Recognition of Activities of Daily Living for Smart Home Environments

Konstantinos Avgerinakis,
Information Technologies Institute, CERTH
Centre for Vision, Speech and Signal Processing
University of Surrey, UK
koafgeri@iti.gr

Alexia Briasouli, Ioannis Kompatsiaris
Information Technologies Institute, CERTH
Thessaloniki, Greece
abria@iti.gr, ikom@iti.gr

***Abstract*—The recognition of Activities of Daily Living (ADL) from video can prove particularly useful in assisted living and smart home environments, as behavioral and lifestyle profiles can be constructed through the recognition of ADLs over time. Often, existing methods for recognition of ADLs have a very high computational cost, which makes them unsuitable for real time or near real time applications. In this work we present a novel method for recognizing ADLs with accuracy comparable to the state of the art, at a lowered computational cost. Comprehensive testing of the best existing descriptors, encoding methods and BoW/SVM based classification methods takes place to determine the optimal recognition solution. A statistical method for determining the temporal duration of extracted trajectories is also introduced, to streamline the recognition process and make it less ad-hoc. Experiments take place with benchmark ADL datasets and a newly introduced set of ADL recordings of elderly people with dementia as well as healthy individuals. Our algorithm leads to accurate recognition rates, comparable or better than the State of the Art, at a lower computational cost.**

I. INTRODUCTION

The problem of human activity recognition is central in computer vision, attracting significant research attention. It is of particular interest in smart homes and assisted living situations, as videos of a person's daily life cannot be effectively parsed by human operators due to their huge volume and the fact that they often contain long segments with uninteresting content. Effective and accurate recognition of ADLs can play a very significant role in determining a person's condition and its progression over time, and is the main goal of this work, where focus is placed on the recognition of ADLs with high accuracy and lowered computational cost, both in lab and realistic environments.

Numerous methods have been developed for activity recognition in recent years, with space-time feature based approaches being among the most popular ones for video representation [1], which, in combination with Bag of Features methods, lead to State of the Art (SoA) results. Feature- based methods have been used successfully in object recognition, which motivated the development of numerous types of features now being tested for activity recognition. Usually action recognition is based on sparse features to reduce the computational cost, but results in performance degradation. Recent methods use dense interest point tracking [3], as it provides more information about the scene and, consequently, more accurate recognition results.

.

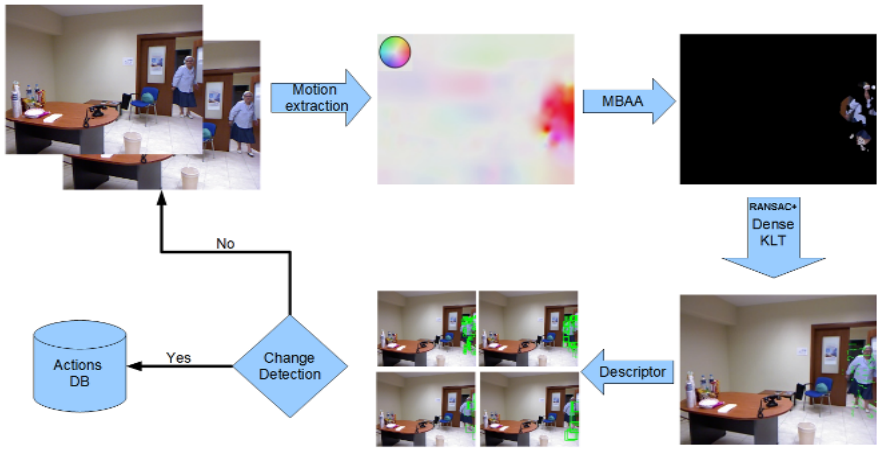


Fig 1. *Feature Extraction: MBAAs localize pixels where velocity changes. Dense sampling is applied to MBAAs to extract interest points. HOG and HOF are estimated around interest points that are tracked by KLT. RANSAC is applied to eliminate outlier matches. CUSUM is applied to detect changes in motion and automatically determine the temporal extent of each trajectory.*

