D4.2

Activities Monitoring & Lifelogging v1

Dementia Ambient Care: Multi-Sensing Monitoring for Intelligent Remote Management and Decision Support

Dem@Care - FP7-288199

Deliverable Information

Project Ref. No.		FP7-288199
Project Acronym		Dem@Care
Project Full Title		Dementia Ambient Care: Multi-Sensing Monitoring for Intelligence Remote Management and Decision Support
Dissemination level:		Public
Contractual date of delivery:		Month 18, 30/04/2013
Actual date of delivery:		Month 19, 14/05/2013
Deliverable No.		D4.2
Deliverable Title		Activities Monitoring & Lifelogging v1
Type:		Report
Approval Status:		Final
Version:		1.5
Number of pages:		118
WP:		WP4 Situational Analysis of Daily Activities
Task:		T4.1 Visual Perception, T4.2 Audio Sensing. T4.3. Instrumental Activities Monitoring, T4.4. Life-logging
WP/Task responsible:		WP4/T4.1, T4.3
Other contributors:		IBM, INRIA, CERTH, DCU
Authors (Partner)		UB1: Vincent Buso, Rémi Mégret, Guillaume Bourmaud CERTH: Konstantinos Avgerinakis, Alexia Briasouli DCU: Eamonn, Newman, Aiden Doherty, Feiyan Hu LTU: Basel Kikhia INRIA: Francois-Bremond, Piotr Bilinsky, Baptiste Fosty, Carlos-Fernando Crispim_junior, Anh-Tuan Nghiem.
Responsible Author	Name	Francois Bremond
	Email	Francois.Bremond@inria.fr
Internal Reviewer(s)		Ceyhun Burak Akgül (VIV)
EC Project Officer		Gerald Cultot
Abstract (for dissemination)		This report presents the current research results of WP4 in Dem@care project. In Dem@care project, WP4 is responsible for analyzing low-level audio and visual sensoring data to provide useful information for behaviour interpretation components in WP5. In this report, we describe our advances concerning video-based posture recognition, action recognition, activity monitoring, and lifelogging relevant to dementia diagnosis and monitoring. All of them have either higher or comparable performance than the state-of-the art in the corresponding domain.

Version Log

Version	Date	Change	Author
0.1	05 March 2013	Originated	Francois-Bremond
0.2	14 March 2013	Added UB1 object recognition	Vincent Buso
0.3	05 April 2013	Added INRIA activity recognition	Francois-Bremond, Piotr Bilinsky, Baptiste Fosty, Carlos-Fernando Crispim_junior, Anh-Tuan Nghiem
0.4	08 April 2013	Added CERTH activity recognition	Konstantinos Avgerinakis, Alexia Briasouli
0.5	12-April-2013	UB1: Added results on CHU Nice dataset for Object recognition. Added pose filtering from wearable camera. Added filtering for posture analysis	Vincent Buso Rémi Mégret, Guillaume Bourmaud
0.7	18-April-2013	Update the contribution of UB1, LTU, CERTH	Francois-Bremond, Piotr Bilinsky, Baptiste Fosty, Carlos-Fernando Crispim_junior, Anh-Tuan Nghiem
0.8	19-April-2013	Update intro and conclusion (UB1)	Rémi Mégret
0.9	19-April-2013	Update intro and conclusion (INRIA)	Anh-Tuan Nghiem
1	22-April-2013	Finish intro and conclusion (INRIA)	Anh-Tuan Nghiem, Carlos-Fernando Crispim_junior
1.1	23-April-2013	Adding some text to section 5.1 to clarify the term "Event", which is used in the lifelogging section (LTU)	
1.2	26-April-2013	Correction of Ceyhun Burak Akgül	Ceyhun Burak Akgül
1.2-UB1	29-April-2013	Corrected sections 2.1, 3.1, 4.1 following internal review (UB1)	Rémi Mégret
1.2-DCU	29-April-2013	Updates to Section 5 following review(LTU)	
1.3	2-May-2013	Integration of all modification (INRIA)	Anh-Tuan Nghiem, Carlos-Fernando Crispim_junior
1.4	9-May-2013	Formatting	Athina Kokonozi (CERTH)
1.5	13-May-2013	Formating(INRIA)	Anh-Tuan Nghiem

Executive Summary

The objective of Dem@Care is to develop a system providing personal health services to people with dementia, as well as medical professionals and caregivers. This system will use a variety of sensors to monitor patients lifestyle, ambient environment and health parameters. In this context WP4 aims to analyse daily activities of the dementia patient in their domestic environment. Specifically, WP4 components first capture visual and audio data using various types of wearable and static sensors. Then from this data, WP 4 extracts useful information concerning behavioural patterns as well as abnormality to monitor the progress of dementia disease and to assist in planning / implementing the therapy.

The present deliverable D4.2. reports the results of the algorithms proposed for processing visual data in order to perform posture recognition, action recognition, activity monitoring, and life-logging.

For posture recognition, the deliverable reports the work of UB1 on an algorithmic approach to filter the pose (position and orientation) information obtained from the wearable camera with the final goal of providing accurate data for posture inference.

For action recognition, the deliverable covers the contribution of three partners UB1, CERTH, and INRIA. UB1 present a method to analyze wearable camera video stream with respect to the visual saliency associated to actions of the person doing them, as well as to the way an observer watching them. Visual saliency models aim at estimating where the action of interest is going on in the video stream, which is helpful in better designing object detection algorithms. CERTH presents a new method of human action recognition using static camera which is faster than most of the state of the art methods. INRIA also proposes a new method of human action recognition using a dynamic coordination system, which makes the proposed method outperform existing methods for human action recognition on popular datasets.

For activity monitoring, the deliverable summarizes the contributions of UB1 and INRIA. UB1 presents a fast object recognition method based on visual attention maps using GoPro camera. With 3D scene model, recognised objects become the input to recognise human trajectories, postures, and analyse human activities. INRIA proposes a description based approach for activity recognition using RGB-D camera. Their experiments show that, compared with normal colour camera, the RGB-D camera has several advantages such as ease of installation and improvement of activity recognition.

For lifelogging, the deliverable reports the work of DCU/LTU on building a lifelog of daily activities of dementia patients. Specifically, it explains how life-logging technology will be used within the Dem@Care project to reason about the day of the person. The deliverable also presents an algorithm to aggregate information from different sensors about the person's current activity. The aim of this algorithm is to reason and return the activity with the highest belief using different indicators of different sensors. As a result, the day of the person will be organized as searchable and browsable lifelogs

Abbreviations and Acronyms

ADL	Activities of Daily Living
AUC	Area Under Curve
BoW	Bag-of-Words
B-BoW	Basic BoW
I-BoW	BoW with Ideal Masks
BPAF	Basic Probability Assignment Function
CMST	Combined Multi-Scale Tracklet
DAG	Directed Acyclic Graph
D-EKF	Discrete Extended Kalman Filter
D-LG-EKF	Discrete Extended Kalman Filter on Lie groups
DPM	Discriminatively Trained Part-Based Model
DST	Dempster-Shafer Theory
ET	Evidence theory
FGS	Fitting Gaussian Surface
HOG	Histogram of Oriented Gradient
HOF	Histogram of Oriented Flow
LPF	Low-Pass Filtering
LOOCV	Leave-One-Out Cross-Validation
mAP	mean Average Precision
MBAA	Motion Boundary Activity Areas
MBH	Motion Binary Histograms
MPEG	Moving Picture Experts Group
OWL	Ontology Web Language
OWL-QL	Ontology Web Language Query Language
OWL-DL	Ontology Web Language Description Language
PCC	Pearson correlation coefficient
PnP	Perspective-n-Point
PwD	Person with dementia
RDF	Resource Definition Framework
ROC	Receiver operating characteristic
ROI	Region of Interest
RMSE	Root Mean Square Error
RMST	Relative Multi-Scale Tracklet
SMST	Shape Multi-Scale Tracklet
SoA	State of Art
W3C	World Wide Web Consortium
XML	eXtensible Markup Language
URADL	University of Rochester Activity of Daily Living dataset

Table of Contents

List of Figures

List of Tables

1 Introduction

The objective of Dem@Care is to develop a system providing personal health services to people with dementia, as well as medical professionals and caregivers. This system will use a variety of sensors to monitor patients lifestyle, ambient environment and health parameters. In this context WP4 aims to analyse daily activities of the people with dementia in their domestic environment. To do this, WP 4 first capture visual and audio data using various types of wearable and static sensors. Then from this data, WP 4 will extract useful information concerning behavioural patterns as well as anomaly to monitor the progress of dementia disease and to assis in planing / implementing the therapy.

The deliverable D4.2. presents the results of proposed algorithms for processing visual data to perform posture recognition, action recognition, activity monitoring, and lifelogging. The deliverable is structured as follows.

