing RGBD technology and WIMUs. Section 3 describes the dataset, hardware, and acquisition settings. Section 4 presents the system. Both vision and wearable modules are explained more in depth

in their respective sections, Section 4.1 and 4.2 respectively. Section 5 introduces the results got by the different modules of the system and their integration for final detection output. Finally, Section 6 concludes the paper and discusses future work.

2 State of the art

In this section, we first cover state-of-the-art systems applied to AAL scenarios. Then, we focus on methods and algorithms that perform action and gesture recognition in the two main modules. Finally, we explain the integration of those modules' outputs.

2.1 Intelligent systems applied to AAL scenarios

AAL is aimed at preventing risks, providing palliative care, and ensuring as much as possible the independent living and well-being of older adults. A recent survey proposed a categorization of AAL tools and technologies into: ambient technologies (or *smart homes*), mobile and wearable sensors, and robotics [69]. In the category of ambient technologies, we find CASAS by [79], a system monitoring activities of the daily living to determine their completeness, and the work of [1], designed specifically to support people with dementia at their places.

2.1.1 Vision systems

There exist many vision systems for activity monitoring applied to elderly care. [70] uses location cues to determine activities. Models of spatial context are learned employing a tracker that uses a coarse ellipse shape and a particle filter to cope with cluttered scenes seen from a fisheye camera. Despite being tested in a realistic environment, the learned models are not transferable to other scenarios. Moreover, the location of the human in the scene is not enough to discern among certain activities. [30] defines 8 different activities and model the transitions from one to another by means of a Hidden Markov Model (HMM). They segment people silhouettes by means of simple but adaptive background subtraction and characterize the silhouette poses in frames with a set of three handcrafted features: height of center of mass, vertical speed of the center, and sparsity of points. In [26], the authors propose to recognize events in a knowledge-driven approach by using an event modeling framework. In knowledge-based approaches, events need to be defined by an human expert; for instance, in here, events are built taking into account a priori knowledge of the experimental scenario. The system is validated in a real-case scenario with Alzheimer patients.

The appearance of Kinect meant the emergence of systems and new techniques that could be applied to in-home AT. In [85], the system monitors human activities while seeking for signs of limb or joint pain. The work defined a set of 7 pain gestures performed by people on an average age of 40 and above. They report good results using MLP for classification on the skeletal features extracted from Kinect. However, the pain gestures are static and in a highly controlled environment. In [12], the authors present a system to monitor and control elderly people in a smart house environment. It recognizes gestures and communicates them through the network to a caregiver. They match candidate gestures to template simply by computing a distance. In this case, quantitative results are not presented. [29] proposes a system capable of recognizing full body actions, such as walk, jump, grab something from the floor, stand, sit, and lie (on the floor). The action class is determined in a heuristic rule-based fashion based on skeletal joints' positions. [16] detects pointing gestures in a smart bedroom to facilitate the elderly people interaction with the environment. In [8], the authors are detecting very simple activities such as standing, sitting, and the standing-sitting transition. In their work, silhouettes are subtracted first using a background subtraction algorithm. From these silhouettes, image moments are extracted, which are then clustered using fuzzy clustering techniques to produce fuzzy labels in the activity categories. During the functioning of the system, the classification is done by a fuzzy clustering prototype classifier. They test several imaging technologies with night vision capacities, including Kinect. The system is tested at a senior house facility with older adults with physical health issues. [104] defines a set of 13 activities of the daily living and the skeletal features are used to generate a codebook of poses. Then, in a BOW approach, poses from individual frames are binned in a histogram representing the pre-segmented activity videos and classified using a SVM classifier. The system is tested in a controlled environment and is not performing detection in continuous streams. [77] utilizes a Gaussian mixture models-based HMM for human daily activity recognition using 3D skeleton features. [38] introduces a system prototype for telerehabilitation for post-stroke patients. They monitor the range of motion of different limbs during the realization of daily living activities using a Kinect and accelerometers attached to objects. However, they only monitor the min and max and compare to the ones from the previous day and quantify the progress. Their proposal

does not tackle the action recognition task. Nonetheless, being able to recognize particular activities will make the evaluation of ranges of motions much more meaningful for therapists.

Skeleton features are widely used for action/gesture recognition in the literature of in-home assistive systems. The systems report very good results using these kind of features, but most of them are applied in very controlled scenarios in which the body is fully visible. However, the skeleton is not reliable in the presence of body occlusions or non-frontal camera angles – as it often occurs when dealing with real world situations. Fortunately, there exist many approaches to action recognition that do not require the use of such features in the literature of computer vision. Next, we review some of the state-of-the-art of action detection and recognition that could deal with some of the aforementioned problems.

2.1.2 WIMU systems

As it is shown in [57], there is a lot of effort in using IMUs to perform activity recognition. The most common approach is trying to detect ambulation-related activities (such as walking, jogging or riding bicycle). Also, the datasets used tend to be recorded in controlled conditions. One of the works analyzed in [57] is [82], which uses ontological reasoning to detect activities by means of accelerometers, physiological sensors and GPS. Most of the activities recognized are ambulation-related, but combined with some daily activities. However, the chosen ones (brushing teeth and writing on the blackboard) are pretty different. Another interesting application is the one presented in [18], in which RFID sensors are used to track objects inside a house. They expect to detect cognitive impairment in morning routines. However, the object-tracking approach is very limited to those problems in which the interaction is large in time and implies an object displacement. In [52], IMUs placed on the legs of the patient are used to track the gait and be able to perform an analysis. It is a proof of concept and the presented system is not fully developed. Finally, in [27], authors use 3D reconstructions from IMUs in order to assist the therapist in the rehabilitation process of a patient. They estimate the pose and generate a 3D animation of the exercise being performed. Additionally, a full system architecture is proposed.

2.1.3 Multi-modal RGBD-WIMU systems

We now center on works combining these two modalities: the RGBD vision from Kinect – or similar devices – with the inertial information provided by WIMUs.

[28] combine the Kinect sensor with 5 WIMUs in order to recognize activities. They fuse the multiple modalities in an early fusion fashion. They use a sliding window approach together with a set of binary trained MLP classifiers to perform the detection in one-vs-all setup. [63] performs hand gesture recognition fusing the inertial and depth data within a HMM framework, demonstrating an overall improvement. Nonetheless, the method is tested on a relatively simple dataset of gestures with 5 gestures, i.e., "wave", "hammer", "punch", "draw an X", "draw a circle", thus with considerable inter-class variability and at a relatively small distance to the Kinect camera. Our dataset was recorded in a much more uncontrolled scenario, actions are observed from a farther position, we deal with occlusions, and have much smaller inter-class variability; in fact, some classes can be only distinguished mostly by considering the interacted object, e.g., drinking and taking the pill. [42] presents some preliminary analyses for fusing Kinect and inertial sensors' data in order to monitor intake gestures. However, they do not present any performance on the task, but only some qualitative results.

In similar in-home AT scenarios but different from activity recognition, we also find systems fusing RGBD vision and WIMUs. [54] proposes a system combining the depth sensor's acquisition with accelerometer data to detect falls. Whereas the accelerometer data is used to detect potential falls, depth sensor is used to authenticate the fall alert. [14] performs joint angle estimation in a rehabilitation scenario. Authors perform on-line calibration of inertial sensor errors whenever measurements from Kinect are available. [40] introduces a real-time body tracking with one depth camera and 6 inertial sensors and improve the state-of-the-art results of tracking by combining the two. Such a tracking system would result very convenient for action/gesture recognition. Inconveniently, such amount of inertial sensors made them very invasive for in-home monitoring applications.

Next, we go in more detail on methods and algorithms, on how a vision component and a wearable component are able to perform respectively action or gesture recognition tasks by either imagery or inertial sensors.

2.2 Methods and algorithms using video, wearables, and multi-modal integration

2.2.1 Action recognition in video

Different approaches exist for action detection in video. Typically, the detection consists in localizing the action within the video either in the temporal domain [35]

or both in space and time [99,37,45]. In fact, the spatial localization of the action makes the problem even more demanding. In our case, and since we are not intended to do that, we focus on the temporal localization. These methods often use a variable-size sliding window in which actual action classification is performed. There exist more cumbersome approaches, like the one of [35], that break down actions into sub-actions (or *actoms*) and model explicitly the length of the window for each class. Unfortunately, it requires expensive manual annotations of *actoms*. In this work, we simply convolve a sliding window of different sizes and perform action classification in it. Next, we review the state-of-the-art methods for action classification.

Following the taxonomy proposed by [35], we divide action recognition methods in videos – it can also be applied to gesture recognition – into three main categories: sequential modeling, template-based methods, and local features. *Sequential modeling methods* can be categorized into (a) those that learn temporal transitions among a set of hidden states, i.e. the Hidden Markov Models-based approaches [91,24], and (b) those that use the alignment score between an action and class exemplars [19]. The main limitation they present is dealing with concurrent actions or gestures. Moreover, Hidden-Markov Models are data-demanding; being highly parametric they suffer with insufficient training data [74].

On the other hand, *template-based methods* consider the video as a 3D volume – where the temporal (third) dimension is the depth of the volume. Matches among volumes can be done directly by means of tensor-based techniques directly [51], or representing first the actor using silhouettes [13,83], optical flow information within the volume [86,49], or space-time energy measurements [15,84]. These methods perform poorly when videos are not in controlled environments where the full body is visible.

Finally, in *local feature-based methods* a set of local patches are extracted on 3D interest points detected in the spatio-temporal volume and later described [87]. The detection is crucial [55]. These interest point detectors are typically combined with the most powerful state-of-the-art descriptors, e.g. STIP+HOG [56]. Lately, the dense trajectories have replaced the use of STIP [97,98]. These are constructed from optical flow fields, by tracking the displacement of a pixel during a subset of frames. Like in STIP approaches, appearance and motion descriptors are computed in the image patches all along the trajectory. Either the STIPs or trajectories provide a representation based on local features, thus a way of computing a global vector-form representation for a video (or temporal segment) is required. This is typically tackled using a Bag-of-Features approach [97] or the more recent Fisher Vector (FV) representation [98] in combination with a discriminative state-of-the-art classifier.

Taking into account the success of Convolutional Neural Networks in image classification, it is worth considering them also for action classification [48]. Unfortunately, they still have not demonstrated a major improvement over other more classical state-of-the-art techniques for action recognition [7]. Yet they can be used in practice in combination with other methods [102].

The methods discussed above are also applicable when having available the depth modality. Many works exist using temporal modeling methods on the skeleton data from Kinect [80,11]. These high-level features extracted from depth maps demonstrated their reliability, even outperforming low/mid-level features when the full body is visible [46][1]. Moreover, the skeleton representation is low-dimensional data, which makes potentially easier the learning of transitions in HMM and also reduces the cost of aligning action candidates to class exemplars in the DTW approach. Regarding template-based methods, they are also applicable in the depth cue [61,60]. In [61], the authors extend the energy-based method from [15] to depth data, extracting the motion-energy features in the three cartesian planes got from depth maps separately to later build a spatio-temporal pyramid cuboid representation of the action videos.

Regarding local features, [41] propose a Bag-of-Visual-and-Depth-Words framework for gesture recognition. The authors use the STIP detector separately in RGB and depth modalities, and then describe the color interest points using HOG and HOF, and the depth interest points with VFH+CRH[2]. Then, in order to compute the global representation of gestures they use spatio-temporal pyramids and a Bag-of-Words approach. Finally, the global gesture representation is matched to the training samples using lazy learning. Despite locally extracted depth features have been used for action/gesture recognition, there is not (to our extent) any work using dense trajectories in RGBD videos. In this work, we propose to enrich the description of the trajectories consisting of the track's relative position, HOG, HOF, and Motion Boundary Histogram (MBH) with a Histogram of Oriented Normals (HON).

[1] This is not stated in the published manuscript, but in an errata document. Check [46] and the errata document for more detail: http://jhmdb.is.tue.mpg.de/show_file?filename=Errata_JHMDB_ICCV_2013.pdf

[2] The concatenation of the Viewpoint Feature Histogram (VFH) and Camera Roll Histogram (CRH).

2.2.2 Gesture recognition using WIMUs

In order to solve the gesture detection and recognition problem, different approaches are used in the literature. For instance, [53] introduce a gesture segmentation (start, middle and end) and train a SVM classifier to predict both gesture phase and class. The presented results are pretty good but they do not deal with non-gesture samples, so at the end they are not detecting gestures but recognizing them. In [33], trajectories from IMUs are computed by taking the second integral of accelerometer data, obtaining displacement. In the presented scenario (really short sequences) the approach performs well, but in a real-case dataset with large sequences the accelerometer drift will make impossible any classification. In [36], a very large dataset is recorded (25 sessions with 4500 gesture repetitions). They compare two approaches (accelerometer and EMG) and an early-fusion of them. In order to model each of the gestures they use a Hidden Markov Model (HMM) that they are able to train due to the amount of data they have. Also using HMM, [4] detect nutrition-related gestures from a dataset with two subjects, giving good results. In [47], the same HMM is used, but a preselection stage is defined, in which interesting regions are selected for further classification. This approach implies a limitation in the number of classification errors since the first classifier is reducing the space. [5] presents a survey on Activity Recognition using Inertial Movement Units; there, authors detail some of the methods that have been used in the literature, such as Decision Trees [9,**?**], Nearest Neighbors [64] and Artificial Neural Networks [44,**?**]. Finally, in the case of study presented in [20], a large number of variables that affect gesture recognition with IMUs (e.g. sensor position, inter/intra subject variability, types of features) are studied and quantitative results are presented.