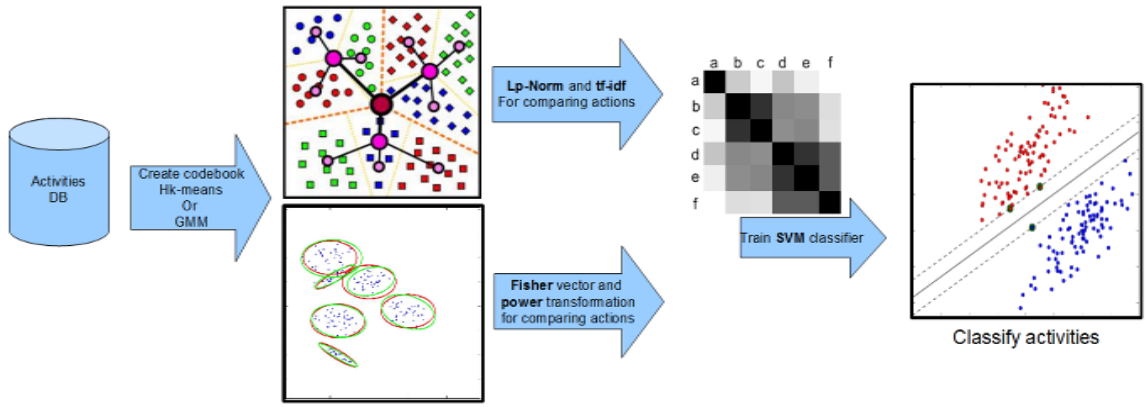


Fig 2. *Action Recognition: local descriptor space is partitioned via k-means and GMM clustering of training data. Several encoding techniques are tested: Lp-Norm with a tf-idf implementation, Fisher transformation and hard binning with a Chi-square Kernel. A multiclass SVM is used for creating the appropriate model that will discriminate the action descriptors.*

Our approach is inspired from Wang [3], who achieves SoA results by extracting dense trajectories. We use dense trajectories to recognize activities of daily living with accuracy, but also reduce the computational cost when extracting and training the features that will describe the actions. For this purpose, we introduce a novel binary mask, called Motion Boundary Activity Area (MBAA), which localizes regions where the velocity *changes* in relation to its neighborhood. A multi-scale dense grid samples interest points in the MBAAs and action descriptors are consequently produced, based on Histograms of Oriented Gradients (HOG) and Histograms of Optical Flow (HOF). Interest points are tracked using the KLT tracker [4] and RANSAC is used to correctg erroneous correspondences. Fig. 1 and Fig. 2 below depict the feature extraction and action recognition frameworks, respectively.

A significant innovation in our method is the use of statistical sequential change detection to determining the temporal length of each trajectory. The motion histograms of each trajectory are used to model its velocity distribution and statistical sequential change detection, namely the Cumulative Sum (CUSUM) method is applied to automatically detect when the velocity distribution changes, to find the correct temporal extent of each sub-activity

We go beyond the state-of-the-art [1, 2, 3, 6, 7] by testing several techniques for vocabulary building and encoding in the most informative manner. We tested k-means and GMM clustering for a rich, discriminative vocabulary. Additionally, we tested a wide range of encoding schemas, like the classic hard binning followed by Chi-square, a smart similarity scoring technique that uses an inverted index for fast acquisition and the Fisher transformation, for comparing video sequences. An SVM classifier is finally deployed to recognize the actions found in the test data.

This paper is organized as follows: Section II describes the general methodology used, and more specifically the feature extraction approach, the introduction of Motion Boundary Activity Areas, the resulting spatiotemporal descriptor and finally, the statistical sequential change detection method applied to determine each trajectory's temporal extent. Section III focuses on the vocabulary building and encoding used for the Bag of Features approach that we use to characterize and then recognize activities in videos. Experimental results with benchmark and new, realistic datasets are presented in Section IV and conclusions are drawn in Section V.

II. METHODOLOGY

A. Feature Extraction

The most important part for describing an action is to represent it in the most discriminative way. Background clutter, great anthropometric variance (different height, different clothes), different camera viewpoints should be dealt with to obtain the most robust and accurate recognition result.

The first step in feature extraction is to sample meaningful spatio-temporal interest points where the motion will be described. Most state-of-the-art methods [2, 7] use the temporal extension of the Harris corner detector, which was introduced in [10]. However, a great limitation of this detector is that all geometric information is lost during the Bag of Words (BoW) step and it is not suitable for action sequences that do not contain enough repeatable space-time corners, such as aperiodic motions. Other techniques [3, 11] sample interest points and track them throughout time. Sampling may occur either densely or by using a corner detector, however both of them proved to be computationally expensive and cannot be used in real time.

The next step of feature extraction includes the description of the action. Most approaches extend image descriptors over time, producing spatiotemporal cuboids that represent the actions. Thus for actions that can be described mainly by their **appearance** characteristics, extensions of the SIFT descriptor [6] and HOG descriptor [7] are used. Motion characteristics can be included via Optical Flow, as in Laptev [12], where

HOGHOF features were introduced. Other descriptors include dense trajectory information [3], or velocity history [11].