Section 2 presents the preliminary works of WP 4 on posture recognition using wearable camera. In section 2.1, an algorithmic approach is introduced to filter the pose (position and orientation) information obtained from the wearable camera with the final goal of providing accurate data for posture inference.

Section 3 presents the results of WP 4 on action recognition. In Section 3.1, wearable camera video stream is analyzed with respect to the visual saliency associated to actions of the person doing them, as well as to the way an observer watching them. Visual saliency models aim at estimating where the action of interest is going on in the video stream, which is helpful in better designing object detection algorithms in Section 4.1. In Section 3.2, a new method of human action recognition using static camera is introduced. Compared with the state of the art, this algorithm is highly accurate but at a lower computational cost. In Section 3.3, another algorithm for human action recognition is introduced. By using a dynamic coordinate system, this algorithm outperforms the accuracy of existing methods in action recognition.

Section 4 presents the results of WP 4 on activity monitoring. In Section 4.1, object recognition from wearable camera video stream is addressed, since objects are strongly correlated with the activities of the person. State of the art object recognition is extended with the visual saliency models to better differentiate between relevant and irrelevant parts of the images, and therefore improve of the recognition performances. In Section 4.2, activity monitoring using RGB-D is introduced. In this section, we present a set of people detection and tracking techniques as well as an evaluation framework for mid to long-term event activity recognition using hierarchical model-based approach combined with a RGB-D camera.

Section 5 presents the results of WP 4 on lifelogging. Lifelogging means a database which is is searchable and browsable using common daily activities as queries. This section explains how life-logging technology will be used within the Dem@Care project to reason about the day of the person.

2 Posture recognition

2.1 Pose estimation from wearable camera for inside-out location and posture information inference

2.1.1 Introduction

In this section, we are interested in developing tools to support the estimation of the pose parameters from the image content of the camera worn by the person (GoPro component in the Dem@Care system, see D7.1 "System Specifications & Architecture V1"). This information then provides an inside-out point-of-view of the posture state, which complements the observation from outside using static sensors and the motion sensors. In particular, in the context of activities monitoring, the inside-out view shows the instrumental space that is in front of the person (the PwD in our case), which contains manipulated objects, the part of the environment that is the current focus of the person and in which the person is positioning himself/herself.

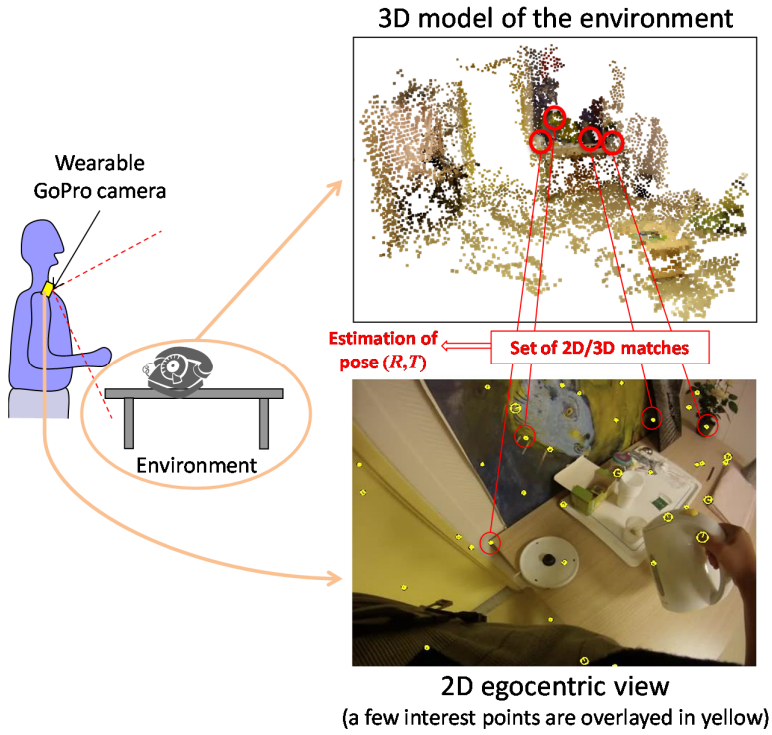

Figure 2.1-1 Illustration of the principle of 3D pose estimation using the GoPro camera.

Figure 2.1-1 illustrates the principle of 3D pose estimation using the GoPro camera. In this figure, image on the left shows the GoPro worn by the person capturing the environment. Image on the top Right shows the 3D model of the phone and tea places in CHUN @Lab room. Image on the bottom Right shows 2D image captured by the GoPro during the activity "preparing tea", with some automatically detected keypoints overlayed; The automatic

matching between 2D image keypoints and 3D points of the model provides information used by the Perspective-n-Point to estimate the pose (R,T).

If a visual model of this 3D environment is known, then it is possible to estimate the 3D pose trajectory (time varying position and orientation) of the person by matching the 2D image content and the 3D model as discussed in D4.1 "Audio & Visual Sensing v1", and as illustrated in figure 2.1-1. The estimated 3D pose provides information that is relevant to later infer the posture of the person within the 3D environment, such as:

The precise location with respect to places of interest,

The orientation of the person with respect to these places (is the person engaged towards this place or simply standing around?),

The height, providing information about whether the person is standing, bending, crouching or sitting,

The motion, such as the lateral motion, which informs if the activity is focused in one place or not.

To represent the camera pose, the state corresponds to the couple (R, T) where R represents the pose (3D orientation) and T represents the 3D position within the environment. This corresponds to 6 degrees of freedom. In order to take into account the natural smoothness of the trajectory, a constant velocity model is suitable and can be represented by the rotational motion vector ω and translational motion vector v.

Estimating the trajectory on-line corresponds to a non-linear filtering problem, with the difficulty that the rotational component does not belong to the Euclidean space, but to a more generic manifold, called Lie Group. Typical examples of Lie groups include rotation matrices SO (3), unitary quaternions SU(2), rigid-body motion SE(3), homographies SL(3), invertible matrices GL(3), etc. In order to take this into account in a unified way, we detail a new filtering framework. It can be tailored to specific applications by designing the Lie Groups to which the state and the observations belong.

This framework covers the GoPro pose filtering problem we face within Dem@Care. Indeed an instantaneous estimates of the pose can be obtained from image to model matching using a state-of-the-art Perspective-n-Point (PnP) estimator such as [.2.17]. This algorithm produces estimates of the (R,T) pose, which belongs to a Lie Group. Because of the poor observability of some degrees of freedom, these visual estimates have to be filtered to obtain stable parameters, therefore requiring a filter that takes Lie Group observations and estimate a Lie Group state. For this problem, standard filtering approaches such as the Discrete EKF [2.1.4] do not apply, as they are designed for states and observations in an Euclidean space

We introduce hereafter a generic framework called Discrete Extended Kalman Filter on Lie groups (D-LG-EKF) that solves this limitation, and can therefore be applied to the GoPro pose estimation problem.

The rest of this section is organized as follows: after introducing the formulation of pose estimation as a filtering problem on Lie groups, we will provide an overview of the D-LG-EKF approach and discuss how it relates to previous work. The proposed approach will be compared to state of the art approaches on synthetic data chosen to be representative of the positioning problem within a room, which is one of the target applications within Dem@Care.

2.1.2 Discrete Extended Kalman Filter on Lie Groups

The target problem deals with estimating the camera position $T \in \mathbb{R}^3$ and orientation $R \in SO(3)$. Both the angular velocity $\omega \in \mathbb{R}^3$ and the translational velocity $v \in \mathbb{R}^3$ are also estimated, in order to effectively smooth the trajectory using a constant speed model. We assume R and T are directly observed, such as when considering as input the result of the pose estimation algorithm applied on each frame of the video [2.1.17]. This setup corresponds to the Dem@Care setup, where the wearable camera can capture absolute location in each frame, but the velocities have to be inferred from that instantaneous information. This problem contrasts with what can be found in the literature of orientation filtering (such as [4.1.6], [4.1.7], [4.1.8], [4.1.9]) where the speed can be observed, which facilitates the derivation of a filter.

For modelling this problem, we use the following Lie groups:

$G = SO(3) \times \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^3$ represents the state $X_k=(R, \omega, T, v)$ to be estimated

$G' = SO(3) \times \mathbb{R}^3$ represents the observation $z_k=(R', T')$ resulting from frame based pose estimation algorithm.