2.2.3 Multi-modal fusion techniques

Focusing on the fusion part, there are two different strategies: combine the values for each modality at the beginning of the pipeline (feature-level), named early fusion or after computing the prediction values for each of the inputs (decision-level), late fusion.

State of the art late fusion strategies use the scores given as outputs from early stages. For example, in [39], authors use product, sum and weight as fusion strategies for probabilities. The work presented in [71] use a bayesian model based on the scores given by the classifiers. Also other methods such as using a SVM trained with the concatenation of the outputs is used ([89]) or more complicated strategies as it is shown in [107], where a top-down approach is used, in a first stage, a coarse label is generated and then, it is fed to the fusion module which gives a fine-grained category. Finally, in [22] two different strategies are compared, a uniform average and a weighted average, the latter giving better results.

3 Data, hardware, and settings

This section describes the dataset, the hardware used to record the data, and the system's physical settings and software infrastructure.

3.1 Data

The SARQuavitae Claret dataset consists of a total of 31 sequences of 1-3 minutes of duration each, with 14 elderly people performing different scripted scenarios that involve the realisation of activities of the daily living: "taking pill", "drinking", "eating", and "reading". These activities emerge from the interaction with four different objects: a plastic eating plate that may appear with a plastic fork, a plastic cup, a photography book, and a small two-lid pillbox. Yet other irrelevant objects appear. For instance, a juice tetra-brick or the objects the elderly bring to the scene, e.g., purses, wallets, or a walking stick. Table 1 summarizes the most important aspects of the dataset.

We defined and manually annotated two levels of annotations: activities and gestures. Whereas the vision part directly performs recognition on activities, the wearable module requires more atomic annotations - we named *gestures*. The number of gesture classes coincide with the number of activities, those being "spoonful", "drink", "turn page", and "take-pill". These more atomic annotations reduce intra-class variability of the original activities, making the task of the wearable model easier. Fig. 1 introduces both the visual data (frames and objects) and inertial data (accelerometer's gestures).

For the manual annotation task, we first synchronized the streams of the two cameras along with the one from the sensor. Then, we annotated both activities and gestures at frame-level. In the case of the activities, beginning and end coincide with the interaction with each particular object – an interaction being considered the intentional manipulation of an object. For instance, for "eating" the activity starts when the subject reaches the dish and/or fork, continues during a variable number of spoonfuls, and finishes when the subject drops the fork after the last spoonful. On the other hand, we defined the beginning and the end of gestures in different ways depending on the class. For

	Modules	
	Vision	Wearable
Task	Action detection (4 actions)	Gesture spotting & recognition (4 gestures)
Hardware	2x RGBD sensors (Kinect)	1x WIMU (Shimmer)
Type of data	RGB Depth	Accelerometer Gyroscope Magnetometer
No. sequences	31	
No. subjects	14	
No. frames	3,747 + 3,701	36,858
No. actions (gestures)	86 (162)	
General challenges	Elderly subjects, uncontrolled behavior	
Specific challenges	Ambient light reflections and shadows Small objects Low framerate Depth noise	Gesture intra-inter variability Device noise

Table 1: Summary of the SARQuavitae Claret dataset

"take-pill", "drink", or "spoonful", the gesture begins when the hand – already touching either the pillbox, cup, or fork – starts accelerating towards the mouth of the subject and ends in the very same instant when the hand starts accelerating again moving away from the mouth. In the case of "turn page", the gesture begins when the page starts to be turned and ends when it has been completely turned.

The dataset presents several challenges that have to be addressed:

Objects' viewpoint variability and size. The objects can present heavy changes in terms of appearance in the color cue, due to either partial occlusions or the viewpoint. The viewpoint might in fact cause also huge variations in the objects' shape observed from depth maps. Note the variability of the book: opened in Fig. 1(b) versus closed in Fig. 1(d). What is more, the relatively small size of some of the objects causes them to be completely shapeless when observed at 2 meters distance because of the inherent noise introduced by the Kinect depth sensor.

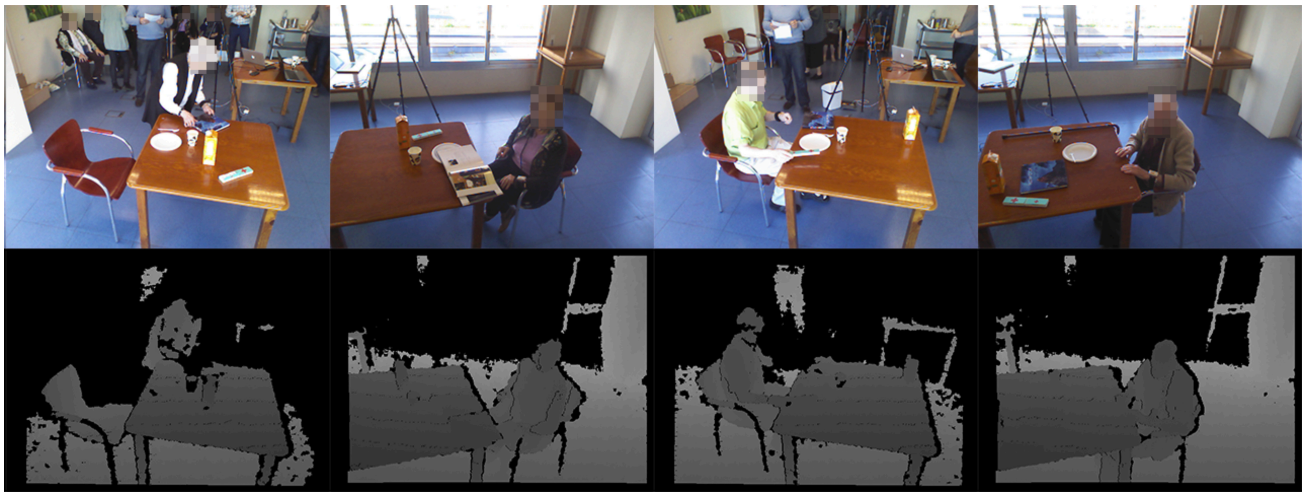
Uncontrolled behavior and introduction of external objects. Despite the scenarios were scripted to ensure a balance of activities' examples in the dataset, the participants were not always following the given instructions, thus introducing a certain degree of improvisation in the scenarios; such as, for example, the entrance of external objects – like the walking stick in Fig. 1(d).

Shadows and specular surfaces. Because of the light coming from the window, one of the views presents shadows and the other mostly reflections on the shiny table surface. Fig. 1(a-c) and Fig. 1(b-c) illustrate this phenomenon. This situation requires the implementation of techniques general enough to adapt to both situations. However, reflections are much more difficult to deal with than shadows from the point of view of the color cue: reflections can cause not only brightness variations, but changes of color hue. Moreover, the noisiness of the depth measurements is worsened on shinny surfaces. In the view corresponding to Fig. 1(b-d), where the table surface reflections ambient infrared light coming from the window, we observed the depth readings to be noisier.

Inter- and intra-variability of activity/gesture examples. Some of the activities/gestures we are intended to recognize are quite similar one to another: the arm movement is very similar in "drinking" and "taking pill" from the perspective of the vision module. In addition, in the inertial cue, there is also a certain degree of similarity between gestures of different classes compared to the "no-gesture" – that is, when the participants are almost steady. On the other hand, the dataset presents a significant variability within each category regardless of the data cue utilized. This becomes particularly evident when observing the recorded instances of "reading a book"/"turn page". In addition, dealing with the inertial data becomes even harder when the sensor is worn by elderly people with shaky hands.

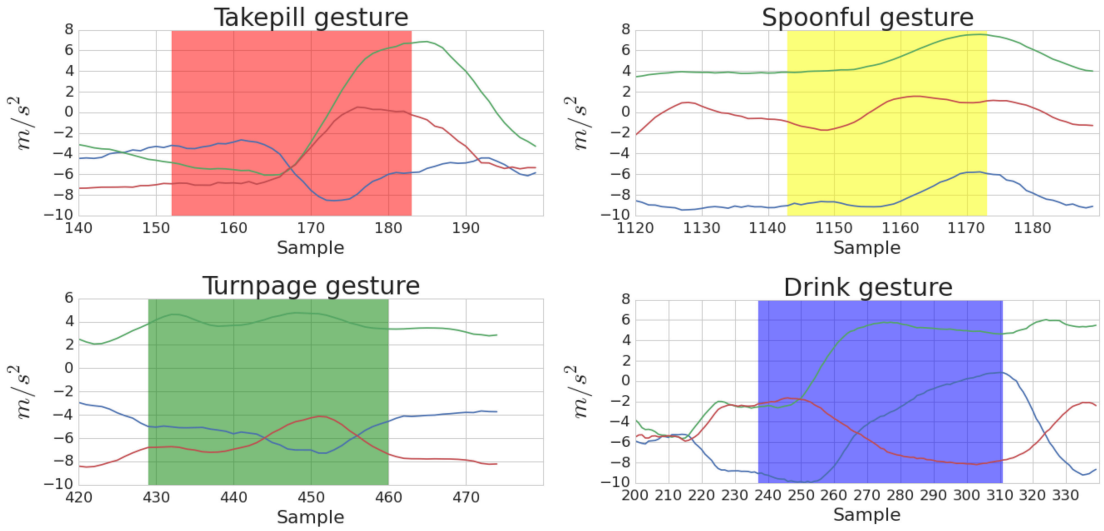
3.2 Hardware

We used two RGBD cameras, each connected to a different laptop computer. Among the existing RGBD cameras, we chose the popular Kinect device for its price and reliability. The device uses a structured IR light



(a) (b) (c) (d)

(e)



(f)

Fig. 1: In (a)-(d), the visual data acquired, with color frames in the top row and depth maps in the bottom row: (a)-(c) correspond to view #1 and (b)-(d) to view #2; in (e), the objects we are intended to recognize; and, in (f), examples of the four gestures' accelerometer readings for the three axis, x (red), y (green), and z (blue)

pattern which is projected to the scene by the emitter and read back by the IR sensor.

Among existing WIMU sensors, we chose Shimmer, which consists of accelerometer, gyroscope, and magnetometer. An IMU is a high-frequency sampling sensor; and particularly Shimmer is able to sample in a wide range of frequencies, up to 200Hz. In contrast to other IMU providing inertial data in millivolts (mV), the Shimmer device converts the mV to standard units, that are m/s^2 for accelerometer, rad/s for the gyro-

scope, and *Gauss* for the magnetometer. The device can also stream the signals to other devices via Bluetooth.

3.3 Acquisition settings

The Kinect devices were both elevated at 2 meters height using tripods and pointing to the table in which the activities took place. In both views, the closest table point was at 1 meter and the furthest at 1.8 meters. The devices recorded the scene from complementary viewpoints with intersecting frustums, so as to obtain the most complete picture of the scenario. However, using such setup, IR patterns interfere with each other causing the devices to provide unreliable depth measures. The solution adopted was to use two Kinect for Windows edition, which offered the possibility to turn on and off the IR-emitter. Since the turn-on/turn-off have to be synchronous, we implemented a ping-pong recording setup alternating in time the devices' acquisition. While this solved the problem, it also reduced the sampling rate to 4 FPS (2 frames per second and per view); and the exact time was determined empirically from an experiment explained in more detail in Section 5.1.1.

The Shimmer sensor was attached with a velcro strap to the right wrist of each of the participants. We set the sampling rate to 25Hz with the aim to minimize delays on the communication due to data processing bottlenecks. Since the sensor communicates over Bluetooth, and this technology has a short range, we used an Android phone as a bridge for transmitting data from this Personal Area Network (PAN) to the Local Area Network (LAN) by emitting over WiFi to a laptop computer. The phone is also responsible of time-labeling the samples using a timestamp. This configuration increases the freedom of the user in the environment and provides a more realistic setting. Since the magnetometer measurements depend on the sensor's orientation w.r.t. the magnetic north, we discarded these features so as to make the gesture recognition magnetic-orientation invariant.

4 System

The proposed system consists of two main components, the vision component and the wearable one. After performing separately, a third component integrates their outputs in a late fusion fashion, as shown in Fig. 2.