In this work we propose a novel schema for interest point detection that is based on the statistical analysis of the motion vectors over time. This technique drastically reduces the computational cost of the algorithm without degrading its discriminative power. A binary mask which localizes *changes in motion* (in contrast to the commonly used foreground masks, which localize changes in luminance over time) is first introduced. Multi-scale grids are formed and dense sampling within the precomputed mask is performed, producing a spate of spatio-temporal interest points, robust to scale variance. These points are then tracked over time by using a recent pyramidal implementation of the KLT tracker [4]. Outlier correspondences are eliminated by RANSAC, increasing the algorithm's accuracy. We use the HOGHOF descriptor for representing actions because of its low computational cost. Finally, we automatically determine the temporal extent of the trajectories by the application of statistical sequential change detection of each trajectory's velocity values over time.

B. Motion Boundary Activity Area

Motion boundary activity areas (MBAAs) are an extension of Activity Areas that were introduced in [5]. In this step, we analyze the Optical Flow gradients, which indicate the regions where motion *changes* (Motion Boundaries). *This leads to a smaller region of interest, without losing discriminative power.* Thus, we obtain a binary mask that separates motion regions that are produced from changes in motion due to a subject's activity (e.g. a person in a smart home or another assisted living situation) and not from background clutter or illumination variance. We assume that data induced by noise follows a Gaussian distribution (hypothesis H_0), while changes in motion introduce deviations from Gaussianity (hypothesis H_1), as expressed below:

$$H_0 : u_k^0(r) = z_k(r)$$

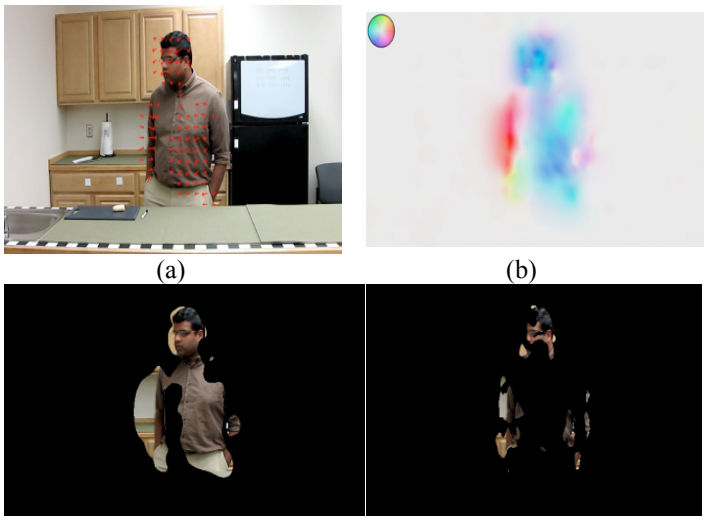
$$H_1 : u_k^1(r) = u_k(r) + z_k(r) ,$$

where $u_k(r)$, $z_k(r)$ denote actual optical flow values and noise in the flow, respectively. The kurtosis measure of Gaussian data is known to be equal to zero and can thus be used to detect whether motion boundaries are caused by noise or by changes in optical flow, in order to localize regions of changing motion. We extend the work of [5] by using the unbiased kurtosis estimator [13], given by:

$$G_2[y] = \frac{3}{W(W-1)} \sum_{i=1}^{W} (u_i(r)^4) - \frac{W+2}{W(W-1)} \left(\sum_{i=1}^{W} (u_i(r)^2)^2\right)$$

where W is a manually chosen temporal window from which data (motion values) are obtained. The kurtosis values will be significantly higher in regions of pixels whose motion is changing, so the binarized 2D kurtosis "maps" of each frame, called Motion Boundary Activity Areas (MBAAs) indicate which pixels are undergoing changing motion.

Fig 3 shows the optical flow vectors with direction and magnitude, the original activity area (AA) and the motion boundary activity area (MBAA). We can clearly see the reduced size of the binary mask in the second case.



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

Fig 3. (a) Direction vectors of Optical Flow. (b) Magnitude and coloured direction of Optical Flow. (c) Activity area. (d) Motion boundary activity area. We can see the reduced size of the MBAA, which leads to lower computational cost whithout losing its discriminate power.

C. The SpatioTemporal Descriptor

The next goal of our algorithm is to detect informative spatio-temporal interest points so that we can build a robust descriptor. For this purpose we build 4-scale grids in a pyramidal manner. Within these grids we analyze the values that the binary mask produces. If more than 50% of the values have a positive flag and no other interest points coexist in the block than we declare it as moving and use its center as an interest point. The collected interest points are tracked throughout time using a pyramidal implementation of the renowned KLT tracker [4] and their correspondence is verified by computing the homography of the two interest point sets.

Action trajectories are formed by accumulating the correct interest point correspondences over time. Around each interest point, we compute the HOG and HOF descriptors, acquiring appearance and motion information respectively and building, in this manner, an Action Cuboid. Both descriptors are computed in 4 scales kσ (every kσ = 8, 16, 24, 32 pixels), around each interest point to guarantee scale invariance. In order to maintain the spatial localization of each descriptor, we divide each block into 2x2 cells. The resulting histograms that are computed within the cells are accumulated and normalized, leading to a global motion histogram, which still contains localization information.