In both cases, the classical basis of SO(3) is used [2.1.15]. We now describe how to represent the evolution and observation equations of the model:

The detailed description of the Lie group/Lie algebra theory, which is quite technical, is presented in annex 2.1.5. We report now how these mathematical tools are used to solve the problem of filtering GoPro pose estimates, and invite the interested reader to proceed to the annex for more technical explanations of the concepts used.

a) System model for the GoPro camera pose estimation problem

Let the system state be modeled as satisfying the following equation:

$$\begin{aligned} X_k &= f(X_{k-1}, u_{k-1}, n_{k-1}) \\ &= X_{k-1}\, exp_G\left([\Omega(X_{k-1}, u_{k-1}) + n_{k-1}]^{\wedge}_G\right) \end{aligned} \quad (1)$$

where $X_k \in G$ is the state we wish to estimate at time k and G is a p-dimensional Lie group. $u_{k-1} \in \mathbb{R}^w$ corresponds to a control input (not used in the current problem, but could be used to include inertial data) and $n_{k-1} \sim \mathcal{N}_R(0p\times 1, Rk\text{-}1)$ is a white Gaussian noise that represents the uncertainty about the temporal evolution of the motion.

In this equation, X_{k-1} is composed to the right with an increment that belongs to the Lie Group. The function exp_G represent the exponential map that maps an increment expressed in the Lie algebra onto the Lie group. Both composition and exp-map are key elements of the proposed framework to guarantees that the state remains a point within the group manifold and guarantee the consistency of the estimation.

The map $\Omega: G \times \mathbb{R}^w \rightarrow \mathbb{R}^p$ is a non-linear C^2 function that corresponds to the evolution model expressing at each frame the increment in the Lie algebra..

In the application to GoPro pose estimation, the model is a constant speed model, which fits the assumption of a smooth trajectory. It is expressed in a very straightforward way within our framework as:

$$\Omega\left(\begin{bmatrix} R_k \\ \omega_k \\ T_k \\ v_k \end{bmatrix}\right) = \begin{bmatrix} \omega_k \\ 0 \\ v_k \\ 0 \end{bmatrix} \quad (2)$$

We also consider discrete measurements on a q-dimensional Lie group G':

$$z_k = h(X_k)\, exp_{G'}\left([w_k]^{\wedge}_{G'}\right) \quad (3)$$

where $z_k \in G'$ and $w_k \sim \mathcal{N}_{(}(0q\times 1, Qk)$ is a white Gaussian noise. This modelling of the covariance is particularly useful to be able to include dependencies between the variables: orientation and translation estimates are correlated and their correlation needs to be taken into account for a consistent filtering.

In the application to GoPro pose estimation the observed measurements are the absolute rotation and translation obtained from the framewise pose estimation module. They are modeled as a function of the unknown state as:

$$h\left(\begin{bmatrix} R_k \\ \omega_k \\ T_k \\ v_k \end{bmatrix}\right) = \begin{bmatrix} R_k \\ T_k \end{bmatrix} \quad (4)$$

b) Overview of the proposed filtering solution

The filter takes as input the framewise pose estimates obtained using the EPnP pose estimation algorithm [2.1.17] along with their covariances and produces as output filtered pose estimates with an estimate of velocity and covariance.

We assume the state posterior distribution to be a concentrated Gaussian distribution (see annex) on Lie groups: $p(X_k|z_1 \ldots z_l) \sim \mathcal{N}_G(\mu_{k|l}, P_{k|l})$. Then, the D-LG-EKF solutions can be classically decomposed as two steps: propagation ($l = k-1$) and update ($l = k$). Therefore, the aim of the D-LG-EKF is to propagate and update the distribution parameters: average value on the Lie Group $\mu_{k-1|k-1}$ and covariance matrix in the Lie algebra $P_{k-1|k-1}$.

For the sake of clarity of the presentation, the technical details of the proposed solution can be found in annex 2.1.5. The precise implementation of the algorithm and its rationale are detailed there.

c) Discussion

Related work. Taking into account the geometry of a manifold usually leads to well-posed problems hence can boost the performance of an algorithm. A few works tried to extend

discrete Euclidean filtering algorithms to manifolds. For example, particle filters for states evolving on a Riemannian [2.1.1], Stiefel [2.1.2] or Grassmann [2.1.3] manifolds have been proposed. The approach we propose extends the Discrete Extended Kalman Filter (D-EKF) [2.1.4] defined for a state and measurements evolving on Euclidean spaces to the case of a state and measurements evolving on Lie groups. Following [2.1.5], this problem could be cast into a generic constrained filtering problem by enforcing an equality constraint taking the zero value only for matrices belonging to the group. However such algorithms take into account the geometry of the manifold in an extrinsic manner, which may cause the divergence of the filter, as will be shown in the experimental section.

A large amount of works modeling the state on a Lie group has dealt with the specific groups SO(3), SU(2) or SE (3). Among them [2.1.6] and [2.1.7] modified the unscented Kalman filter to estimate a unitary quaternion. In [2.1.8] an algorithm able to estimate the trajectory of a state evolving on SE (3) is described. In [2.1.9], an Invariant Momentum-tracking Kalman Filter is derived to estimate a unitary quaternion and an angular momentum vector. None of these specific approaches correspond to the problem faced for the GoPro pose estimation filtering with constant speed model and estimated absolute pose as observations.

Why not use a standard Euclidean D-EKF to solve this problem ? Estimating a state $X \in \mathrm{R}^{n \times n}$ while considering measurements $z \in G' \subset \mathbb{R}_{m \times m}$, where G and G' are Lie groups of dimension p and q respectively, is not coherent with the D-EKF theory which was developed to estimate states evolving on Euclidean spaces.

However, it is possible to adapt the constrained D-EKF formalism [2.1.9] in an ad hoc manner to fit to this problem, assuming $X \in \mathrm{R}^{n \times n}$, vectorizing it and considering the group geometry as a state constraint. Such an algorithm (denoted *D-EKF Constr* in the experiments) treats the geometry of the Lie group as an extrinsic constraint, thus the filtering is performed in the Euclidean embedding space R^r of the Lie group, where r>p. Consequently, both the state and the measurement covariance matrices are singular which causes issues during the Kalman gain computation, as shown in the experiments.

Another way to employ a D-EKF to solve the target problem is to consider a Lie algebra state $x = [log_G(X)]_G^{\vee}$ instead of the Lie group state X [2.1.14] and to consider also measurements transformed into the Lie algebra: $[log_{G'}(z)]_{G'}^{\vee}$. To apply such a filter, log_G and $log_{G'}$ must be defined over the whole group. This approach is denoted *D-EKF LieAlg* in the following, and is a suitable alternative to the D-LG-EKF as it does not produce singular covariance matrices. However, $log_{G'}$ may be discontinuous for some groups such as SO(3) which would yield the innovation to be incorrectly large even with a small error on the group.

2.1.3 Experimental evaluation

We evaluated the proposed formalism for the GoPro camera pose estimation problem on simulations designed to be representative of the problem encountered in Dem@Care.

Figure 2.1.-2 illustrates the gain of filtering the pose information compared to a simple framewise estimation. In this first experiment, we generate a smooth trajectory in a 3D model tht has been build from real data (kitchen place environment) and provide the observations both to a localization framework which does not take into account any smoothness a priori of the camera trajectory (EPnP refined using a Maximum Likelihood Estimator [2.1.17]) and to

our D-LG-EKF algorithm taking the EPnP pose estimate as input. On the figure, cones represent the camera pose estimated for each frame while the 3D ellipsoids correspond to the 3D position uncertainty. One can see that without the smoothness prior, the estimated trajectory is jittered and the uncertainties may be large. On the contrary, the D-LG-EKF provide a smoother estimated trajectory with reduced uncertainties, which is necessary to reduce jitter noise for posture analysis.

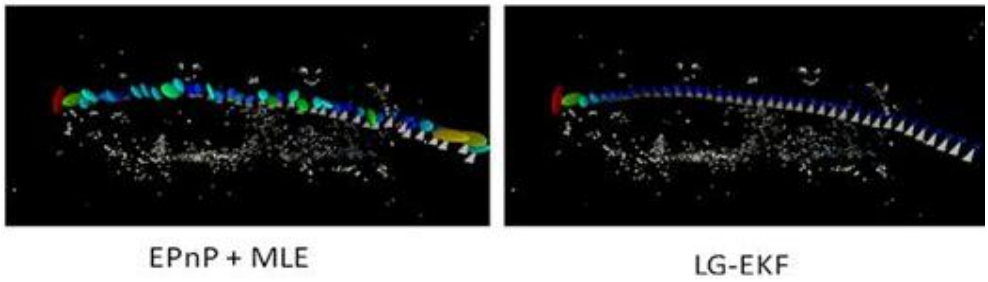

Figure 2.1-2: Example of the estimated trajectory

Figure 2.1-2 shows an example of the estimated trajectory. A PnP pose estimation and PnP filtered were applied with the D-LG-EKF to a simulated observation of a 3D point cloud constructed from real data of a kitchen (shown in gray levels in the background). The cones represent the estimated camera positions. The colored ellipsoids represent the covariance of each position estimate.