In the *vision module*, multi-modal dense trajectories (MmDT) are extracted from the multiple RGBD streams acquired from the Kinect devices. We refer to the dense trajectories as "multi-modal" since a HON

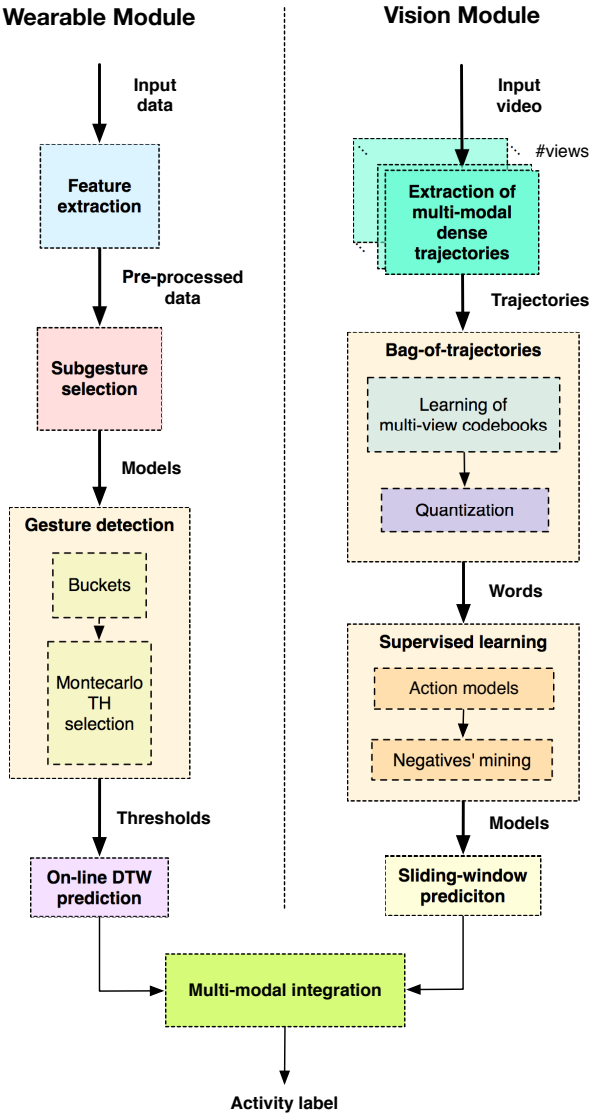


Fig. 2: General pipeline of the system consisting of two modules: a vision module and a wearable module

descriptor computed from the depth modality in addition to the ones from the color cue. The additional depth descriptor adds extra shape information to the appearance or motion information throughout the trajectory.

Following a Bag-of-Words approach, we generate next a set of codebooks for the different kinds of MmDT features: relative displacement features of the trajectories ("Trj" from now on), HOG, HOF, MBH, and HON features; making a total of 5 codebooks. Codebooks are multi-view, i.e., they are trained using MmDT from different views.

For the detection, we slide a temporal window over the videos. A word representation is built for each win-

dow and descriptor. We then determine its category using a SVM classifier. The classifier is trained with examples of each of the activities (positives) altogether with negative examples.

The *wearable module* preprocesses the acceleration and angular velocity data as a first step. Since raw inertial data is inherently noisy, we normalize and filter out outliers. Then, we extract a set of features that are: raw data from accelerometer, sorted data from accelerometer, jerk, and complementary filter.

Secondly, the module clusters the data in order to find a set of representative gesture models. For each model, we learn a distribution of alignment costs (from a separate data sample), in such a way during the prediction phase we can simply threshold the alignment cost of any potentially observed gesture instance and hence determine if it is one of the gesture classes being performed. For the computation of thresholds, we use a random-selection Montecarlo method.

Finally, the *fusion module* takes the binary outputs of the vision and wearable components and fuses them. This is done by applying the intersection. This is a fast method which improves the performance over the single modalities.

4.1 Vision module

The vision module implements a pipeline with three main stages: the feature extraction, the construction of a mid-level representation, and the sliding window-based action detection itself. Next, we explain in more detail each of the stages of the vision pipeline (illustrated in Fig. 2).

4.1.1 Multi-modal dense trajectories (MmDT)

The features extracted are based on the dense trajectories from [97]. As in it, trajectories are sampled from dense optical flow fields[3] by tracking the displacement of pixels during L frames. Moreover, state-of-the-art appearance and motion descriptors are computed in $N \times N$ sized image patches along each trajectory which complement the relative displacement information that inherently characterizes the trajectory. More precisely, the trajectory is divided into $n_x \times n_y \times n_t$ sub-volumes in which the descriptors are computed and averaged. The descriptors are finally concatenated to build the actual description of the trajectory. This is repeated at different spatial scales with fixed sampling stride s.

While trajectories are computed solely from the color cue, we compute the surface normals on the observed depth maps and summarize this information as a histogram of oriented normals (HON) that we attach to the original descriptors (Trj, HOG, HOF, and MBH). Fig. 4 illustrates the construction of the multi-modal dense trajectories. HON representation enriches the representation of the multi-modal dense trajectories as demonstrated later in our experiments. Next, we briefly explain how we compute the HON representation of a depth map.

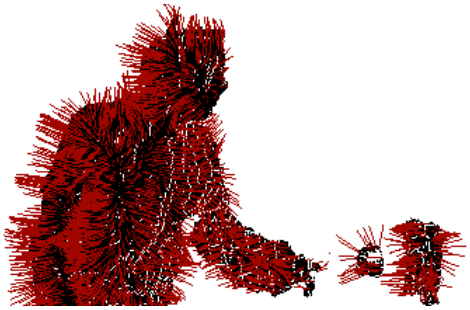


Fig. 3: Surface normals computed from a depth map in which a person tries to reach some objects. Black dots are 3D points and red lines are vectors representing surface normals (arrow heads are not drawed for the sake of the visualization)

Histogram of oriented (depth) normals (HON) Based the work of [92], we compute a histogram counting the orientations of normal vectors computed from depth map. In order to do so, we first transform the map to a point cloud $\mathcal{P}$ in which we have 3D points in "real-world" coordinates (values representing actual distances in $\mathbb{R}^3$). Then, finding the surface normal 3D vector at a given point $\vec{p} = (p_x, p_y, p_z) \in \mathcal{P}$ can be seen as a problem of determining the perpendicular vector to a 3D plane tangent to the surface at $\mathbf{p}$. Let denote this plane by the origin point $\mathbf{q}$ and its normal vector $\mathbf{u} = (u_x, u_y, u_z) \in \mathbb{R}^3$. From the neighboring points $\mathcal{K}$ of $\mathbf{p} \in \mathcal{P}$, we first set $\mathbf{q}$ to be the average of those points:

$$\mathbf{q} \equiv \bar{\mathbf{p}} = \frac{1}{|\mathcal{K}|} \sum_{\mathbf{p} \in \mathcal{K}} \mathbf{p}. \tag{1}$$

The solution of $\mathbf{u}$ can be then approximated as the smallest eigenvector of the covariance matrix $\mathbf{A} \in \mathbb{R}^{3\times3}$ of the points in $\mathcal{P}^{\mathcal{K}}_{\mathbf{p}}$. The sign of $\mathbf{u}$ can be either positive or negative, so we adopt the convention of consistently re-orienting all normal vectors towards the depth sensor $\mathbf{z}$ viewpoint. In Fig. 3, we illustrate the normals extracted.

[3] Dense optical flow is computed using [32].

The normal vectors already computed are represented in cartesian coordinates using 3 parameters. However, when expressing them in spherical coordinates (radius r, inclination θ, and azimuth φ), one of the parameters (r) turns out to be constant in our case. This more compact representation is calculated as follows: $\theta := \arctan(u_z/u_y)$ and $\varphi := \arccos\sqrt{(u_y^2+u_z^2)}/u_x$. Hence, the final HON representation consists of a two-dimensional $\delta_\theta \times \delta_\varphi$ histogram, with each bin counting occurrences of pairs of (θ,φ). This structure is vectorized an added as the fifth feature of our MmDT framework.

4.1.2 Mid-level Bag-of-Words (BOW) representation

Since MmDT are locally extracted along videos, we need to compute a mid-level representation for each temporal segment we are intended to classify later during the detection phase. As in [97], we use a BOW-like approach. For this purpose, we generate multi-view codebooks of k_{vis} centers using K-Means (with the euclidean distance metric) from a sample of M examples each, one for each of the five trajectory features: $\mathcal{C} = \{\vec{C}_d\},\ \forall d \in \{\text{traj}, \text{HOG}, \text{HOF}, \text{MBH}, \text{HON}\}$. From them, we generate the mid-level representations or words. The words simply count the frequency of each of the k_{vis} codes in a particular temporal video segment.

4.1.3 Action detection

In order to perform detections in a video sequence, we follow a sliding window and detection-by-classification approach. We choose BOW to be particularly convenient in the sliding window scenario. Having the BOW representation computed at frame-level, it is possible to compute the representation of a window centered at certain frame in an "integral" efficient way.

Let us denote $\mathbf{B} \in \mathbb{N}^{k_{\text{vis}} \times F}$ a matrix-like structure representing the set of BOW descriptors for a sequence of F frames as columns and $\mathbf{V} \in \mathbb{N}^{k_{\text{vis}} \times F}$ the column-wise accumulation of $\mathbf{B}$. Then, the representation BOW representation of a window $\vec{w}_t$ centered at frame t can be computed as follows:

$$\vec{w}_t := \vec{v}_{t+\lfloor \frac{w}{2} \rfloor} - \vec{v}_{t-\lfloor \frac{w}{2} \rfloor - 1}, \tag{2}$$

where w is the width of the window and $\vec{v}_t$ the accumulation of $\mathbf{B}_{1:t}$. Given a window, we obtain $|\mathcal{C}|$ different words – as many as description codebooks.

In order to classify a window, we use a non-linear SVM with a RBF-$\mathcal{X}^2$ kernel. As in [97], the different channels are combined at kernel level [93]:

$$K(\vec{w}_i, \vec{w}_j) := \exp\left(-\sum_{c\in\mathcal{C}} \varpi^{(c)} \frac{1}{A^{(c)}} D(\vec{\hat{w}}_i^{(c)}, \vec{\hat{w}}_j^{(c)})\right), \tag{3}$$

where $D(\vec{\hat{w}}_i, \vec{\hat{w}}_j)$ is the chi-square distance between the pair of normalized words $(\vec{\hat{w}}_i, \vec{\hat{w}}_j)$, $A^{(c)}$ is the mean chi-square distance among training words in channel c, and $\varpi^{(c)}$ is a weight assigned to c. The weights sum up to 1. Words are l1-normalized as suggested for the computation of non-linear kernel maps [95].

Finally, we generate the detection output. Since we slide temporal windows of different sizes, at a certain position t windows from different sizes can give different responses. Let us define $\vec{Y}^{(i)} \in \mathbb{N}^{|S|\times T^{(i)}}$ as the response matrix for the i-th video, where S is the set of activity categories and $T^{(i)}$ the duration of the video. Then, given a category c and temporal instant t, $\vec{Y}^{(i)}$ is assigned as follows:

$$\vec{Y}^{(i)}_{c,\, t-\lfloor \frac{w}{2} \rfloor : t+\lfloor \frac{w}{2} \rfloor} := \mathbb{1}\left\{ g\left(\vec{x}^{(i)}_{t-\lfloor \frac{w}{2} \rfloor : t+\lfloor \frac{w}{2} \rfloor}\right) = c \right\}, \tag{4}$$

where c is a particular category, $\mathbb{1}\{\cdot\}$ is the indicator function, and $g(\cdot)$ is the functional representation of the multi-class classifier. If the classifier provides a response of c, the sliding window's temporal extent centered at t in the c-th row of $\vec{Y}^{(i)}$ is marked as positive, i.e., 1.

4.2 Wearable module

The wearable module takes accelerometer and gyroscope data recorded using the WIMU and learns the models with the goal of detecting the gestures. First of all, features are extracted from the raw data, namely, raw acceleration, "sorte" acceleration, complementary filter, and "jerk". The large variability in the execution of gestures motivates to look for different patterns under the same named class. In this sense, we apply a clustering strategy over segmenting gestures in order to obtain the most representative models for each class. The detection of gestures in a sequence requires the elastic comparison of each possible sub-sequence with all the model gesture patterns. For this task, we use a dynamic time warping technique. The final detection of a gesture is obtained when the accumulated aligned similarity between a sub-sequence and the tested pattern is below an acceptance threshold. The selection of the threshold is a critical step for obtaining accurate detection results. In the training step one has to find an acceptance threshold for each candidate gesture. However, if we use multiple models for each gesture the process becomes considerably more complex.