Temporal trajectory length is calculated by applying statistical sequential change detection on the HOF histograms. This technique extracts points in time where motion changes abruptly. This leads to trajectories of a different, optimally derived temporal extent, depending on the motion changes in each activity. This technique is further analyzed in the next section. After the trajectory acquisition, we build our spatio-temporal descriptor by stacking all the trajectory descriptors. For providing a common dimension descriptor for each

trajectory, we divide the trajectory structure in three parts and average the histograms within them. Thus, taking into account that we use 9 bins for each HOG and HOF histogram, we produce a 216 dimensional descriptor for each trajectory.

D. Change Detection for Optimal Trajectories

As mentioned in the previous paragraph, one of the innovations of our method is the computation of optimal trajectories, acquired by analyzing motion cues. For this purpose, we implemented a change detection algorithm inspired by [14], which analyzes the motion histograms taken of each trajectory's HOF descriptors. Sequential change detection, and in particular the Cumulative Sum (CUSUM) method is applied to each trajectory for determining its optimal length.

An initial distribution f_0 of the velocities of the tracked points is firstly computed from the first w_0 frames by modeling that trajectory's velocity histograms $H_0 = f\{h_1,h_2, ...,h_{wo}\}$. We model each trajectory's motion by a multi-variate Gaussian distribution. For simplicity, it is assumed that the HOGHOFs are uncorrelated between different time instances i ≠ j, giving:

$$C_0(i,j)=\begin{cases} 0 & i\neq j \\ \sigma_{i,j}^2 & i=j \end{cases}$$, i.e. zero cross-correlation for different trajectories and auto-correlation values equal to the square of the variance. The initial pdf is then given by:

$$f_0(h_i)=\frac{1}{(2\pi)^{N/2}\,|C_0|^{1/2}}\exp\left(-\frac{1}{2}(h_i-\mu_0)^T C_0^{-1}(h_i-\mu_0)\right)$$

This initial distribution is compared at each frame k with the "current" one, estimated from the most recent w_0 set of histograms $H_1=\{h_{k-w0-1},h_{k-w0-2},..h_k\}$, including the latest one, and is given by:

$$f_1(h_i)=\frac{1}{(2\pi)^{N/2}\,|C_1|^{1/2}}\exp\left(-\frac{1}{2}(h_i-\mu_1)^T C_1^{-1}(h_i-\mu_1)\right)$$

$$\mu_1=\frac{1}{W_0}\sum_{i=k-W_0+1}^{k} h_i$$

$$C_1(i,j)=E\left[(h_i-\mu_1)^T(h_i-\mu_1)\right]=\frac{1}{W_0}\sum_{i=k-W_0+1}^{k}(h_i-\mu_1)^T(h_i-\mu_1)$$

$$C_1(i,j)=\begin{cases} 0 & i\neq j \\ \sigma_{i,j}^2 & i=j \end{cases}$$, as before.

The algorithm is then required to determine whether a change occurs or not at the current frame based on the log-likelihood ratio of the trajectory velocity distributions [15], which is used as a test statistic T_k, given by:

$$T_k=\log\left(\frac{f_1(h_k)}{f_0(h_k)}\right)$$

CUSUM can then be applied in the computationally efficient iterative form introduced by Page in [15]:

$$S_k=\max\left(0,S_{k-1}+T_k\right),S_0=0$$

For Gaussian data under each hypothesis H_0 and H_1, the test statistic, i.e. the log-likelihood ratio, becomes:

$$T_k=\frac{1}{2}\ln\left(\frac{|C_0|}{|C_1|}\right)+\frac{1}{2}(h_k-\mu_0)^T C_0^{-1}(h_k-\mu_0)$$
$$-(h_k-\mu_1)^T C_1^{-1}(h_k-\mu_1)$$

with diagonal covariance given by:

$$C_i=diag\left(\sigma_{i,1}^2,\sigma_{i,2}^2,...,\sigma_{i,N}^2\right),i=\{0,1\}$$
$$\sigma_{i,k}^2=E\left[(h_i-\mu_1)^T(h_i-\mu_1)\right],k=[1,2,...,N]$$

The inverse of each diagonal matrix is given by:

$$C_i^{-1}=diag\left(1/\sigma_{i,1}^2,1/\sigma_{i,2}^2,...,1/\sigma_{i,N}^2\right),i=\{0,1\}$$

so its determinant is given by:

$$|C_i|=\prod_{j=1}^{N}\sigma_{i,j}^2,i=\{0,1\}$$

Thus, by plugging in T_k calculated at each frame, we get a value for the test statistic S_k which significantly increases when there is a change in our data, i.e. the motion features. At each frame, S_k is then compared against an empirically derived threshold, which is chosen to lead to the lowest number of false alarms [17] and, when it surpasses it, a change is detected. This leads to the temporal segmentation of the extracted trajectories based on *actual changes in the activities taking place*, rather than their segmentation using a manually selected constant threshold (e.g. setting all trajectories to 15 frames long), which is often the case in the current literature.