To obtain a more quantitative evaluation on a larger controlled data set, we have produced a synthetic dataset. We will assume for the sake of simplicity that the room is rectangular and the surfaces of the rectangular room will be sampled into a 3D point cloud. This model is illustrated in Figure 2.1-3. The camera is assumed to be calibrated. Then, trajectories are generated in this volume from which sequences of measurements are created using a maximum likelihood algorithm such as [2.1.16]. The covariance of each measurement is estimated by propagating the covariance from each 3D observed point. For all the filters, T and R are initialized at the correct position with small variances whereas ω and v are set to zero with large variances.

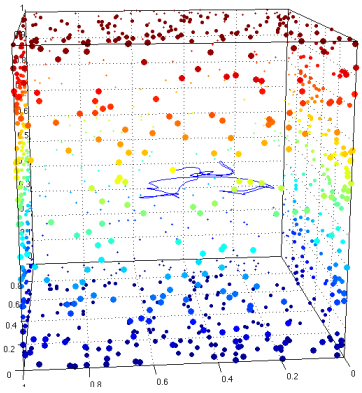

Figure 2.1-3: Example of 3D point cloud representing a rectangular room

Figure 2.1-3 shows a 3D point cloud representing a rectangular room, with visual landmarks on the sides, and one realization of the test trajectories to be estimated (only position is shown)

The D-LG-EKF was implemented following the models described previously. As a basis of comparison, we implemented two state-of-the-art filters: a D-EKF Constr and a D-EKF LieAlg discussed previously. All filters were given the output of the EPnP algorithm.

Figure 2.1-3: reports the RMSE (Root Mean Square Error) of each filter w.r.t sampling period δt. The RMSE is defined as the square root of the average of the following squared errors : $\|\mu - T\|_2$ (Euclidean position error) and $\left\|\log_{SO(3)}([\mu_R{}^T R]^{\vee}_{SO(3)}\right\|_2$ (orientation error as a Lie algebra residual).

As it was expected by the theoretical differences outlined previously, both the D-EKF Constr and the D-EKF LieAlg provide a very bad estimate (error larger than the input observations). In the case of D-EKF Constr, the more the sampling period δt grows, the more the state estimate is projected far from the true optimal state, which results in the incorrect estimates of the filter and numerical instabilities. For small δt, these effects are limited. In the case of D-EKF LieAlg, when the norm of the vector describing the rotation in the Lie algebra go over , the estimation becomes incorrect because of the SO(3) logarithm discontinuity. As opposed to these two filters, the D-LG-EKF does not suffer from those limitations and consequently it does not diverge, and efficiently smoothes the camera trajectory. As δt grows, the state model becomes less informative which is why the D-LG-EKF RMSE comes closer to the measurements RMSE, but always improve the precision compared to the EPnP input.

Finally, we also considered the case where the matrices Φ_G in the D-LG-EKF algorithm are replaced by identity matrices. We call this version of our formalism: D-LG-EKF NoPhi. It turns out that neglecting the matrices Φ_G only slightly reduces the performances of the algorithm. Therefore, depending on the required accuracy of the considered application, one can choose to replace them by identity matrices.

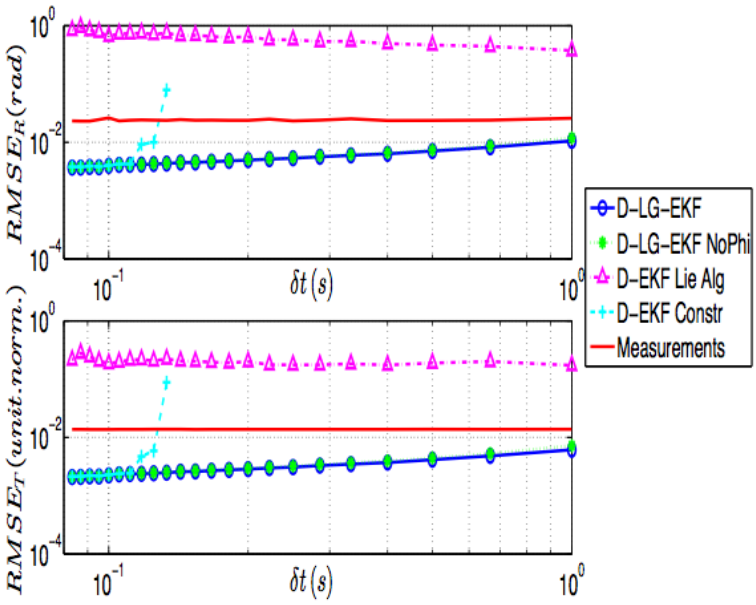

Figure 2.1-4: RMSE of the filters (in position T and orientation R) as a function of the sampling period δt.

Figure 2.1-4 shows errors computed on 2000 generated trajectories. The proposed algorithms (D-LG-EKF and D-LG-EKF NoPhi) achieve the lowest errors in all δt ranges

2.1.4 Conclusion

In this section, we have proposed a new generic algorithm called Discrete Extended Kalman Filter on Lie Groups that is motivated by the need to filter pose information in order to provide smoothed estimates of the person trajectory to be used for posture inference. The proposed algorithm generalizes the Discrete Extended Kalman Filter to the case where the state and the observations evolve on Lie group manifolds, such as the tuple (R,T,ω,v). Assuming the posterior distribution is a concentrated Gaussian distribution, we showed how to propagate and update the distribution parameters. The systematic methodology of our algorithm was illustrated on the camera pose estimation problem where both a constrained D-EKF and a D-EKF applied on the Lie algebra of the Lie group were outperformed.

This approach provides a unifying framework to filter the pose information obtained from the images. Performances are promising on synthetic data. From a computational point of view, the complexity of such a filtering is negligible compared to video frame decoding and feature extraction and matching. The 3d model of the environment is a prerequisite. Its computation for selected places was shown in deliverable D4.1 “Audio & Visual Sensing v1”. Future work will therefore will consider the integration of both the 3D model of the environment and the 3D pose estimation and filtering modules in the processing chain of the wearable camera video stream as an input to the posture estimation algorithms.

2.1.5 Technical annex

This annex introduces the technical description of the new Discrete Lie Group Extended Kalman Filter used for pose filtering in previous section. This work has been submitted to the conference EUSIPCO 2013.

Lie groups and Lie algebras

In this section we give the definitions and basic properties of matrix Lie Groups and Lie Algebra. For a detailed description of these notions the reader is referred to [2.1.10]. We focus on matrix Lie Groups since they cover most Lie groups of interest in signal and image processing. A Lie Group G is a group which has also the structure of a smooth manifold such that group composition and inversion are smooth operations. If G is a matrix Lie group, then $g \in \mathbb{R}^n$ and its operations are matrix multiplication and inversion with the identity matrix as identity element $Id_{n\times n}$. Note that an Euclidean space is a trivial matrix Lie Group. The matrix exponential exp_G and matrix logarithm log_G mappings establish a local diffeomorphism between an open neighborhood of $0_{n\times n}$ in the tangent space at the identity T_eG, called the *Lie Algebra* $\mathfrak{g}$, and an open neighborhood of $Id_{n\times n}$ in G. The Lie Algebra $\mathfrak{g}$ associated to a p-dimensional matrix Lie group is a p-dimensional vector space defined by a basis consisting of real matrices E_i for i = 1…p. Hence there is a linear isomorphism between $\mathfrak{g}$ and $\mathbb{R}^p$ that we denote as follows: $[\cdot]_G^\vee : \mathfrak{g} \to \mathbb{R}^p$ and $[\cdot]_G^\wedge : \mathbb{R}^p \to \mathfrak{g}$. For example, let $\mathfrak{a} \in \mathfrak{g} \subset v^{n\times n}$, then we have $[\mathfrak{a}]_G^\vee = a \in \mathbb{R}^p$. Thus we can define a basis $[E_i]_G^\wedge = e_i$ where $\{e_i\}$ is the natural basis of $\mathbb{R}^p$ and $\mathfrak{a} = \sum_{i=1}^{p} a_i E_i$ with $a=(a_1, \dots a_p)^T$. We also define $M \subset G$ and $S \subset \mathbb{R}^p$ as the sets on which exp_G and log_G are bijective functions. The two previous notions are summarized in the next diagram:

$$M \subset G \subset \mathbb{R}^{n\times n} \quad \underset{exp_G}{\overset{log_G}{\rightleftarrows}} \quad \mathfrak{g} \subset \mathbb{R}^{n\times n} \quad \underset{[\cdot]_G^\wedge}{\overset{[\cdot]_G^\vee}{\rightleftarrows}} \quad S \subset \mathbb{R}^p$$