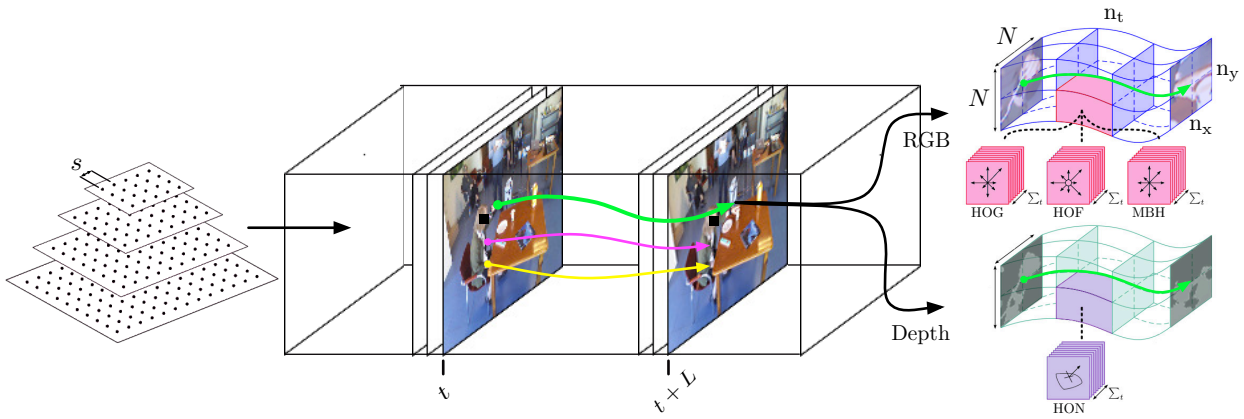


Fig. 4: The multi-modal trajectories are extracted at different spatial scales. In each spatial scale, dense optical flow fields extracted from the color cue are used to track the pixels during L frames at most. The trajectories are represented then by the relative displacements along its duration together with a set of descriptors (HOG, HOF, MBH, and HON) computed in $n_x \times n_y \times n_t$ sub-volumes from the descriptors' corresponding cues

For this reason a Montecarlo optimization technique is used for establishing the acceptance threshold for each sub-pattern in each gesture.

Next, we describe the details of this module, corresponding to the left part of Fig. 2.

4.2.1 Feature extraction

Prior to the feature extraction, we smooth the sequences using a mean filter with kernel size of 10 samples. Errors in the measurements in the form of outliers that largely deviate from the mean are removed by applying a thresholding operation on values above or below three standard deviations.

With the input signal properly preprocessed, we compute a set of features. These features are the following:

Raw accelerometer data Data recorded on the scenario regarding only to accelerometer. Used in some works as in [66].

Sorted accelerometer A set of discrete features which account for a relative rank among the three axis of the accelerometer is defined. For each sample we assign a value (-1, 0 or 1) according to the ranking of its value compared to the other axis, i.e. the axis with the lowest value is set to -1, the axis with the largest value is set to 1, and the remaining one to 0.

Complementary filter The complementary filter mixes gyroscope and accelerometer values in order to get a smoother signal with less noise and transforming acceleration into rotations. In essence, we transform the acceleration vector $\vec{a} = (a_x, a_y, a_z) \in \mathbb{R}^3$ of each sample into the rotation vector, then we apply a low-pass filter to the accelerometer in order to remove noise, and a high-pass filter to the gyroscope for removing the drift (an almost constant component). Then we merge both measures in order to get the orientation of the sensor. The rotation vector from the accelerometer is computed as follows: $\alpha := cos^{-1}(a_x/||\vec{a}||_2)$, $\beta := cos^{-1}(a_y/||\vec{a}||_2)$, and $\gamma := cos^{-1}(a_z/||\vec{a}||_2)$. Then, for each sample i we are able to apply the complementary filter defined on the next equation:

$$C_x^{(i)} := \sigma\psi_x^{(i)} + (1-\sigma)\alpha^{(i)}, \quad (5)$$

$$C_y^{(i)} := \sigma\psi_y^{(i)} + (1-\sigma)\beta^{(i)}, \quad (6)$$

$$C_z^{(i)} := \sigma\psi_z^{(i)} + (1-\sigma)\gamma^{(i)}, \quad (7)$$

where (ψ_x, ψ_y, ψ_z) represent the gyroscope values, while the value of σ controls the response of the filter. As demonstrated in [50], the complementary filter reduces drift and noise presented by accelerometer and gyroscope while maintaining the computational complexity (compared to Kalman Filter [65]).

Jerk We use Jerk, which is the derivative of the acceleration $\vec{j}(t) = \frac{\mathrm{d}\vec{a}(t)}{\mathrm{d}t}$. It shows the transitions of the acceleration and is numerically computed using centered differences [21].

4.2.2 Sequence similarity using Dynamic Time Warping

The DTW algorithm aligns two time-series of different length and returns an alignment cost [96]. This particular property makes possible to compare sequences with different duration without losing information. This procedure is used in two different steps in our system. First, it will be used as a similarity metric in the definition of the sub-patterns in each gesture category. And second, it will be used when checking the sequence against each sub-pattern, both in the training step for the threshold selection and then in the test step for the detection.

4.2.3 Sub-gesture selection

As previously commented, we observe a large variability in the way a gesture is performed. This motivates the search of a set of models under each gesture category. In order to perform this task we use K-Medoids algorithm using DTW as a distance function. The training sequences are segmented in order to obtain the individual gestures. Then, we compute the DTW alignment costs for all of them. Because DTW is not a proper metric (it is not symmetric) we define a pseudo-metric by adding to the DTW matrix its transpose. The result of the K-Medoids is a set of k_{wear} training examples. These examples will be considered as different model prototypes, M_i, for the gesture.

4.2.4 Gesture detection

The dynamic programming matrix from DTW enables the reconstruction of the matching path. This provides the temporal extent of the matched pattern within a sequence of observations. Because we want to detect full patterns inside the sequence, we need to set the first column of the dynamic programming matrix to infinity. In this way we ensure the best alignment to always start at the first row. Also, we want the algorithm to detect the pattern as a sub-sequence inside our full sequence. For doing so, we set the first row to 0, then we allow a gesture to begin at any position.

A cell $\{i, j\}$ of the DTW matrix is computed by taking the minimum of the three upper-left neighbours $min(\{i-1, j\}, \{i, j-1\}, \{i-1, j-1\})$ and adding the euclidean distance between the two corresponding frames $\|P_i - S_j\|_2$. Since the last row of DTW matrix represents the alignment cost of a certain pattern against a sub-sequence, we should expect that a gesture will have lowest value than other parts of the sequence. Then, we use a threshold per model for detecting the gestures. Next, we explain Montecarlo optimization for the selection of those thresholds.

Montecarlo threshold optimization As commented, the main difficulty for learning the acceptance thresholds is the multiplicity of models for each gesture. The problem lies in the fact that the ground-truth data only defines the gesture category. However, because we have several models per gesture we do not know which model best represents that gesture in a given sequence. The naive strategy for solving this problem would be to select a single acceptance threshold for all models in a gesture. However this severely hinders the expression power of each of the models. We opt for learning a different threshold per gesture model. Thus the learning problem needs to find the best acceptance threshold for the best model among the models for each gesture. In order to solve this problem we use Montecarlo optimization.

We assume that given a sequence, there will be a unique model for each gesture, since that sequence is performed by a single user. Then, for each gesture, we compute all the DTW matrices associated to all the models of that gesture. We will have as many matrices as number of models for each sequence.

Montecarlo optimization is based on randomly generating solutions of the problem at hand and choosing the solution that optimizes the objective function. This kind of optimization technique is specially suitable when the objective function is easily computed and the solutions are complex, for example, structured solutions. The convergence speed of this method is very slow, of the order of the square root of the number of samples generated.

In our problem the objective function is the F1 score. Given a set of m sequences and n models per gesture, a solution is composed by m pairs model-threshold $\{M_i, \tau_i\}$ out of the n models, one for each of the sequences. If we order all the sequences, this is easily illustrated as a graph in which, for each sequence we have a node for every model. Then, a solution is the combination of the path that goes through all sequences combined with the corresponding thresholds selection for each model. An example of the graph is shown in Fig. 5. Observe that each path defines the correspondence of a single model for each sequence. Thus, a path may involve the same model applied on different sequences. For example, in Fig. 5, the blue path considers M_1 in sequence 1 and 2. Given a path, the threshold selected for each model is the one that maximizes the average F1 score over all sequences that consider that model. Each path is then scored according to the average F1 score achieved applying the selected thresholds. The final solution corresponds to the path that achieves the highest average F1 score.

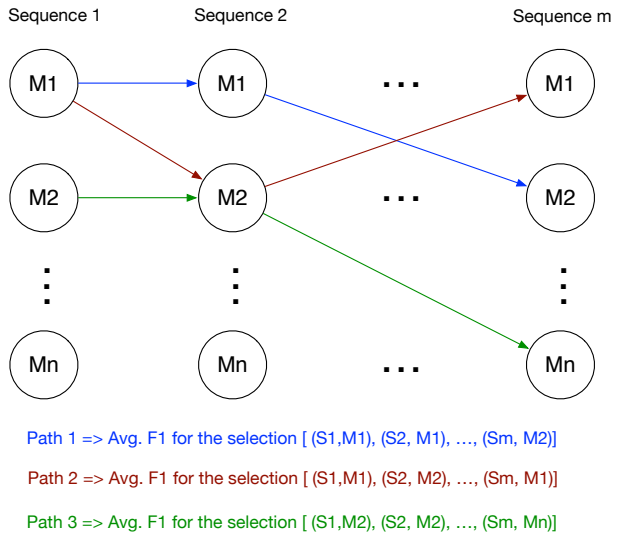


Fig. 5: The montecarlo threshold-selection method

As a practical note, in order to define the range of thresholds we evaluate for each model, we compute the DTW matrices corresponding to each model over each training sequence. Using the annotations of the ground-truth, we can get the alignment costs of each of the gestures. We take not only those values but also, the ones corresponding to the frames close to the end point defined by the groundtruth. All these values will be used as positive samples. By doing so, we allow a certain distortion on the gestures. The parameter that controls the number of points we take is called tolerance and is a percentage over the length of the gesture. We also consider all values corresponding to negative points. The range of thresholds to be evaluated goes from the minimum value of the positive samples up to the first quartile value of the negative ones.

4.3 Integration module: learning-based fusion

Given a particular time instant, the vision and wearable modules provide a detection decision for each of the activity/gesture classes. We designed a learning-based fusion strategy consisting on stacking a centered window of size around each predictions of size ωN on a $2\omega N$-valued feature vector representation that can be input to a discriminative classifier:

$$\mathbf{x}^{\mathrm{v+w}} = [\mathbf{y}_1^{\mathrm{v}}, \mathbf{y}_2^{\mathrm{v}}, \ldots, \mathbf{y}_{\omega N}^{\mathrm{v}}, \mathbf{y}_1^{\mathrm{w}}, \mathbf{y}_2^{\mathrm{w}}, \ldots, \mathbf{y}_{\omega N}^{\mathrm{w}}] \,,$$

where $[\cdot]$ is the concatenation operation and $\mathbf{y}^{\mathrm{v}} \in \{0, 1\}$ and $\mathbf{y}^{\mathrm{w}} \in [0, 1]$ are prediction values, respectively, from the vision (v) and wearable (w) module. Note the prediction values from the wearable module are binary, whereas the ones from the vision one are real-valued confidences. The latter are calculated as the ratio of positive binary predictions for different sliding window sizes divided by the total number of window sizes.

In order to perform the classification, we train a neural network per activity class, consisting of a $2\omega N$-neuron input layer, two fully connected layers and 2-neuron output. The net is trained using *adam* optimizer and *cross entropy loss* function. For the output layer, we use a soft-max function.

During the training of each epoch, we feed the net with 80% positive examples and 20% of negatives. Hence, the loss function is weighted in order to compensate the bias introduced by this difference.

5 Results

First, we illustrate the experiments carried out to establish some settings in the different components of the system and detail which are the system's parameters. We also explain different strategies to fuse the outputs of the visual and inertial modules. Finally, we illustrate the results of the two main modules separately and eventually the visual-inertial fusion results.

5.1 Settings and parameters

In this section, we first present a preliminary experiment for the IR-emitter's delay calculation and then we introduce the parametrization of each system's module.

5.1.1 IR-emitter turn-on time delay

The blocking turn-on instruction of Kinect for Windows does not ensure the IR light bulb having reached full power before the acquisition of a new depth frame – thing that causes erroneous depth map readings. In order to determine the "real" re-activation time, we performed the following experiment: we fixed a sensor in an indoor still scene and captured 2,500 depth maps, selecting one out of those as a reference frame. Next, we subtracted the reference frame to each of the rest and calculated for each of those frame differences the accumulation of absolute differences, i.e. the magnitude of the difference of that frame respect to the reference. Fig. 6 depicts these magnitudes in function of the turn-on time the blocking instruction took, as well as the effect of forcing a turn-on delay or none (0 ms) instead of relying on the instruction's own optimistic delay. Given the results, and seeking to ensure

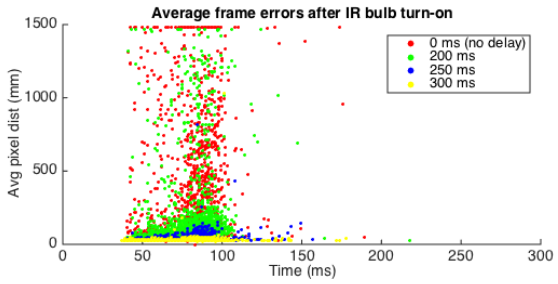


Fig. 6: Error introduced by the IR bulb forcing different time delays in milliseconds (ms) or none (0 ms)

a balance in the frames' quality/quantity trade-off, we decided to force a minimum time delay of 275 ms.