III. BAG OF FEATURES

The bag-of-features pipeline is one of the most widely used methods for representing a video sequence, as well as images, as a normalized frequency histogram of local space-time features. The central idea of this concept is to partition the local descriptor space into informative spaces and then associate each video sequence to the obtained cluster centers of these regions by quantizing each local descriptor. We examine several variations of this method, using State of the Art encodings from video analysis and object recognition to improve upon existing activity recognition algorithms.

A. Vocabulary Construction

K-means clustering is the most common technique used to construct the desired visual vocabulary [2, 3, 6, 7, 12]. Given a set $\{x_1,...,x_N\} \in R^D$ of N training descriptors, k-means seeks K Cluster Center vectors $CC_1,\ldots,CC_K \in R^D$ that can partition the local descriptor space in the best possible way.

The most commonly used implementation of this technique is proposed in [16]. Recently another, more sophisticated technique was proposed in [9], based on hierarchical tree structures. It is very useful for describing large scale datasets, as it is much faster than traditionally used k-means, as it uses an approximate nearest neighbours, based on hierarchical tree structures, without losing its accuracy. In our work we test both K-means and hierarchical K-means, to detect changes in their recognition accuracy and computational speed.

GMM (Gaussian Mixture Model) clustering, on the other hand, has been successfully used for image recognition and retrieval purposes and has produced very promising results [12, 18]. This procedure is more sophisticated, as it models the data with a parametric probability density $p(x|\theta)$ given by :

$$p(x \mid \theta) = \sum_{k=1}^{K} p(x \mid \mu_k, \Sigma_k)\pi_k,$$

$$p(x \mid \mu_k, \Sigma_k) = \frac{1}{\sqrt{(2\pi)^D |\Sigma_k|}} e^{-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k)},$$

where $\theta=(\pi_1,\mu_1,\Sigma_1,\ldots, \pi_k,\mu_k,\Sigma_k)$ is the vector of parameters of the model, including the prior probability values $\pi_k \in R_+$ (which sum to one), the means $\mu_k \in R^D$, and the positive definite covariance matrices $\Sigma_k \in R^{DxD}$ of each Gaussian component. Expectation maximization (EM) is applied on a training set of descriptors $\{x_1,\ldots,x_N\} \in R^D$ in order to learn the data parameters and provide a set of Cluster Centers for soft data-to-cluster assignment, given by:

$$CC_{ki} = \frac{p(x_i \mid \mu_k, \sigma_k)\pi_k}{\sum_{j=1}^{K} p(x_i \mid \mu_j, \sigma_j)\pi_j}, k = 1,\ldots,K$$

B. Visual Encodings and Kernels

Hard binning encoding followed by **non-linear Kernels** is the most common practice that follows k-means clustering for training an SVM classifier. The most famous non-linear Kernel that follows binning belongs to additive homogeneous kernels and is called χ^2 (Chi square) kernel, given by:

$$K(h_i, h_j) = sign(f, g)\sum_{i=1}^{D} \frac{2 \mid h_i \mid \mid h_j \mid}{(\mid h_i \mid + \mid h_j \mid)}$$

The resulting vectors are then l_2-normalized.

Inverted documents are usually used in approximate nearest neighbor, for fast acquisition and efficient memory management. Term frequency – inverted document frequency (**TF-IDF**) is usually the metric that is used on these structures, following normalized **hard binning**. A **non-linear Kernel** is then constructed by computing similarity scores among video sequences. The similarity score between two video sequences q, d is computed by using the L_p norm metric:

$$Dist(q,d) = |q-d|^p = \sum_i |q_i - d_i|^p$$

The non-linear Kernel is then given by:

$$K = \exp(-\gamma Dist(q,d))$$

The last technique examined for encoding a video sequence is **Fisher encoding**, after building a GMM vocabulary. Fisher encoding has been used very successfully for object recognition leading to very accurate state of the art results, and is expected to also lead to accurate video recognition. Our experiments in Section IV show that, indeed, this is the case for video activity recognition.