Lie groups are usually non-commutative. The two following operators capture this property (for $X \in G$, $a,b \in \mathbb{R}^p$):

The Adjoint representation of G on R^p is defined as the operator Ad_G:

$$Ad_G(X)a = \left[X [a]_G^\wedge X^{-1}\right]_G^\vee$$

The adjoint representation of $\mathbb{R}^p$ on $\mathbb{R}^p$ is defined as the operator ad_G :

$$ad_G(a)b = \left[[a]_G^\wedge [b]_G^\wedge - [b]_G^\wedge [a]_G^\wedge\right]_G^\vee$$

Finally let's introduce the Baker-Campbell-Haussdorff formula which expresses the group product directly in $\mathbb{R}^p$

$$\left[log_G\left(exp_G\left([a]_G^\wedge\right) exp_G\left([b]_G^\wedge\right)\right)\right]_G^\vee = a + b + O\left(|a,b|^2\right) \quad \text{(A1)}$$

The following related formula will be useful for our derivations:

$$\left[log_G\left(exp_G\left([-a]_G^\wedge\right) exp_G\left([a+b]_G^\wedge\right)\right)\right]_G^\vee = a + \Phi_G(a)b + O\left(|b|^2\right) \quad \text{(A2)}$$

where $\Phi_G(a) = \sum_{m=0}^{\infty} \frac{(-1)^m}{(m+1)!} ad_G(a)^m$.

Concentrated Gaussian Distribution on Lie Groups

In this section we introduce the concept of concentrated Gaussian on Lie groups [2.1.11, 2.1.12] as a generalization of the normal distribution in Euclidean space which is used in the D-EKF formalism. In order to define such a distribution, the considered Lie group has to be a connected unimodular matrix Lie group. Henceforth, in the rest of the paper, when referring to Lie groups, we will consider this assumption to hold. Note that this is the case of most Lie groups of interest such as SO(3), SE(3), SL(3), $\mathbb{R}^n$... From [2.1.11] the following distribution can be defined:

$$\rho(X) = \alpha e^{-\frac{1}{2}\left([Log_G(X)]_G^{\vee} P^{-1} [Log_G(X)]_G^{\vee}\right)} \quad \text{(A3)}$$

Where is a normalizing constant, X∈G, G is a p-dimensional Lie group and P is a definite positive matrix. Probability of elements outside of M is set to zero. Let's define ε as follows: $\varepsilon = [log_G(X)]_G^{\vee}$ where ε∈S. When ρ(X) is tightly focused around the group identity (i.e the maximum of the eigenvalues of P is small), the distribution of ε can be approximated by a classical Euclidean Gaussian distribution defined on $\mathbb{R}^p$ of mean $0_{p\times1}$ and covariance matrix P. In this case, the distribution of X is called a concentrated Gaussian distribution on G around the identity. It can be moved around μ∈G using the left action of the Lie group, producing a concentrated Gaussian on G centered around (denoted $X \sim \mathcal{N}_G(\mu, P)$):

$$X = \mu exp_G\left([\epsilon]_G^{\wedge}\right) \quad \text{(A4)}$$

μ will be called the mean of X, ε can be seen as a Lie algebraic error of mean $0_{p\times1}$ and covariance P. Figure 2.1-5 provides a graphical interpretation of the transfer of the probability distribution from ε to X. Such a distribution allows us to describe the covariance of the state in $\mathbb{R}^p$ and hence using Euclidean tools while being invariant w.r.t the left action of the group on itself.

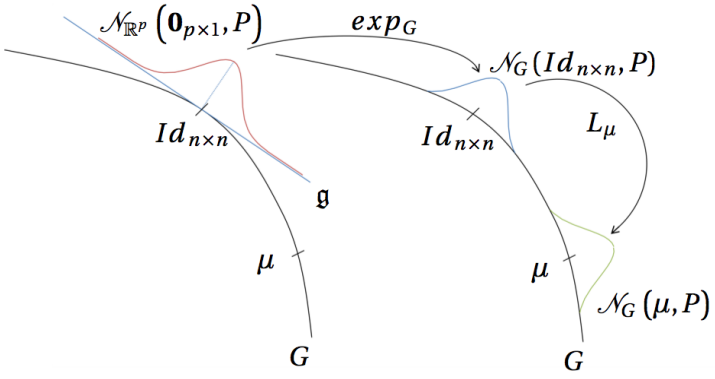

Figure 2.1-5 Concentrated Gaussian on Lie groups

DG-EKF Propagation step

We assume that the state posterior distribution at time k-1 is represented by $\mathcal{N}_G(\mu_{k-1|k-1}, P_{k-1|k-1})$. Therefore, the aim of this section is to show how to propagate $\mu_{k-1|k-1}$ and $P_{k-1|k-1}$ between two consecutive sensor measurements.

Mean Propagation : The state estimate is propagated using the state model without noise:

$$\mu_{k|k-1} = \mu_{k-1|k-1} exp_G\left(\left[\hat{\Omega}_{k-1}\right]_G^{\wedge}\right) \tag{A5}$$

where $\hat{\Omega}_{k-1} = (\mu_{k-1}, u_{k-1})$.

Covariance Propagation : In order to propagate the covariance, we study the Lie algebraic error propagation. The state error on G can be expressed as follows:

$$\begin{aligned} exp_G\left(\left[\epsilon_{k|k-1}\right]_G^{\wedge}\right) &= \mu_{k|k-1}^{-1} X_k \\ &= exp_G\left(\left[-\hat{\Omega}_{k-1}\right]_G^{\wedge}\right) exp_G\left(\left[\epsilon_{k-1|k-1}\right]_G^{\wedge}\right) \\ &\quad exp_G\left(\left[\Omega(X_{k-1}, u_{k-1}) + n_{k-1}\right]_G^{\wedge}\right) \end{aligned} \tag{A6}$$

Linearizing in $\varepsilon_{k-1|k-1}$ and using equations (A1) and (A2), one can obtain the following Lie algebraic error propagation:

$$\epsilon_{k|k-1} = \mathcal{F}_{k-1}\epsilon_{k-1|k-1} + \Phi_G\left(\hat{\Omega}_{k-1}\right) n_{k-1} + O\left(\left|\epsilon_{k-1|k-1}, n_{k-1}\right|^2\right) \tag{A7}$$

where

$$\mathcal{F}_{k-1} = Ad_G\left(exp_G\left(-\hat{\Omega}_{k-1}\right)\right) + \Phi_G\left(\hat{\Omega}_{k-1}\right) \mathcal{C}_{k-1} \tag{A8}$$

and

$$\mathcal{C}_{k-1} = \frac{\partial}{\partial \epsilon} \Omega\left(\mu_{k-1|k-1} exp_G\left(\left[\epsilon\right]_G^{\wedge}\right), u_{k-1}\right)|_{\epsilon=0} \tag{A9}$$

As in the D-EKF case, terms in $\mathcal{O}(|\varepsilon_{k-1|k-1}|^2)$ are neglected. Moreover, we do not consider terms in $\mathcal{O}(|\varepsilon_{k-1|k-1}, n_{k-1}|^2)$ since, because of the concentrated Gaussian assumption, n_{k-1} is assumed to be small.

Under these conditions: $\mathbb{E}(\varepsilon_{k|k-1}) = m_{k|k-1} = 0$. Finally, we obtain the following covariance propagation formula:

$$\begin{aligned} P_{k|k-1} &= \mathbb{E}\left[\epsilon_{k|k-1}\epsilon_{k|k-1}^T\right] \\ &= \mathcal{F}_{k-1} P_{k-1|k-1} \mathcal{F}_{k-1}^T + \Phi_G\left(\hat{\Omega}_{k-1}\right) R_{k-1} \Phi_G\left(\hat{\Omega}_{k-1}\right)^T \end{aligned} \tag{A10}$$

Propagation step summary : At the end of the propagation step, the estimated state is parametrized as follows:

$$X_k|z_1,\ldots,z_{k-1} \sim \mathcal{N}_G(\mu_{k|k-1}, P_{k|k-1}) \tag{A11}$$

where $\varepsilon_{k\,|\,k-1} \sim \mathcal{N}_{\mathbb{R}^r}(mk|k\text{-}1 = 0, Pk|k\text{-}1)$.

DG-EKF Update step

This step consists in incorporating the information coming from the measurement z_k into the Lie algebraic error. It is followed by a reparametrization of the state to satisfy to the concentrated Gaussian distribution assumption.