5.1.2 Vision module parameters

In the vision part, we decided to stick to the default parameters of [97] when possible. Those were the patch side size $N = 32$, the sampling stride $s = 5$, the number of spatial divisions within a trajectory, $n_x = n_y = 2$, the number of temporal divisions $n_t = 3$, the number of codes $k_{\text{vis}} = 4,000$, the sample of trajectories used to compute the codebooks of size 200K (100K per view), and SVM regularization parameter set to 1. These parameters have been largely validated in many action recognition datasets and thus guaranteed to provide state-of-the-art results. Notwithstanding, we set the trajectory length to a lower value of $L = 4$, more suited to our framerate than the original value of 15. This was done after experimentally testing different values for L in the set $\{3,4,5\}$.

Some parameters were fixed also for the computation of HON descriptors. The radius distance when computing the normal vectors was set to 2 cm, this being a standard value used in object recognition scenarios [3]. For the construction of the HON histogram, δ_θ and δ_φ were both set to 5.

For the negative mining of examples in the classification, we randomly sampled temporal segments having less than 0.2 of temporal overlap with activities' annotations. In particular, we sampled 10 negatives for each positive example. Moreover, we also included 1M negative trajectories during the generation of codebooks, apart from the 200K from the positive examples.

Finally, in order to determine the optimal set of weights ϖ to assign to the different modalities when computing RBF-$\mathcal{X}^2$ kernel during the action detection, we performed an exhaustive search in the training set. For this purpose, we generated all the possible 5-sized vectors of weights that sum up to 1 with incremental steps of 0.1 and evaluated average classification accuracy on the set of pre-segmented action gestures. Moreover, the weights were optimized independently for each action.

5.1.3 Wearable module parameters

Regarding the wearable module, there is also some parameters that needed to be validated. One of them is the parameter that controls the number of clusters (or prototypes) computed by K-Medoids. This depends on the complexity of each gesture class. We tested 1, 2, and 3 prototypes. No more classes are considered due to the reduced number of instances per gesture class.

When we are reconstructing our gesture predictions using DTW, it is common to have more than one reconstruction. This means that there are several matching paths that reached a cost below the threshold. This phenomenon is caused by having low values along the DTW matrix spread over their neighbour cells. A parameter regulates the activation of these reconstructions by thresholding the reconstruction cost. The tested values were 5, 10, 15, 20, 25, and 30 activations. Moreover, and in order to avoid short activations, we set a minimum length for considering a gesture. The values tested are 0, 10, 15, 20, 25, and 30 frames.

A tolerance parameter was introduced in Section 4.2.4 that controlled how many frames around a gesture end are considered so their values are put inside the positive bucket. Having a large value here will make our threshold be too large and then we will let pass a bigger number of false positives. This value has been experimentally set to 0.2.

Regarding the Montecarlo method, a number of samples has to be defined. We have set this value to 10K. As it can be seen in the following section, increasing the number of paths does not involve a large computational effort. The expected error rate is $E = \frac{1}{\sqrt{n}}$. By setting n to 10K, we expect an error rate of 0.01 which we consider is enough for the system.

5.2 Efficiency and computational cost

The vision module is based on the dense trajectories framework, which originally runs at 10-12 fps in VGA video. However, we extended the set of descriptors with HON, thus involving the computation of surface normals. The computation of normals is an expensive process when done naively, but can be greatly optimized by parallelizing computations or using approximated methods. If optimized, this module could run much faster than the 2-fps acquisition rate of the two Kinects.

In the case of the wearable module, there are two expensive processes: DTW matrix computation and Montecarlo threshold optimization. On the one hand, we have that the cost of the first is $O(n \cdot m)$, where m is the length of the model and n the length of the sequence. DTW matrix computation is not easily parallelizable. On the other hand, Montecarlo threshold optimization can be expensive if no optimization is applied. In our case, we have designed the algorithm in order to first precompute all the needed values, that is, when computing a path, the algorithm only has to select values, but do not compute them.

Let us analyze the computational cost of the Montecarlo method.

Let S be the number of sequences, M the number of models and P the number of randomly generated paths. For a gesture, let l_g be its length and for a model and l_M the length of each sub-model prototype. Note that these two quantities define the size of the dynamic time warping matrix, i.e. $l_g \times l_M$.

First of all, the reconstruction and the F1-score involving every pair of sequence-model pairs are pre-computed. The cost of this operations are:

- Dynamic time warping alignment for all models and sequences: $\mathcal{O}(S \cdot M \cdot l_g \cdot l_M)$
- Reconstruction and F1-score computation: $\mathcal{O}(S \cdot M \cdot l_g)$

For each of the paths, a model is selected at random for each of the sequences. Then, the F1-score is computed using the pre-computed reconstruction and the ground-truth. For each of the P paths we select among S choices, the cost of this computation is $\mathcal{O}(P \cdot S)$. Given that pre-computed F1-scores are stored in a hash table, for a certain sequence-model pair the cost of retrieving the score is $\mathcal{O}(1)$. This makes the global cost $\mathcal{O}(P \cdot S)$.

The computational cost of the whole Montecarlo method is $\mathcal{O}(P \cdot S)$

As a result, the computational cost of all the process is considering a sequential approach is:

$$O(S \cdot M \cdot l_g \cdot l_M) + O(S \cdot M \cdot l_g) + O(P \cdot S), \quad (8)$$

$$O(S \cdot M \cdot l_g \cdot l_M) + O(P \cdot S), \quad (9)$$

$$O(P + (S \cdot M \cdot l_g \cdot l_M)) \quad (10)$$

However, this method is easily parallelizable, both in generating reconstructions and computing F1-score, as the computation of each dynamic time warping matrix for each gesture and model is independent of the rest. Additionally, the global F1-score computation is also easily parallelized as it is independent of each of the paths.

As shown in [76], DTW is suitable to run in real-time since it only has to compute a column for each of the samples. In this case, having the sub-classes imply an increase in the number of DTW matrices computed. However, it could be easily parallelized and thus, the algorithm test phase could be run on-line.

5.3 Experimental results on SARQuavitae dataset

We validated the proposed system in the SARQuavitae dataset. We first explain the validation procedure and then we illustrate the results got by both the individual modules and the fused results from the fusion module.

In the experiments, we used a leave-one-subject-out cross-validation (LOSOCV) procedure in order to ensure a proper generalization of the methods. Besides, in order to validate any of the parameters described in 5.1, we used an internal cross-validation within the training partition of the LOSOCV.

We used two different metrics in order to quantify the performances: F1-score and intersection-over-union (IoU) overlap. During the computation of the F1-score, we use a 20% of minimum overlap in order to consider a true positive detection. These results were calculated at sequence level and averaged within the corresponding LOSOCV's fold. Then, the final performance was calculated averaging again the performances of all the folds.

We separately computed the performance for all the classes, so we can better illustrate performance issues and difficulties of the system. Regarding the F1-score, if neither the groundtruth nor the prediction presented any activation, we counted the F1-score to be 1.

In the case of IoU overlap, an additional parameter is also taken into account, which is the number of *do-not-care* frames. This value regulates the amount of discrepancy in the borders of the detections when evaluating respect to the groundtruth. Since, a system of this nature does not need a perfect matching but only a rough detection, we could afford using relatively large do-not-care values. The maximum do-not-care value used in the later validation is approximately half of the shorter action's duration, i.e., 50 frames.

5.3.1 Vision experiments

As a preliminary analysis was to determine somehow the contribution of the different modalities in the action detection task. Since we need to find the best set of ϖ weights for each subject, we can average the performance of all the weight combinations across the LOSOCV's training partitions. The set of weights selected per action is illustrated in Table 2. These weights provided

	Modalities					
Action name	Traj	HOG	HOF	MBH	HON	Accuracy
Drinking	0.5	0	0	0.3	0.2	91.54%
Eating	0.1	0.1	0	0.4	0.4	86.42%
Reading	0.2	0.1	0.1	0.1	0.5	91.57%
Takingpill	0	0	0.4	0.1	0.5	83.80%

Table 2: Best performing weight combinations for each of the classes (in terms of accuracy)

us some intuition about the contribution of HON. It demonstrated to be very relevant for the task of action recognition and complementary to the other modalities. Trj and MBH also demonstrated to be quite important, in contrast to HOF and HOG, with the latter being the less relevant in our dataset.

Once we had selected the weights, they were used in the action detection task. In Fig. 7 and Fig. 8, we illustrate the performance of the detection, respectively, in terms of F1-score and IoU overlap. These results show the performance of the vision component in detecting quite differs from one action to another. It is able to more successfully detect "eating" and "drinking" actions, while not doing so well at "eating" or "takingpill". Our hypothesis is that the vision part is better at detecting large actions than smaller ones. This causes overlap values to be larger, whereas in terms of F1 the detector is highly penalized.

5.3.2 Wearable experiments

Regarding the use of K-medoids and Montecarlo methods in order to have more than one model per class, we performed the same experiments with both modalities. In Fig. 9 and Fig. 10, we can see the improvement, respectively, over to Fig. 11 and 12. This is due to the better representation obtained by establishing different models per class. Given the nature of this dataset (recorded in the wild, without constraints), the intra-class variability is expected to be large. From the results presented in the supplemental material, it is observed that this happens even with the better representing features we have computed. From these results, we demonstrate the convenience of computing sub-classes in order to better model the gestures.

From the final results, one can see the classifier is outperformed in most off the cases by the wearable module, something we expected given the difficulties presented by the module in terms of intra-class variability. The most difficult class in terms of F1-score (concerned to the number of detections) is taking-pill. As it has been shown in the supplemental material, it is the one that is more confused against the others. Nonetheless, eating and drinking, that are the ones obtaining greater F1-scores, were the ones with less confusion as observed in the distance matrices.

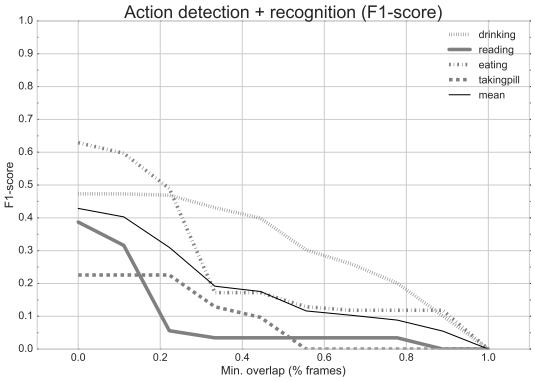


Fig. 7: F1-score for each class (and mean) and different minimum overlap values

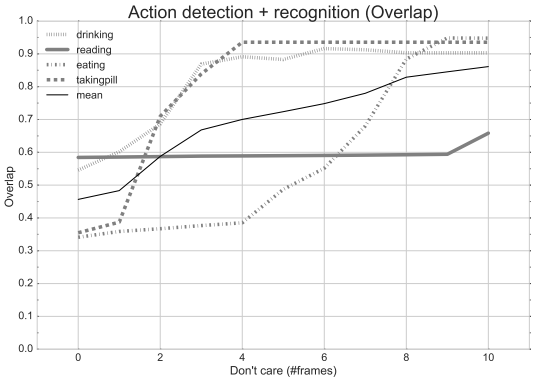


Fig. 8: Overlap for each class (and mean) and different *do-not-care* values

5.3.3 Integration experiments

The two modules, vision and inertial, are able to provide binary detection outputs for each of the classes in a particular time instant. Given that, intersection and union are possible alternatives to our learning-based fusion strategy to come up with the final integrated detection. We hence report these as baseline results to compare the learning-based integration.

Recall our goal is to detect activities, not precise temporal localization. Nonetheless, we analyze first overlap results. In Table 3 and Table 4, we show the results of the three different integration strategies in terms of both F1-score and IoU overlap respectively. We found the vision module performing individually was the most successful in 3 out of 4 classes. Nonetheless, the neural network is able to improve by 2% respect to the vision module.