The first step of this procedure computes the first and second order differences between the action descriptors and the centres of GMM. For instance, given a set of descriptors $x_1,\ldots,x_N$ sampled from a video sample, let $q_{k,i}$, k =1,…,K, i=1,…,N be the soft assignments of the N descriptors to the K Gaussian components. For each k = 1,…,K, we then define the vectors:

$$u_k = \frac{1}{N\sqrt{\pi_k}} \sum_{i=1}^{N} q_{ik} \sum_{k}^{-1/2} (x_t - \mu_k),$$

$$v_k = \frac{1}{N\sqrt{2\pi_k}} \sum_{i=1}^{N} q_{ik}[(x_t - \mu_k)\sigma_k^{-1}(x_t - \mu_k) - 1]$$

The Fisher encoding of the set of local descriptors is then given by the concatenation of u_k and v_k for all K components, giving an encoding of size 2DK:

$$f_{Fisher} = [u_1^T, v_1^T, \ldots, u_K^T, v_K^T]^T$$

This procedure is performed for several spatial pyramids in order to provide geometry information to our encoding and boost the accuracy. Each vector is normalized using a Hellinger kernel:

$$K(h_i, h_j) = sign(f, g)\sum_{i=1}^{D} \sqrt{h_i h_j}$$

The resulting vectors are then accumulated in a single one, which is l_2-normalized for best results. Finally, in all the experiments, a linear SVM is used on top of each encoding for activity recognition, as is the case in the current literature.

IV. EXPERIMENTS

Our experiments are first conducted on videos of activities of daily living, namely the University of Rochester ADL videos (URADL: http://www.cs.rochester.edu/~rmessing/uradl/). The names of the activities of this dataset are encoded in the tables as AP = Anser Phone, CB = Chop Banana, ES = Eat Snack, DP = Dial Phone, DW = Drink Water, EB = Eat Banana, LiP = Look up in Phonebook, PB = Peel Banana, US = Use Silverware, WoW = Write on Whiteboard. The activities were performed three times by five different people resulting in 150 video sequences. Our evaluation consisted of training on all repetitions of activities by four of the five subjects, and testing on all repetitions of the fifth subject's activities. This leave-one-out testing was averaged over the performance with each leftout subject.

In addition to URADL, a new, realistic dataset of people carrying out activities of daily life in a home-like environment was also created, within the European Project Dem@Care (Dementia Ambient Care: Multi-Sensing Monitoring for Intelligent Remote Management and Decision Support, http://www.demcare.eu/), which aims at supporting elderly people with dementia and promoting their independence at home, by monitoring their daily living and detecting behavior and lifestyle patterns, as well as deviations from them (which may indicate an improvement or worsening of their condition).

Video recordings of activities of daily living, conducted by people diagnosed with mild dementia to Alzheimer's, as well as healthy individuals, at the Greek Association for Alzheimer's and Associated Disorders[1]. 32 individuals were recorded from each group, performing eleven activities of daily life. 21 patients were used for training our recognition model, providing 230 video sequences and 11 subjects of the patient population were used for testing purposes, providing 118 video sequences. This resulted in a much more challenging and realistic dataset than the preceding one, due to its subject variability and the unconstrained environment.

Fig 4. *Characteristic samples from the Dem@Care dataset with ADL, namely: Drink Beverage, Enter Room, Handshake, Read Paper, Start Phonecall, Talk to Visitor.*

A. URADL Dataset

[1] http://www.alzheimer-hellas.gr/english.php

The activities of DemCare dataset are encoded in the tables as CU = Clean Up table, DB = Drink Beverage, EP = End Phonecall, ER = Enter Room, ES = Eat Snack, HS = HandShake, PS = Prepare Snack, RP = Read Paper, SB = Serve Beverage, SP = Start Phonecall, TV = Talk to Visitor. The faces of the individuals shown in Fig. 4 are not visible for privacy protection reasons. All individuals, both health volunteers and people with dementia (or their guardians) have signed the appropriate ethical consent forms before the recordings took place.

Experiments took place to determine which bag-of-features and encoding schema perform best with our improved descriptor. Tables 1 – 3 below show the results of k-means with χ^2, hierarchical k-means with L_p non-linear kernel and GMM – fisher vector. The best results were acquired with a GMM vocabulary combined with Fisher encoding.

Table1. *Activity recognition with K-means vocabulary (K=4000 CC) and χ^2 kernel.*

k-means 4000 CC, chi square	HOGHOF RANSAC									
	AP	CB	DP	DW	EB	ES	LiP	PB	US	WoW
AP	**0.47**		0.4		0.13					
CB		**0.93**	0.07							
DP	0.07		**0.93**							
DW				**1**						
EB			0.07		**0.8**	0.07		0.07		
ES			0.07		0.07	**0.8**		0.07		
LIP							**1**			
PB								**1**		
US									**1**	
WOW										**1**
Average Accuracy 0.893										

Table2. *Activity recognition with HK-means vocab., L_p kernel.*

Hk-means 9^4	HOGHOF RANSAC									
	AP	CB	DP	DW	EB	ES	LiP	PB	US	WoW
AP	**0.27**		0.33	0.2	0.2					
CB		**0.93**	0.07							
DP	0.2		**0.67**	0.07				0.07		
DW				**1**						
EB					**1**					
ES						**1**				
LIP							**1**			
PB				0.07				**0.93**		
US									**1**	
WOW										**1**
Average Accuracy 0.887										