Lie algebraic error update : Let's define the following innovation term:

$$\begin{aligned} \bar{z}_k &= \left[log_{G'}\left(h\left(\mu_{k|k-1}\right)^{-1} z_k\right)\right]^{\vee}_{G'} \\ &= \left[log_{G'}\left(exp_{G'}\left(\mathcal{H}_k \epsilon_{k|k-1} + O\left(\left|\epsilon_{k|k-1}\right|^2\right)\right) exp_{G'}\left([w_k]^{\wedge}_G\right)\right)\right]^{\vee}_{G'} \end{aligned} \tag{A12}$$

where

$$\mathcal{H}_k = \frac{\partial}{\partial \epsilon}\left[log_{G'}\left(h\left(\mu_{k|k-1}\right)^{-1} h\left(\mu_{k|k-1} exp_G\left([\epsilon]^{\wedge}_G\right)\right)\right)\right]^{\vee}_G|_{\epsilon=\mathbf{0}} \tag{A13}$$

Using equation (A1), we obtain:

$$\bar{z}_k = \mathcal{H}_k \epsilon_{k|k-1} + w_k + O\left(\left|\epsilon_{k|k-1}, w_k\right|^2\right) \tag{A14}$$

As in the D-EKF case, terms in $\mathcal{O}(|\varepsilon_{k|k-1}|^2)$ are neglected. Moreover, we do not consider terms in $\mathcal{O}(|\varepsilon_{k|k-1}, w_k|^2)$ since, because of the concentrated Gaussian assumption, w_k is assumed to be small.

Equation (A14) is linear in $\varepsilon_{k|k-1}$ which evolves on $\mathbb{R}^p$. Therefore, we can apply the classical update equations of the Kalman filter [2.1.13] to update $\varepsilon_{k|k-1}$ into the posterior distribution as $\varepsilon_{k|k} \sim \mathcal{N}_{G'}(\bar{m}_{k|k}, \bar{P}_{k|k})$ where $\bar{m}_{k|k}$ and $\bar{P}_{k|k}$ can be calculated as follows:

$$\begin{cases} K_k = P_{k|k-1}\mathcal{H}_k^T\left(\mathcal{H}_k P_{k|k-1}\mathcal{H}_k^T + Q_k\right)^{-1} \\ m^-_{k|k} = \mathbf{0}_{p\times 1} + K_k\left(\bar{z}_k - \mathcal{H}_k \mathbf{0}_{p\times 1}\right) \\ P^-_{k|k} = (Id - K_k\mathcal{H}_k)P_{k|k-1} \end{cases} \tag{A15}$$

State Reparametrization : At the end of the update step, we expect to have $X_k = \mu_{k|k}\, exp_G([\varepsilon_{k|k}]_G^{\wedge})$ with $\mathbb{E}(\varepsilon_{k|k}) = 0_{p\times 1}$ (conditionally to $z_1;\ldots;z_k$) to satisfy the concentrated Gaussian distribution definition (A4). However we have $\mathbb{E}(\bar{\varepsilon}_{k|k}) = m_{k|k} \neq 0_{p\times 1}$. Hence, we perform the following re-parameterization:

$$\mu_{k|k} = \mu_{k|k-1} exp_G\left(\left[m^-_{k|k}\right]^{\wedge}_G\right) \tag{A16}$$

Thus, using equation (A2) and neglecting terms in $\mathcal{O}(|\bar{\varepsilon}_{k|k}|^2)$, we obtain:

$$m_{k|k} = \mathbf{0}_{p\times 1} \tag{A17}$$

$$P_{k|k} = \Phi_G\left(m^-_{k|k}\right) P^-_{k|k} \Phi_G\left(m^-_{k|k}\right)^T \tag{A18}$$

Update Step Summary : At the end of the update step, the estimated state is parametrized as :

$$X_k | z_1, \ldots, z_k \sim \mathcal{N}_G\left(\mu_{k|k}, P_{k|k}\right) \tag{A19}$$

which corresponds to $\varepsilon_{k|k} \sim \mathcal{N}_{\mathbb{R}}(mk|k = 0p\times 1, Pk|k)$.
The complete LG-EKF algorithm is summarized below

Algorithm 1: D-LG-EKF Algorithm

Inputs : $\mu_{k-1|k-1}, P_{k-1|k-1}, u_{k-1}, z_k$
Outputs : $\mu_{k|k}, P_{k|k}$

Propagation :

$$\mu_{k|k-1} = \mu_{k-1|k-1} exp_G\left(\left[\hat{\Omega}_{k-1}\right]^\wedge_G\right)$$
$$P_{k|k-1} = \mathcal{F}_{k-1} P_{k-1|k-1} \mathcal{F}^T_{k-1} + \Phi_G\left(\hat{\Omega}_{k-1}\right) R_{k-1} \Phi_G\left(\hat{\Omega}_{k-1}\right)^T$$

Update :

$$K_k = P_{k|k-1}\mathcal{H}^T_k\left(\mathcal{H}_k P_{k|k-1}\mathcal{H}^T_k + Q_k\right)^{-1}$$
$$m^-_{k|k} = K_k\left(\left[log_{G'}\left(h\left(\mu_{k|k-1}\right)^{-1} z_k\right)\right]^\vee_{G'}\right)$$
$$\mu_{k|k} = \mu_{k|k-1} exp_G\left(\left[m^-_{k|k}\right]^\wedge_G\right)$$
$$P_{k|k} = \Phi_G\left(m^-_{k|k}\right)\left(Id_{l\times l} - K_k\mathcal{H}_k\right) P_{k|k} \Phi_G\left(m^-_{k|k}\right)$$

3 Action recognition

3.1 Automatic prediction of visual attention maps in egocentric videos for content-action interpretation

3.1.1 Introduction

It is clearly noticeable throughout the Dem@Care project, that two different persons are involved in the video acquisition and video interpretation process from the GoPro wearable camera: the Actor (in our case the patient) who is wearing the video camera and the Viewer (in our case the doctor or the automatic system) who is interpreting the video. Automatic extraction of visually salient areas from video content provides important information to better understand which visual content is relevant to understand and analyze a given action or activity, in order to improve their automatic analysis. We detail in this Section the study of visual saliency from wearable video and will use these insights to complement the understanding of visual saliency, which is used to improve object classification for action recognition and activity monitoring in Section 4.1 .

During an action, the visual saliencies of the Actor and the Viewer are not the same. Indeed according to the physiological studies ([3.1.2], [3.1.3]), the human gaze anticipates the motor action of limbs when fulfilling an activity. When the viewer interprets the video acquired with wearable devices, he/she is much more interested in the action recorded and hence his saliency is different from the one of an actor. In various problems of video content interpretation, such as studies of neurodegenerative diseases [3.1.2], there is a need to predict a physiologically normal saliency map of an actor and to do this in an automatic way.

Since our action recognition approach is based on such automatically predicted saliency maps, it was necessary to complete a study of the comparison between the patient and doctor's saliency maps. This study allowed us to identify a temporal shift between these two saliency maps. Using this relation we propose an adapted prediction of saliency maps of an Actor for the beginning of actions using the objective saliency models we previously developed (see Section 4.1.2).

3.1.2 Study between actors' and viewers' points of view

In this section, we first explicit the methodology of building subjective saliency maps or in other words visual attention maps" and then provide a comparison. Furthermore we estimate the temporal relation between subjective saliency maps of Actor and Viewer using manual and automatic metrics.

Subjective saliency maps building method

The subjective saliency maps in images and videos are built from eye position measurements in image/video plan. With the help of eye-trackers, the gaze projection in video frames can be recorded. There are two reasons for which eye positions cannot be directly used to represent the areas of visual attention. First, the eye positions are only spots on the frame and do not represent the field of view. Secondly, in the case of Viewers to get accurate results, the eye positions of several experimental subjects observing video content, are recorded. These positions vary from one subject to another and represent sparse discrete maps. In order to

determine the areas of visual attraction in images and videos, we need dense maps. The method proposed by D. S. Wooding [3.1.7] has become the reference [3.1.8] since it fulfills these two constraints. In this method a two dimensional Gaussian is applied at the center of every eye-fixations. The Gaussian spread is set to an angle of 2° to reproduce the fovea projection of the screen as proposed in [3.1.9]. Then the Gaussians are summed-up and the final map is normalized. No matter for which recording of fixations is the eye-tracker used for, Wooding's method can be applied. Hence in our work, we apply this method to build both Actor's and Viewer's attention maps from the eye-recordings. We remind that the Actor data is obtained by the eye-tracker worn by the actor and hence the data of only one subject is recorded for each video, while several Viewers observe the same video to simulate video interpretation conditions.