More importantly, learning-based approach improves F1-score results respect single modalities or baseline

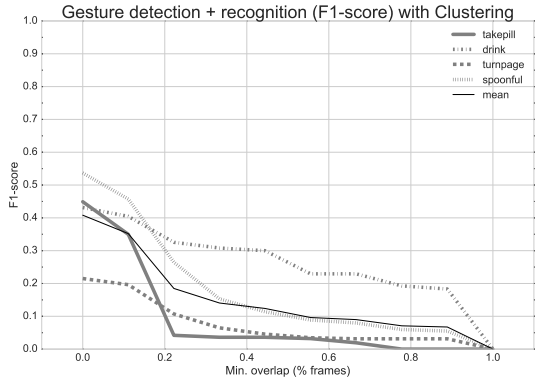


Fig. 9: F1-score for each class (and mean) and different minimum overlap values

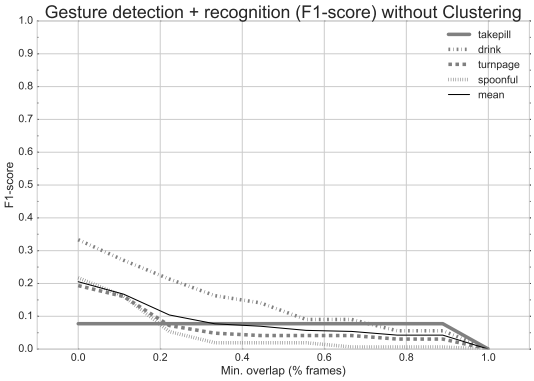


Fig. 11: F1-score for each class (and mean) and different minimum overlap values having one model per class

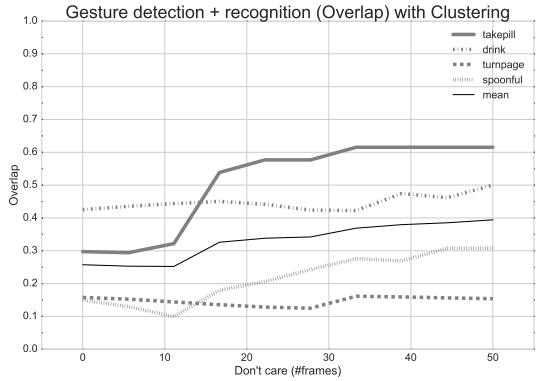


Fig. 10: Overlap for each class (and mean) and different *do-not-care* values

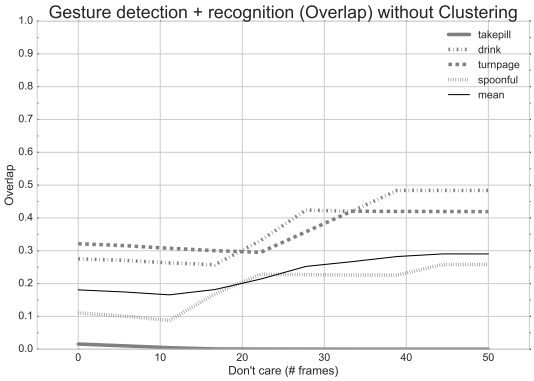


Fig. 12: Overlap for each class (and mean) and different *do-not-care* values having one model per class

integrations for all the classes except for "Drinking" (-4%) and obtained a particularly large improvement for "Taking-pill" (+12%) and "Reading" (+10%). In average, the learning-based fusion improves the vision module by 6%.

For the sake of completeness, we also illustrate the effect of varying the minimum overlap for TP in detection and the don't care size (varying the number of frames) have on learning-based (Fig. 13-14), intersection (Fig. 15-16), and union (Fig. 17-18) integration strategies.

6 Conclusion

We proposed a two-module system combining two Kinect devices together with a Shimmer in order to recognize activities of the daily living in a real-world scenario. The Kinect devices were placed one in front of another with intersecting frustums, so as to have an occlusion-free view of the scenario. We extracted first MmDT features and bagged them into multi-view codebooks, one for each kind of description. In order to perform the detection in this part, a sliding approach was used. The BOW representation allowed to compute each window's BOW representation in an "integral" efficient way. For classification of words, a multi-class SVM with $\mathcal{X}^2$-kernel was utilized in order to combine Traj, HOG, HOF, MBH, and HON descriptions at kernel level. The egocentric module, in charge of performing gesture recognition, used a Shimmer sensor placed on the elderly's dominant wrist. In order to recognize the gesture, we first pre-processed the data in order to extract accelerometer, rotation angles, and jerk features. Then, we select a set of models from the sequences. Those models were used to obtain alignment distances (costs) by means of a DTW algorithm. During the process of detecting the gestures, a DTW performed the alignments respect to the models and determined new gestures being performed by comparing on-line alignment costs to a set of learnt cost thresholds.

	Single modalities		Integration		
	Vision	Wearable	Intersection	Union	Learning-based
Taking-pill	**0.93**	0.61	0.87	0.51	0.54
Drinking	**0.89**	0.42	0.80	0.56	0.67
Eating	**0.38**	0.27	0.27	0.27	0.33
Reading	0.58	0.16	0.57	0.30	**0.60**
TOTAL (mean)	**0.69**	0.36	0.63	0.41	0.53

Table 3: Results in terms of overlap, for each of the classes and for all the integration strategies

	Single modalities		Integration		
	Vision	Wearable	Intersection	Union	Learning-based
Taking-pill	0.22	0.04	0.00	0.08	**0.34**
Drinking	**0.46**	0.32	0.30	0.37	0.42
Eating	0.48	0.26	0.06	0.22	**0.49**
Reading	0.05	0.10	0.04	0.05	**0.20**
TOTAL (mean)	0.30	0.18	0.10	0.18	**0.36**

Table 4: Results in terms of F1-score, for each of the classes and for all the integration strategies

The system was validated in dataset with actual elderly volunteers from the SARQuavitae Claret elder home, who participated performing different scripted scenarios involving the realization of some activities such as “taking pill”, “drinking”, “eating”, and “reading”. The learning-based integration proved to be the most successful strategy for activity detection, achieving better F1-score than single modalities or baseline integrations (union/intersection).

In fact, the results in terms of F1-score and IoU overlap indicate that our system is more effective for action detection than temporally predicting their temporal extent. For the latter task, the vision module performs better. The vision module uses action groundtruth annotations, while the wearable module uses gesture annotations. And hence, when fused, the temporal extents of actions predicted by the integration no longer coincide with the action groundtruth. Nonetheless, determining if the elder took the medication is far more important in a system of this kind than knowing the exact time frames in which the action occurred.

Regarding other contributions, we found the depth cue to have an important contribution to MmDTs. On the wearable side, the clustering strategy shown also its effectiveness.

Acknowledgements This work was partly supported by the spanish project TIN2016-74946-P and CERCA Programme / Generalitat de Catalunya. The work of Albert Clapés was supported by SUR-DEC of the Generalitat de Catalunya and FSE. We would also like to thank the SARQuavitae Claret elder home and all the people who volunteered for the recording of the dataset.

References

1. Adlam, T., Faulkner, R., Orpwood, R., Jones, K., Macijauskiene, J., Budraitiene, A.: The installation and support of internationally distributed equipment for people with dementia. IEEE transactions on information technology in biomedicine: a publication of the IEEE Engineering in Medicine and Biology Society **8**(3), 253–257 (2004)
2. Akl, A., Valaee, S.: Accelerometer-based gesture recognition via dynamic-time warping, affinity propagation, & compressive sensing. In: Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on, pp. 2270–2273. IEEE (2010)
3. Alexandre, L.A.: 3d descriptors for object and category recognition: a comparative evaluation. In: Workshop on Color-Depth Camera Fusion in Robotics at the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vilamoura, Portugal, vol. 1. Citeseer (2012)
4. Amft, O., Junker, H., Tr, G.: Detection of eating and drinking arm gestures using inertial body-worn sensors pp. 2–5 (2005)
5. Avci, A., Bosch, S., Marin-Perianu, M., Marin-Perianu, R., Havinga, P.: Activity recognition using inertial sensing for healthcare, wellbeing and sports applications: A survey. In: Architecture of computing systems (ARCS), 2010 23rd international conference on, pp. 1–10. VDE (2010)
6. Bagalà, F., Becker, C., Cappello, A., Chiari, L., Aminian, K., Hausdorff, J.M., Zijlstra, W., Klenk, J.: Evaluation of accelerometer-based fall detection algorithms on real-world falls. PloS one **7**(5), e37,062 (2012)
7. Bagheri, M., Gao, Q., Escalera, S., Clapes, A., Nasrollahi, K., Holte, M.B., Moeslund, T.B.: Keep it accurate and diverse: Enhancing action recognition performance by ensemble learning (2015)
8. Banerjee, T., Keller, J.M., Skubic, M., Stone, E.: Day or night activity recognition from video using fuzzy clustering techniques. Fuzzy Systems, IEEE Transactions on **22**(3), 483–493 (2014)
9. Bao, L., Intille, S.S.: Activity recognition from user-annotated acceleration data. In: Pervasive computing, pp. 1–17. Springer (2004)

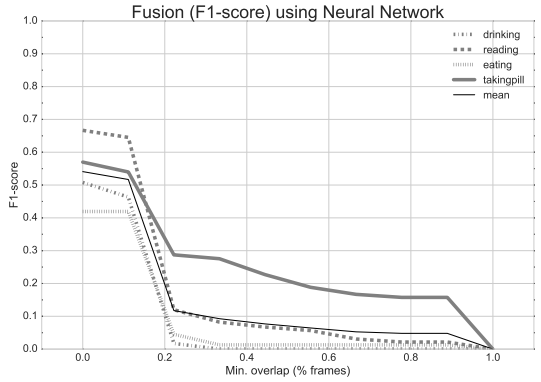


Fig. 13: F1-score for each class (and mean) and different minimum overlap using a Neural Network as the integration strategy

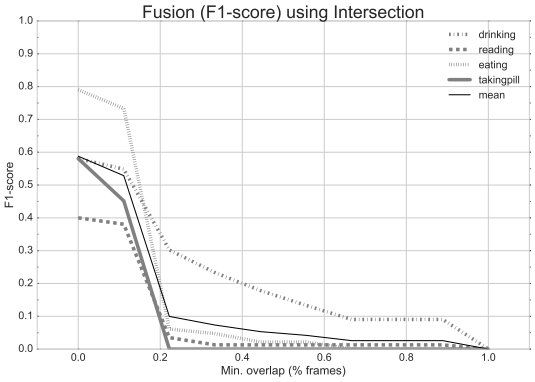


Fig. 15: F1-score for each class (and mean) and different minimum overlap using Intersection as the integration strategy

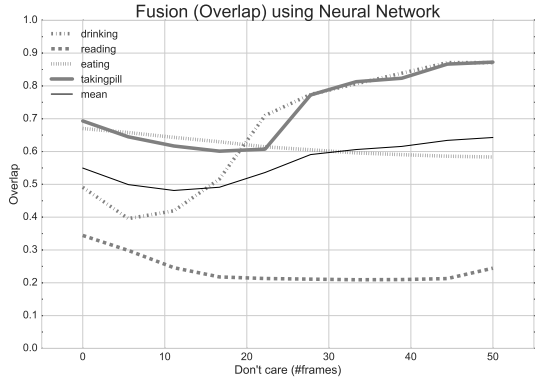


Fig. 14: Overlap for each class (and mean) and different *do-not-care* values using a Neural Network as the integration strategy

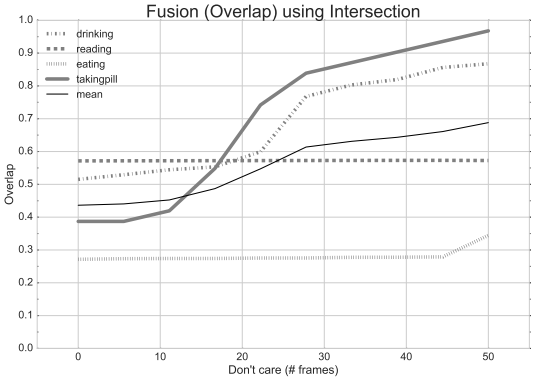


Fig. 16: Overlap for each class (and mean) and different *do-not-care* values using Intersection as the integration strategy

10. Barbosa, I.B., Cristani, M., Del Bue, A., Bazzani, L., Murino, V.: Re-identification with rgb-d sensors. In: Computer Vision–ECCV 2012. Workshops and Demonstrations, pp. 433–442. Springer (2012)
11. Bautista, M.A., Hernández-Vela, A., Ponce, V., Perez-Sala, X., Baró, X., Pujol, O., Angulo, C., Escalera, S.: Probability-based dynamic time warping for gesture recognition on rgb-d data. In: Advances in Depth Image Analysis and Applications, pp. 126–135. Springer (2013)
12. Ben Hadj Mohamed, A., Val, T., Andrieux, L., Kachouri, A.: Assisting people with disabilities through kinect sensors into a smart house. In: Computer Medical Applications (ICCMA), 2013 International Conference on, pp. 1–5. IEEE (2013)
13. Blank, M., Gorelick, L., Shechtman, E., Irani, M., Basri, R.: Actions as space-time shapes. In: Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on, vol. 2, pp. 1395–1402. IEEE (2005)
14. Bo, A., Hayashibe, M., Poignet, P.: Joint angle estimation in rehabilitation with inertial sensors and its integration with kinect. In: EMBC'11: 33rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 3479–3483. IEEE (2011)
15. Bobick, A.F., Davis, J.W.: The recognition of human movement using temporal templates. Pattern Analysis and Machine Intelligence, IEEE Transactions on **23**(3), 257–267 (2001)
16. Booranrom, Y., Watanapa, B., Mongkolnam, P.: Smart bedroom for elderly using kinect. In: Computer Science and Engineering Conference (ICSEC), 2014 International, pp. 427–432. IEEE (2014)
17. Botia, J.A., Villa, A., Palma, J.: Ambient assisted living system for in-home monitoring of healthy independent elders. Expert Systems with Applications **39**(9), 8136–8148 (2012)
18. Bouchard, K., Bilodeau, J.s., Fortin-simard, D., Gaboury, S.: Human Activity Recognition in Smart Homes Based on Passive RFID Localization (2014)
19. Brendel, W., Todorovic, S.: Activities as time series of human postures. In: Computer Vision–ECCV 2010, pp. 721–734. Springer (2010)
20. Bulling, A., Blanke, U.L.F., Schiele, B.: A Tutorial on Human Activity Recognition Using Body-Worn Inertial Sensors **46**(3), 1–33 (2014)
21. Casale, P.: Approximate ensemble methods for physical activity recognition applications. ELCVIA: electronic