Table3. *Activity recognition with GMM vocabulary and Fisher vector.*

Hellinger L2 norm.	HOGHOF 256CC RanSac, (13 Spat Pyramids)									
	AP	CB	DP	DW	EB	ES	LiP	PB	US	WoW
AP	**0.47**		0.47		0.07					
CB		**0.93**						0.07		
DP	0.13		**0.87**							
DW				**1**						
EB			0.13		**0.87**					
ES						**1**				
LIP							**1**			
PB								**1**		
US									**1**	
WOW										**1**
Average Accuracy 0.913										

Based on the above observations from the results of Tables 1-3, we chose to use the GMM – fisher vector encoding in our next experiments. Experiments, whose results are shown in Table 4, were conducted in order to determine whether the RANSAC outlier elimination will boost the performance of our algorithm. Indeed, an obvious drop on the performance was observed when we omitted RANSAC estimator, with average accuracy falling to 86.7% from 91.3%.

Table 4. *Activity recognition without RANSAC estimator*

Hellinger L2 norm.	HOGHOF 256CC No RANSAC									
	AP	CB	DP	DW	EB	ES	LiP	PB	US	WoW
AP	**0.47**		0.4		0.13					
CB	0.2	**0.73**						0.07		
DP			**0.93**					0.07		
DW				**1**						
EB			0.13		**0.67**			0.2		
ES						**1**				
LIP							**1**			
PB		0.07						**0.93**		
US				0.07					**0.93**	
WOW										**1**
Average Accuracy 0.867										

Finally, we compare our results on the URADL dataset with the state of the art (SoA), activity recognition algorithm of [3] and the work of the URADL authors [11]. Table 5 shows that our method's accuracy surpasses that of URADL and is comparable with [3].

The results vary depending on the encoding used for the vocabulary construction. It should be noted that [3] only uses Kmeans and the chi-square kernel, whereas we extend their method by also testing out GMM-Fisher encoding, shown in Table 5. The speed of our approach is much higher than that of [3], mostly due to the use of MBAAs which reduce the size of the HD data being processed.

Table5. *Average accuracy surpasses/is comparable to the SoA. Fisher boost is observed when applied to [3].*

	Our	[11]	[3]Kmeans χ^2 kernel	[3]GMM Fisher
Av. Acc	91.33%	89.33%	92%	92.67 %

For URADL, Wang's [3] method needed 23 hrs, 1 min and 15 sec for activity recognition, while our method leads to a lower computation time of 15 hrs, 50 min and 23 sec. This is mainly because the approach of [3] processes the entire high definition video frame, while we extract descriptors on the MBAAs, which are significantly smaller than the entire frame.

B. Dem@Care Dataset

It was shown that GMM with the Fisher vector provided the best results on the URADL data, applied to our method and to [3]. Thus, we only examine GMM with Fisher for the Dem@Care dataset. Tables 6 and 7 show that our algorithm now *surpasses [3] by around 3.37%*, although the Dem@Care videos are more challenging than URADL.

Dem@Care videos contain more complex activities, in less constrained environments, with a large variety of subjects, occlusions and behaviours. Their difficulty further compounded by the fact that many of the volunteers suffer from dementia, ranging from mild to Alzheimer's disease (in most cases the subjects were accompanied to the site of the recordings), which affects their performance in carrying out ADLs. Also, the Dem@Care videos involve walking around a room, behind tables/chairs, similarly to a real home environment, whereas in URADL the actors are for the most part standing in the same spot.

Table 6. *Activity recognition results of our algorithm on the Dem@Care videos*

HOGHOF with compact spatial pyramid											
	CU	**DB**	**EP**	**ER**	**ES**	**HS**	**PS**	**RP**	**SB**	**SP**	**TV**
CU	0.69	0.08			0.08					0.15	
DB		1									
EP			0.88							0.13	
ER				1							
ES		0.57			0.38			0.05			
HS						1					
PS		0.23					0.31		0.46		
RP								1			
SB									1		
SP										1	
TV											1
Average Accuracy 0.8414											

Table 7. *Activity recognition of Wang's method [3] applied to the Dem@Care dataset.*

HOGHOF Wang											
	CU	**DB**	**EP**	**ER**	**ES**	**HS**	**PS**	**RP**	**SB**	**SP**	**TV**
CU	0.69		0.23							0.08	
DB		0.79	0.05		0.16						
EP			1								
ER				1							
ES		0.43		0.29	0.24			0.05			
HS						1					
PS		0.08			0.08		0.31		0.54		
RP								1			
SB									1		
SP										1	
TV				0.14							0.86
Average Accuracy 0.807											