Corpus description

For this work, a dataset containing the eye locations of the persons performing the actions (Actors) is needed in order to compare their gaze-recordings with the gaze coordinates of the people watching these actions on video (Viewers). Along with their paper [3.1.6], the authors have publicly released two datasets. The GTEA gaze dataset has been obtained using the Tobii eye-tracking glasses. The videos and gaze locations are recorded thanks to a camera and infrared light system integrated to the glasses. The videos are at a 15fps rate and a 640x480 pixel resolution. For the gaze location, two points per frame are recorded (30 samples per second). The subjects are asked to prepare a meal for themselves based on the different ingredients placed on the table in front of them. In total, 17 videos of 4min average are available, performed by 14 different participants. The different noticeable actions related with the preparation of a meal (e.g. spread jam, take milk, etc.) are listed in [3.1.6].

Eye-tracker setup

In order to get the eye location of the people watching the videos provided by the authors of [3.1.6], an eye–tracker experiment has been performed. The gaze positions have been recorded with a HS-VET 250Hz from Cambridge Research Systems Ltd at a rate of 250 eye positions per second. The experiment conditions and the experiment room were compliant with the recommendation ITU-R BT.500-11 [4.1.2]. Videos were displayed on a 23 inches LCD monitor with a native resolution of 1920x1080 pixels. To avoid image distortions, videos were not re-sized to the screen resolution. A mid-gray frame was inserted around the displayed video. 31 participants have been gathered for this experiment, 9 women and 22 men. For 3 participants some problems occurred in the eye-tracking recording process and so they have been discarded.

Human-based comparison of actions beginning

For our first comparison between actors and viewers, we manually annotated the moments when each of both sides focused on the beginning of a new action for 8 of the videos provided by the GTEA dataset. To decide whether a party was indeed focusing on a new action, we used the gaze provided by GTEA and the gaze recorded by our Eye-Tracker experiment. We considered the focusing of viewer's or actor's gaze on an object of interest related to a new action to be an acknowledgment of the realization from the corresponding party that a new action is happening. Since most of the actions cannot be considered as starting at a specific frame number, the results are an average value of every 4 frames to avoid the noise induced by manual annotation. Results are displayed in Figure 3.1-1. From this histogram one can clearly notice a peak of time difference between the realizations of actions for the two parties.

Indeed most of the actions are acknowledged by the viewer around 8 frames later than the actor (= 533ms which corresponds with the findings of ([4.1.3], [3.1.1]). This difference in frames/time will later on be referred as time-shift.

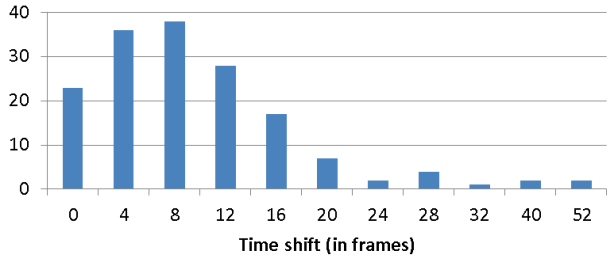

Figure 3.1-1: Histogram displaying the differences of frames between the viewer's and actor's focus on a new action

Comparison metrics for saliency maps

The normalized saliency maps of Actor and Viewer can be compared with help of dedicated metrics. A good survey has recently been published in [3.1.10] about them. From this survey and anterior work [4.1.12] we retained the Normalized Scan Path, the Pearson correlation coefficient (PCC) and the ROC area, or the Area Under Curve(AUC) as most frequently used and suitable for the comparison of pixel-based saliency maps. Since results prove the scores to be highly correlated between these metrics (Table 3.1-1), only the AUC is displayed here.

In AUC the problem is limited to a two-class prediction (binary classification). Pixels of one saliency map which is considered as "ground truth" as well as those of the predicted saliency map are labeled either as fixated or not fixated. A ROC curve plotting the false positive rate as a function of the true positive rate is used to present the classification result. The metric consists in computing the area under this ROC curve.

Comparison of Actor's and Viewer's saliency maps

After looking at the previous manual annotation results (Figure 3.1-1) confirming our expectations one can wonder whether this time-shift phenomenon is still observable when comparing two subjective saliency models. Based on the three metrics described in the previous paragraph we compared the similarity of saliency maps between actors and viewers for the frames belonging to the beginning of actions. The corresponding results are given in Figure 3.1-2. The AUC scores are displayed for different values of time-shift between actors' (fixed) and viewers' (varying in time) saliency maps. The NSS and PCC metrics are not displayed since the scores are highly correlated with AUC: see Table 3.1-1. The computation of these three metrics clearly brings to the same conclusion as the one two paragraphs before: the actors' saliency maps show more correspondence with those of the viewers when the latter are considered with a time-shift. An also noticeable and expectable result to be extracted from this figure is that the standard deviation (grey bars) gets lower when the correspondence score gets higher (around 14 frames =933ms time-shift).

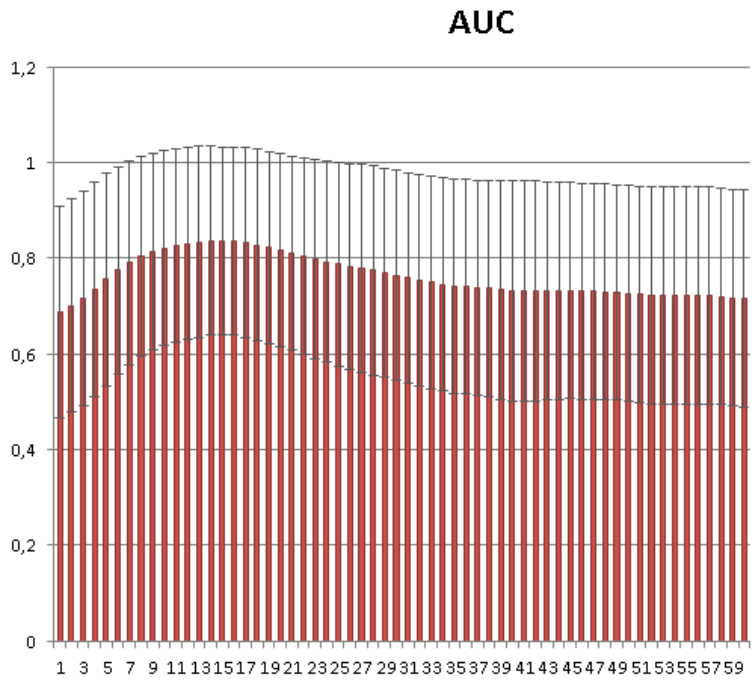

Figure 3.1-2 AUC scores between actor's and viewer's saliency maps for different time-shifts (in frames)

Table 3.1-1 Correlation scores between the three different metrics for section 3.1.2

AUC/NSS	AUC/PCC	PCC/NSS
0.996	0.997	1.0

3.1.3 Adaptation of objective saliency maps to retrieve the actor's saliency maps

In the current literature, all the automatic saliency maps models are proposed aiming to approach the viewer's one in the best manner. In the previous section we have both by manual and automatic calculations showed that the viewer's and actor's points of view are indeed more correlated when shifted in time. In this section we tackle a new problem: based on the previous results can we adapt the objective saliency maps automatically extracted from signals to match those of the actor?

Time-shift based model adaptation

The objective saliency models presented in 4.1.2 have been designed to locate the areas of interest in videos. Since one can conclude based on the previous results (Figure 3.1-1, Figure 3.1-2) that actors have a tendency to focus on the areas of interest before the viewers, it is fair to assume that the automatic saliency maps can be adapted to match the actor's one for the beginning of actions by taking into account this shift in time. We firstly compared the AUC, PCC, and NSS scores when comparing different automatic saliency maps with either those of the viewers or the actors for the frames corresponding to the beginning of actions. According to the results in Figure 3.1-2 where the highest score is computed with a time-shift of 14

frames (=933ms), we computed the same metrics scores when comparing actors' saliency maps with the automatic ones shifted by 14 frames.

Results

Results of the AUC scores computed for the three different comparisons described in the previous paragraph are shown in Figure 3.1-3. Again the NSS and PCC scores are not displayed since highly correlated with AUC (see Table 3.1-2). Firstly, as can be expected, we can see the difference of scores when comparing the automatic saliency maps to the actor's one versus the viewer's. The results demonstrate that the objective maps correlate more with the viewer indeed, the scores of correspondence with the actor's being low. The results obtained with the new automatic time-shift based model display higher scores when compared to the subjective actor's saliency maps as expected.