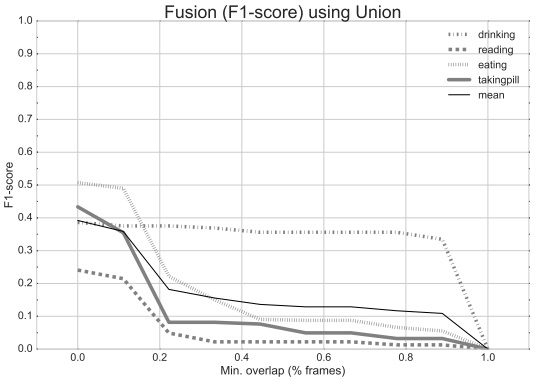


Fig. 17: F1-score for each class (and mean) and different minimum overlap values using Union as the integration strategy

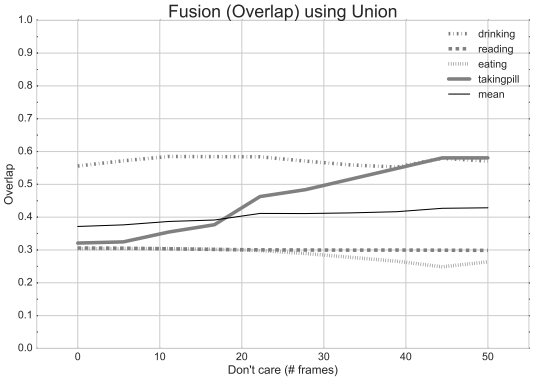


Fig. 18: Overlap for each class (and mean) and different *do-not-care* values using Union as the integration strategy

letters on computer vision and image analysis **13**(2), 22–23 (2014)
22. Chang, S.F., Ellis, D., Jiang, W., Lee, K., Yanagawa, A., Loui, A.C., Luo, J.: Large-scale multimodal semantic concept detection for consumer video. In: Proceedings of the international workshop on Workshop on multimedia information retrieval, pp. 255–264. ACM (2007)
23. Chattopadhyay, P., Roy, A., Sural, S., Mukhopadhyay, J.: Pose depth volume extraction from rgb-d streams for frontal gait recognition. Journal of Visual Communication and Image Representation **25**(1), 53–63 (2014)
24. Chen, C.C., Aggarwal, J.: Modeling human activities as speech. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, pp. 3425–3432. IEEE (2011)
25. Clapés, A., Reyes, M., Escalera, S.: Multi-modal user identification and object recognition surveillance system. Pattern Recognition Letters **34**(7), 799–808 (2013)
26. Crispim, C.F., Bathrinarayanan, V., Fosty, B., Konig, A., Romdhane, R., Thonnat, M., Bremond, F.: Evaluation of a monitoring system for event recognition of older people. In: Advanced Video and Signal Based Surveillance (AVSS), 2013 10th IEEE International Conference on, pp. 165–170. IEEE (2013)
27. Daponte, P., De Vito, L., Sementa, C.: A wireless-based home rehabilitation system for monitoring 3D movements. MeMeA 2013 - IEEE International Symposium on Medical Measurements and Applications, Proceedings pp. 282–287 (2013). DOI 10.1109/MeMeA.2013.6549753
28. Delachaux, B., Rebetez, J., Perez-Uribe, A., Mejia, H.F.S.: Indoor activity recognition by combining one-vs.-all neural network classifiers exploiting wearable and depth sensors. In: Advances in Computational Intelligence, pp. 216–223. Springer (2013)
29. Dell'Acqua, P., Klompstra, L.V., Jaarsma, T., Samini, A.: An assistive tool for monitoring physical activities in older adults. In: Serious Games and Applications for Health (SeGAH), 2013 IEEE 2nd International Conference on, pp. 1–6. IEEE (2013)
30. Dubois, A., Charpillet, F.: Human activities recognition with rgb-depth camera using hmm. In: Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE, pp. 4666–4669. IEEE (2013)
31. Escalera, S., Bar, X., Gonzlez, J., Bautista, M.A., Madadi, M., Reyes, M., Ponce, V., Escalante, H.J., Shotton, J., Guyon, I.: Chalearn looking at people challenge 2014: Dataset and results. In: ECCV Workshops (2014)
32. Farnebäck, G.: Two-frame motion estimation based on polynomial expansion. In: Image analysis, pp. 363–370. Springer (2003)
33. Fenu, G., Steri, G.: IMU based post-traumatic rehabilitation assessment. 2010 3rd International Symposium on Applied Sciences in Biomedical and Communication Technologies, ISABEL 2010 pp. 1–4 (2010). DOI 10.1109/ISABEL.2010.5702813
34. Fernandez-Sanchez, E.J., Diaz, J., Ros, E.: Background subtraction based on color and depth using active sensors. Sensors **13**(7), 8895–8915 (2013)
35. Gaidon, A., Harchaoui, Z., Schmid, C.: Temporal localization of actions with actoms. Pattern Analysis and Machine Intelligence, IEEE Transactions on **35**(11), 2782–2795 (2013)
36. Georgi, M., Amma, C., Schultz, T.: Recognizing Hand and Finger Gestures with IMU based Motion and EMG based Muscle Activity Sensing. Proceedings of the International Conference on Bio-inspired Systems and Signal Processing pp. 99–108 (2015). DOI 10.5220/0005276900990108. URL http://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0005276900990108
37. Gkioxari, G., Malik, J.: Finding action tubes. arXiv preprint arXiv:1411.6031 (2014)
38. Golby, C., Raja, V., Lewando Hundt, G., Badiyani, S.: A low cost 'activities of daily living' assessment system for the continual assessment of post-stroke patients, from inpatient/outpatient rehabilitation through to telerehabilitation (2011)
39. Gunes, H., Piccardi, M.: Affect recognition from face and body: early fusion vs. late fusion. In: Systems, Man and Cybernetics, 2005 IEEE International Conference on, vol. 4, pp. 3437–3443 Vol. 4 (2005). DOI 10.1109/ICSMC.2005.1571679
40. Helten, T., Muller, M., Seidel, H.P., Theobalt, C.: Real-time body tracking with one depth camera and inertial sensors. In: Computer Vision (ICCV), 2013 IEEE International Conference on, pp. 1105–1112. IEEE (2013)

41. Hernández-Vela, A., Bautista, M.A., Perez-Sala, X., Ponce, V., Baró, X., Pujol, O., Angulo, C., Escalera, S.: Bovdw: Bag-of-visual-and-depth-words for gesture recognition. In: Pattern Recognition (ICPR), 2012 21st International Conference on, pp. 449–452. IEEE (2012)
42. Hondori, H.M., Khademi, M., Lopes, C.V.: Monitoring intake gestures using sensor fusion (microsoft kinect and inertial sensors) for smart home tele-rehab setting. In: 2012 1st Annual IEEE Healthcare Innovation Conference (2012)
43. Hongeng, S., Nevatia, R., Bremond, F.: Video-based event recognition: activity representation and probabilistic recognition methods. Computer Vision and Image Understanding **96**(2), 129–162 (2004)
44. Jafari, R., Li, W., Bajcsy, R., Glaser, S., Sastry, S.: Physical activity monitoring for assisted living at home. In: 4th International Workshop on Wearable and Implantable Body Sensor Networks (BSN 2007), pp. 213–219. Springer (2007)
45. Jain, M., Van Gemert, J., Jégou, H., Bouthemy, P., Snoek, C.G.: Action localization with tubelets from motion. In: Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, pp. 740–747. IEEE (2014)
46. Jhuang, H., Gall, J., Zuffi, S., Schmid, C., Black, M.J.: Towards understanding action recognition. In: Computer Vision (ICCV), 2013 IEEE International Conference on, pp. 3192–3199. IEEE (2013)
47. Junker, H., Amft, O., Lukowicz, P., Tröster, G.: Gesture spotting with body-worn inertial sensors to detect user activities **41**(2008), 2010–2024 (2010). DOI 10.1016/j.patcog.2007.11.016
48. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale video classification with convolutional neural networks. In: Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, pp. 1725–1732. IEEE (2014)
49. Ke, Y., Sukthankar, R., Hebert, M.: Volumetric features for video event detection. International journal of computer vision **88**(3), 339–362 (2010)
50. Kim, J., Yang, S., Gerla, M.: Stroketrack: wireless inertial motion tracking of human arms for stroke telerehabilitation. In: Proceedings of the First ACM Workshop on Mobile Systems, Applications, and Services for Healthcare, p. 4. ACM (2011)
51. Kim, T.K., Cipolla, R.: Canonical correlation analysis of video volume tensors for action categorization and detection. Pattern Analysis and Machine Intelligence, IEEE Transactions on **31**(8), 1415–1428 (2009)
52. Kong, W., Sessa, S., Cosentino, S., Zecca, M., Saito, K., Wang, C., Imtiaz, U., Lin, Z., Bartolomeo, L., Ishii, H., Ikai, T., Takanishi, A.: Development of a real-time IMU-based motion capture system for gait rehabilitation (December), 2100–2105 (2013)
53. Kratz, S., Back, M.: Towards Accurate Automatic Segmentation of IMU-Tracked Motion Gestures pp. 1337–1342 (2015)
54. Kwolek, B., Kepski, M.: Improving fall detection by the use of depth sensor and accelerometer. Neurocomputing **168**, 637–645 (2015)
55. Laptev, I.: On space-time interest points. International Journal of Computer Vision **64**(2-3), 107–123 (2005)
56. Laptev, I., Marszałek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, pp. 1–8. IEEE (2008)
57. Lara, D., Labrador, M.a.: A Survey on Human Activity Recognition using **15**(3), 1192–1209 (2013)
58. Lei, J., Ren, X., Fox, D.: Fine-grained kitchen activity recognition using rgb-d. In: Proceedings of the 2012 ACM Conference on Ubiquitous Computing, pp. 208–211. ACM (2012)
59. Li, B.Y., Mian, A.S., Liu, W., Krishna, A.: Using kinect for face recognition under varying poses, expressions, illumination and disguise. In: Applications of Computer Vision (WACV), 2013 IEEE Workshop on, pp. 186–192. IEEE (2013)
60. Li, W., Zhang, Z., Liu, Z.: Action recognition based on a bag of 3d points. In: Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on, pp. 9–14. IEEE (2010)
61. Liang, B., Zheng, L.: Spatio-temporal pyramid cuboid matching for action recognition using depth maps. In: International Conference on Image Processing 2015 (ICIP 2015) (2015)
62. Liu, J., Zhong, L., Wickramasuriya, J., Vasudevan, V.: uwave: Accelerometer-based personalized gesture recognition and its applications. Pervasive and Mobile Computing **5**(6), 657–675 (2009)
63. Liu, K., Chen, C., Jafari, R., Kehtarnavaz, N.: Fusion of inertial and depth sensor data for robust hand gesture recognition. Sensors Journal, IEEE **14**(6), 1898–1903 (2014)
64. Lombriser, C., Bharatula, N.B., Roggen, D., Tröster, G.: On-body activity recognition in a dynamic sensor network. In: Proceedings of the ICST 2nd international conference on Body area networks, p. 17. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering) (2007)
65. Luinge, H.J., Veltink, P.H.: Measuring orientation of human body segments using miniature gyroscopes and accelerometers. Medical and Biological Engineering and computing **43**(2), 273–282 (2005)
66. Mace, D., Gao, W., Coskun, A.: Accelerometer-based hand gesture recognition using feature weighted naïve bayesian classifiers and dynamic time warping. In: Proceedings of the companion publication of the 2013 international conference on Intelligent user interfaces companion, pp. 83–84. ACM (2013)
67. Memon, M., Wagner, S.R., Pedersen, C.F., Beevi, F.H.A., Hansen, F.O.: Ambient assisted living healthcare frameworks, platforms, standards, and quality attributes. Sensors **14**(3), 4312–4341 (2014)
68. Mogelmose, A., Bahnsen, C., Moeslund, T.B., Clapés, A., Escalera, S.: Tri-modal person re-identification with rgb, depth and thermal features. In: Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on, pp. 301–307. IEEE (2013)
69. Mubashir, M., Shao, L., Seed, L.: A survey on fall detection: Principles and approaches. Neurocomputing **100**, 144–152 (2013)
70. Nait-Charif, H., McKenna, S.J.: Activity summarisation and fall detection in a supportive home environment. In: Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on, vol. 4, pp. 323–326. IEEE (2004)
71. Natarajan, P., Wu, S., Vitaladevuni, S., Zhuang, X., Tsakalidis, S., Park, U., Prasad, R., Natarajan, P.: Multimodal feature fusion for robust event detection in web videos. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, pp. 1298–1305 (2012). DOI 10.1109/CVPR.2012.6247814