V. CONCLUSIONS

We present a new method for detecting activities of daily living, which will prove particularly useful when monitoring people's daily life, behavioural patterns, lifestyle, as is the case in assisted living situations, in smart homes for example. We introduce MBAAs to localize points of interest, where the velocity changes, so as to reduce the computational cost of the method by working in much smaller regions of each video frame. Trajectories are temporally segmented in a near optimal manner by applying CUSUM to detect changes in velocity and thus separate the trajectories in a meaningful manner. Dense spatiotemporal features on several scales are employed in order to have rich descriptive information about the data and also be scale invariant. Finally, the best encoding methods are tested in various combinations in order to find the most accurate, yet quick method for recognizing activities. Results on benchmark datasets and a more challenging and very realistic dataset of people expected to benefit from assisted living solutions prove that the proposed method is, in the first case comparable with the SoA, and with the latest dataset gives more accurate results at a lower computational cost.

VI. ACKNOWLEDGEMENTS

This work is funded by the European Commission's 7th Framework Program (FP7 2007-2013), under grant agreement 288199 Dem@Care.

VII. REFERENCES

[1] Jingen Liu, Jiebo Luo and Mubarak Shah, "Recognizing Realistic Actions from Videos "in the Wild"", IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2009.

[2] M. Marszałek, I. Laptev and C. Schmid , "Actions in Context", IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2009.

[3] Heng Wang; Alexander Kläser; Cordelia Schmid; Liu Cheng-Lin, "Action Recognition by Dense Trajectories", *IEEE Conference on Computer Vision & Pattern Recognition*, (CVPR) Jun 2011, Colorado Springs, United States. pp. 3169-3176.

[4] Jean-Yves Bouguet. Pyramidal Implementation of the Lucas Kanade Feature Tracker.

[5] A. Briassouli, I. Kompatsiaris, "Robust Temporal Activity Templates Using Higher Order Statistics", IEEE Transactions on Image Processing, Vol. 18, Issue 12, pp. 2756-2768, December 2009.

[6] Paul Scovanner, Saad Ali, and Mubarak Shah, "A 3-Dimensional SIFT Descriptor and its Application to Action Recognition", ACM MM 2007.

[7] Alexander Klaser; Marcin Marsza?ek; Cordelia Schmid, Mark Everingham and Chris Needham and Roberto Fraile, "A Spatio-Temporal Descriptor Based on 3D-Gradients", 19th British Machine Vision Conference, Sep BMVC 2008, Leeds, United Kingdom. British Machine Vision Association, pp. 275:1-10.

[8] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman, "The devil is in the details: an evaluation of recent feature encoding methods", in Proceedings of the British Machine Vision Conference (BMVC), 2011.

[9] D. Nistér and H. Stewénius. "Scalable recognition with a vocabulary tree". In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 2161-2168, June 2006.

[10] I. Laptev and T. Lindeberg, "Space-Time Interest Points", in International Conference in Computer Vision (ICCV), 2003, Nice, France, pp.I:432-439.

[11] Messing, R., Pal, C. & Kautz, H.,"Activity recognition using the velocity histories of tracked keypoints", in International Conference in Computer Vision (*ICCV) 2009.*

[12] Ivan Laptev, Marcin Marszałek, Cordelia Schmid and Benjamin Rozenfeld, "Learning Realistic Human Actions from Movies", in *Computer Vision and Pattern Recognition. CVPR'08*

[13] I. V. Blagouchine and E. Moreau, "Unbiased efficient estimator of the fourth-order cumulant for random zero-mean non-i.i.d. signals: Particular case of ma stochastic process," IEEE Transactions on Information Theory, vol. 56, no. 12, pp. 6450 – 6458, 2010.

[14] V. P. Dragalin, "Optimality of a generalized cusum procedure in quickest detection problem," in Statistics and Control of Random Processes: Proceedings of the Steklov Institute of Mathematics, Providence, Rhode Island, 1994, pp. 107–120.

[15] E. S. Page, "Continuous inspection scheme," Biometrika, vol. 41, pp. 100–115, 1954.

[16] S. Lloyd. Least square quantization in PCM. IEEE Trans. on Information Theory, 28 (2), 1982.

[17] G. V. Moustakides. Optimal stopping times for detecting changes in distributions. Ann. Statist., 14:1379, 1986.

[18] Hervé Jégou, Florent Perronnin, Matthijs Douze, Jorge Sanchez, Patrick Pérez and Cordelia Schmid, "Aggregating local images descriptors into compact codes", IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), September 2012.

[19] L Gorelick,M.Blank,E.Shechtman,M.Irani,and R.Basri,"Actions as space-time shapes," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 12, pp. 2247–2253, December 2007.