72. Ni, B., Wang, G., Moulin, P.: Rgbd-hudaact: A color-depth video database for human daily activity recognition. In: Consumer Depth Cameras for Computer Vision, pp. 193–208. Springer (2013)
73. Nikisins, O., Nasrollahi, K., Greitans, M., Moeslund, T.B.: Rgb-dt based face recognition. In: Pattern Recognition (ICPR), 2014 22nd International Conference on, pp. 1716–1721. IEEE (2014)
74. Oliver, N., Garg, A., Horvitz, E.: Layered representations for learning and inferring office activity from multiple sensory channels. Computer Vision and Image Understanding **96**(2), 163–180 (2004)
75. Organization, W.H., International, A.D.: Dementia: a public health priority. World Health Organization (2012)
76. Pardo, À., Clapés, A., Escalera, S., Pujol, O.: Actions in context: System for people with dementia. In: Citizen in Sensor Networks, pp. 3–14. Springer (2014)
77. Piyathilaka, L., Kodagoda, S.: Gaussian mixture based hmm for human daily activity recognition using 3d skeleton features. In: Industrial Electronics and Applications (ICIEA), 2013 8th IEEE Conference on, pp. 567–572. IEEE (2013)
78. Pylvänäinen, T.: Accelerometer based gesture recognition using continuous hmms. In: Pattern Recognition and Image Analysis, pp. 639–646. Springer (2005)
79. Rashidi, P., Cook, D.J.: Keeping the resident in the loop: Adapting the smart home to the user. Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on **39**(5), 949–959 (2009)
80. Reyes, M., Domínguez, G., Escalera, S.: Featureweighting in dynamic timewarping for gesture recognition in depth data. In: Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on, pp. 1182–1188. IEEE (2011)
81. Ribeiro, P.C., Santos-Victor, J.: Human activity recognition from video: modeling, feature selection and classification architecture. In: Proceedings of International Workshop on Human Activity Recognition and Modelling, pp. 61–78. Citeseer (2005)
82. Riboni, D., Bettini, C.: COSAR: Hybrid reasoning for context-Aware activity recognition. Personal and Ubiquitous Computing **15**(3), 271–289 (2011). DOI 10.1007/s00779-010-0331-7
83. Rodriguez, M.D., Ahmed, J., Shah, M.: Action mach a spatio-temporal maximum average correlation height filter for action recognition. In: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, pp. 1–8. IEEE (2008)
84. Sadanand, S., Corso, J.J.: Action bank: A high-level representation of activity in video. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, pp. 1234–1241. IEEE (2012)
85. Saha, S., Pal, M., Konar, A., Janarthanan, R.: Neural network based gesture recognition for elderly health care using kinect sensor. In: Swarm, Evolutionary, and Memetic Computing, pp. 376–386. Springer (2013)
86. Schindler, K., Van Gool, L.: Action snippets: How many frames does human action recognition require? In: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, pp. 1–8. IEEE (2008)
87. Schüldt, C., Laptev, I., Caputo, B.: Recognizing human actions: a local svm approach. In: Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on, vol. 3, pp. 32–36. IEEE (2004)
88. Shotton, J., Sharp, T., Kipman, A., Fitzgibbon, A., Finocchio, M., Blake, A., Cook, M., Moore, R.: Real-time human pose recognition in parts from single depth images. Communications of the ACM **56**(1), 116–124 (2013)
89. Snoek, C.G., Worring, M., Smeulders, A.W.: Early versus late fusion in semantic video analysis. In: Proceedings of the 13th annual ACM international conference on Multimedia, pp. 399–402. ACM (2005)
90. Sung, J., Ponce, C., Selman, B., Saxena, A.: Unstructured human activity detection from rgbd images. In: Robotics and Automation (ICRA), 2012 IEEE International Conference on, pp. 842–849. IEEE (2012)
91. Tang, K., Fei-Fei, L., Koller, D.: Learning latent temporal structure for complex event detection. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, pp. 1250–1257. IEEE (2012)
92. Tang, S., Wang, X., Lv, X., Han, T.X., Keller, J., He, Z., Skubic, M., Lao, S.: Histogram of oriented normal vectors for object recognition with a depth sensor. In: Computer Vision–ACCV 2012, pp. 525–538. Springer (2012)
93. Ullah, M.M., Parizi, S.N., Laptev, I.: Improving bag-of-features action recognition with non-local cues. In: BMVC, vol. 10, pp. 95–1. Citeseer (2010)
94. Van Hoof, J., Kort, H., Rutten, P., Duijnstee, M.: Ageing-in-place with the use of ambient intelligence technology: Perspectives of older users. International journal of medical informatics **80**(5), 310–331 (2011)
95. Vedaldi, A., Zisserman, A.: Efficient additive kernels via explicit feature maps. Pattern Analysis and Machine Intelligence, IEEE Transactions on **34**(3), 480–492 (2012)
96. Vintsyuk, T.K.: Speech discrimination by dynamic programming. Cybernetics and Systems Analysis **4**(1), 52–57 (1968)
97. Wang, H., Kläser, A., Schmid, C., Liu, C.L.: Action recognition by dense trajectories. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, pp. 3169–3176. IEEE (2011)
98. Wang, H., Schmid, C.: Action recognition with improved trajectories. In: Computer Vision (ICCV), 2013 IEEE International Conference on, pp. 3551–3558. IEEE (2013)
99. Weinzaepfel, P., Harchaoui, Z., Schmid, C.: Learning to track for spatio-temporal action localization. arXiv preprint arXiv:1506.01929 (2015)
100. Wu, D., Zhu, F., Shao, L.: One shot learning gesture recognition from rgbd images. In: Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on, pp. 7–12. IEEE (2012)
101. Wu, J., Osuntogun, A., Choudhury, T., Philipose, M., Rehg, J.M.: A scalable approach to activity recognition based on object use. In: Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on, pp. 1–8. IEEE (2007)
102. Xu, Z., Yang, Y., Hauptmann, A.G.: A discriminative cnn video representation for event detection. arXiv preprint arXiv:1411.4006 (2014)
103. Zhang, B., Jiang, S., Wei, D., Marschollek, M., Zhang, W.: State of the art in gait analysis using wearable sensors for healthcare applications. In: Computer and Information Science (ICIS), 2012 IEEE/ACIS 11th International Conference on, pp. 213–218. IEEE (2012)
104. Zhang, C., Tian, Y.: Rgb-d camera-based daily living activity recognition. Journal of Computer Vision and Image Processing **2**(4), 12 (2012)

105. Zhao, Y., Liu, Z., Yang, L., Cheng, H.: Combing rgb and depth map features for human activity recognition. In: Signal & Information Processing Association Annual Summit and Conference (APSIPA ASC), 2012 Asia-Pacific, pp. 1–4. IEEE (2012)
106. Zhou, F., Jiao, J., Chen, S., Zhang, D.: A case-driven ambient intelligence system for elderly in-home assistance applications. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on **41**(2), 179–189 (2011)
107. Zhu, C., Sheng, W.: Multi-sensor fusion for human daily activity recognition in robot-assisted living. In: Proceedings of the 4th ACM/IEEE international conference on Human robot interaction, pp. 303–304. ACM (2009)

A Appendix: Wearable Module feature comparison

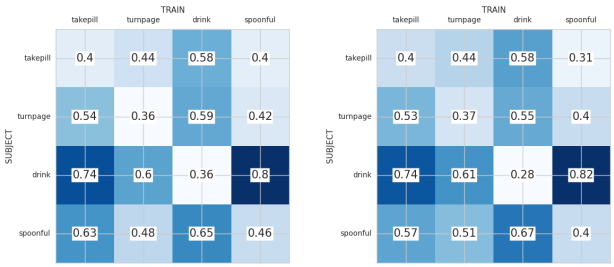
In Section 4.2, several features are detailed. They are used to describe the gestures performed among different magnitudes. Here, an extensive comparison between different features' combinations is presented, with the objective of demonstrating which is the most suitable for performing gesture recognition using the SAR-Quavitae Claret dataset.

Given the set of features described in the article, all their possible combinations have been generated. For each feature set, we have computed the distances between each of the pre-segmented gestures, using DTW as a metric.

Each of the matrices represent the average distance between segmented gestures, using leave-one-subject-out strategy. That is, for each subject, all his gestures are compared to the remaining ones. Finally, the average of all the distances is computed. The objective is to find the combination that is more discriminative. This means that, the best features will be those that minimize the distance between equal classes (diagonal of the matrices), while maximizing the distance against different classes.

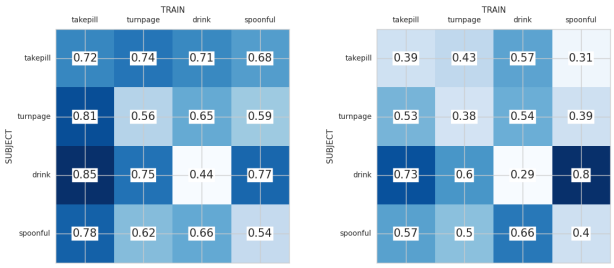
In most of the combinations showed, one can see that "taking-pill" and "turn-page" gestures are easily confused, while "drink" and "spoonful" are, most of the times, distant from the other classes. Looking the matrices with more detail, Figures 20b,21a,21b, and 23a lead us to the conclusion that *Raw accelerometer* is crucial for representing the data correctly. Regarding to those combinations only using a pair of features, Figures 19a, 19b, and 20a show us that these combinations are usually enough to discriminate correctly "drink" and "spoonful" classes, but are not enough for "taking-pill" and "turn-page". The same happens in Figures 22a and 22b, where "taking-pill" and "turn-page" are confused with "spoonful". The combination that is able to discriminate correctly "turn-page", "drink", and "spoonful" is the one showed in Figure 23b. However, "taking-pill" is still close to the other classes. This is the effect of the variability in the performance of the gesture.

Finally, thanks to this deep analysis, we can state that the most suitable combination of features for the presented dataset is the one including *Raw accelerometer*, *Sorted accelerometer*, *Complementary filter*, and *Jerk*. However, the distance matrices presented in this section glimpses some issues related to the intra-class variability. Nevertheless, our hypothesis is that this is due to the reduced set of data available, that hinders the representativeness of the gestures.



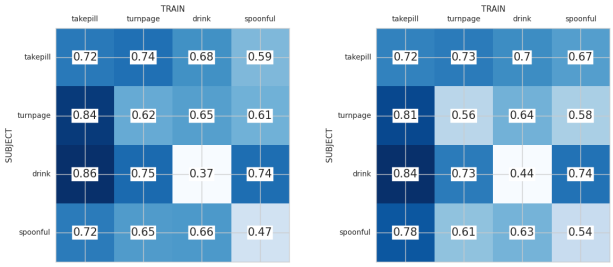
(a) (b)

Fig. 19: In (a), raw accelerometer + sorted accelerometer. In (b), raw accelerometer + complementary accelerometer



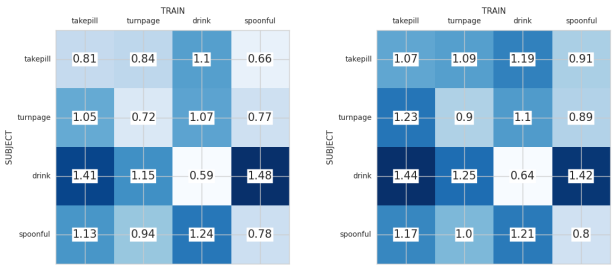
(a) (b)

Fig. 20: In (a), raw accelerometer + jerk. In (b), sorted accelerometer + complementary filter



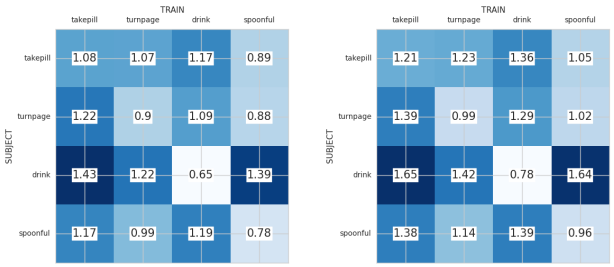
(a) (b)

Fig. 21: In(a), sorted accelerometer + jerk. In (b), complementary filter + jerk



(a) (b)

Fig. 22: In (a), raw accelerometer + sorted accelerometer + complementary filter. In (b), raw accelerometer + sorted accelerometer + jerk



(a) (b)

Fig. 23: In (a), sorted accelerometer + complementary filter + jerk. In (b), raw accelerometer + sorted accelerometer + complementary filter + jerk