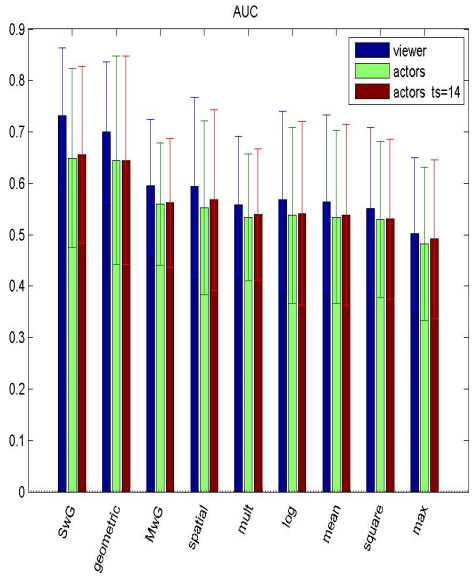

Figure 3.1-3: AUC scores for the comparison between different models of automatic saliency maps

Figure 3.1-3 shows AUC scores for the comparison between different models of automatic saliency maps. (SwG stands for square with geometric, MwG stands for multiplication with geometric, mult stands for multiplication). In blue: viewers vs automatic, in green: actors vs automatic, in red: actors vs the time-shifted adapted model of automatic saliencies.

Table 3.1-2: Correlation scores between the three different metrics for section 3.1.3

AUC/NSS	AUC/PCC	PCC/NSS
0.945	0.947	1.00

3.1.4 Conclusions

Accordingly to the research results in vision and motor control, we formulated the assumption of temporal shift of visual saliency between the person executing different activities, i.e. the Actor (Patient) and the Viewer (doctor) who interprets this content a posteriori. Psychovisual experiments confirm this assumption. Based on these results, we proposed a prediction of visual saliency maps for objective models, that takes into time information and delays to allow for interpretation of actions from the points of view of bith the Actor and the Viewer.

3.2 Robust Human Action Recognition from static cameras

3.2.1 Introduction

Activities of Daily Living (ADL) from video are of particular interest and should be recognized in order to build behavioural and lifestyle profiles of elderly people with dementia. A novel activity recognition method is developed and tested on benchmark datasets, as well as datasets of actual PwD recorded at the Greek Centre for Alzheimer's Disease and Related Disorders. The proposed method is based on the creation of trajectories at multiple spatiotemporal locations and scales, in order to ensure scale and viewpoint invariance. Action descriptors (features like HOG, HOF, HOGHOF) are extracted at these locations and used to form a dictionary describing a set of ADLs of interest (those included in the training videos used). The method we developed features the introduction of statistical sequential change detection, which allows the detection of changes in time in a principled manner, theoretically shown to achieve quickest change detection results. This leads to the temporal segmentation of the trajectory data, which until now has been taking place based on heuristics (e.g. it is very common to assume an activity lasts for 15 frames and apply this to all data).

Our activity recognition leads to the annotation of videos ranging from recordings in limited lab environments to more challenging "in the wild" videos obtained from Youtube or from movies. Often, these methods 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 high accuracy, comparable to the state of the art, but at a lower computational cost.

3.2.2 Static Camera ADL Recognition overview

In this section, we present the main parts of our ADL recognition approach. In particular, the block diagrams below show both action representation and action recognition. At the action representation phase, a spatiotemporal grid is placed over the video and sampled at multiple scales to gauge both the finer and more general details of the activities taking place. Statistical change detection leads to the automated temporal segmentation of the dense multi-scale trajectories extracted, breaking the video into sub-activity segments. Recognition takes place using a variation of the well known Bag of Words approach, which has been shown to lead to accurate recognition results. Testing of numerous SoA encoding and recognition techniques has taken place to ensure the best results are obtained.

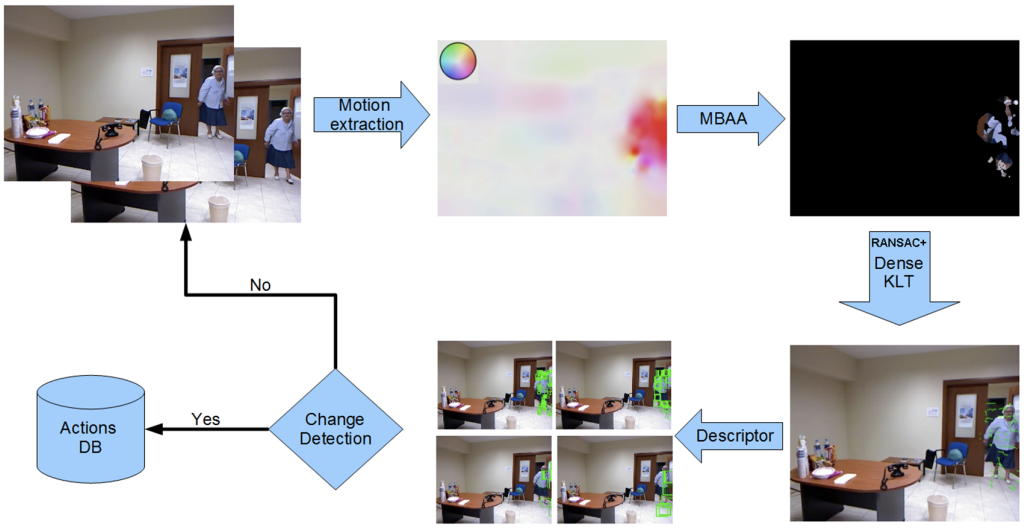

Figure 3.2-1 Action representation for recognition of ADL from static camera using dense trajectories

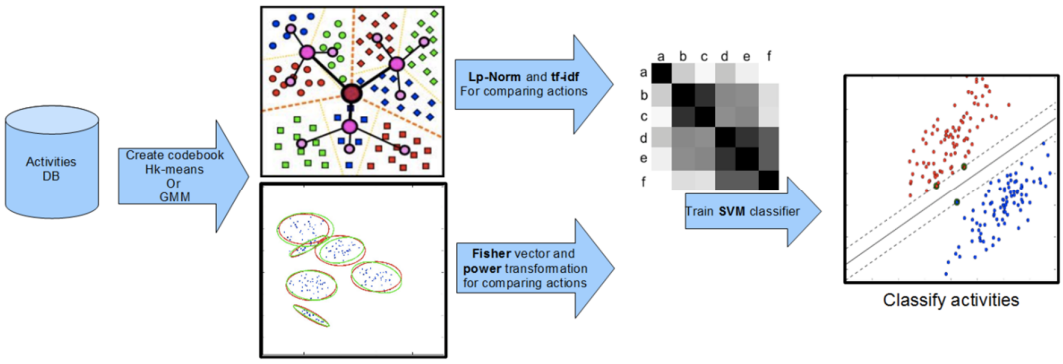

Figure 3.2-2 Action recognition framework for recognition of ADL from static camera

As a first step, we attempt to sample meaningful spatio-temporal interest points to describe the motion taking place in the video. Currently, SoA methods focus on the temporal extension of the Harris corner detector [3.2.5], so called Harris-3D, however it does not suit our purpose as it does not capture geometric information, namely the spatial layout of the interest points, when introduced in a BoW framework. Dense sampling [3.2.6] [3.2.1] and tracking over time has also been deployed, however its drawback is its high computational cost.

Motivated by these needs and drawbacks of current techniques, we develop a very accurate but computationally efficient method, which leads to results comparable to or better than the current SoA.

To increase the efficiency of the method, Motion Boundary Activity Areas (MBAAs) are formed to localize pixels whose motion undergoes a change. As Figure 3.2-1 shows, MBAAs often include a very small area of the video frame, drastically reducing computational cost.

Multi-scale grids are created and dense sampling on these grids, inside the MBAAs, lead to the extraction of interest points, which are then tracked using the pyramidal KLT tracker [3.2.2].

In order to ensure the most accurate tracking results, we eliminate outlier tracked points by applying the RANSAC algorithm, which indeed increases the algorithm's accuracy.

Finally, HOGHOF descriptors characterize the actions in order to obtain a complete description of the feature points' appearance and motion, at a low computational cost.

We automatically determine the temporal extent of the trajectories by the application of statistical sequential change detection on each trajectory's velocity values over time.

These features are then used to form descriptors of the actions taking place, for which various SoA encodings (inspired from recent success in object recognition) are tested in order to determine the best one. Finally, SVMs are used to classify the activities and experiments with benchmark data and Dem@Care data lead to accurate recognition, which provides a clear picture of each scheme's advantages and disadvantages.

3.2.3 Static Camera ADL Feature extraction

Motion boundary activity areas are first extracted by making the assumption that changes in motion are introduced either by an actual change in the gradient velocity values, or by noise in the data. In the latter case we make the assumption that the noise follows a Gaussian distribution, so the following two hypotheses can be used to represent data with and without changes in motion, respectively:

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

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

where vk(r) represents the change in successive flow values and zk(r) the respective Guassian noise. A classical measure of Gaussianity is the kurtosis, whose value is equal to zero for Gaussian data. Based on this, we extract optical flow, changes in optical flow and applying the kurtosis metric on each pixel's change in velocity, we find the MBAAs. In the figure below we can observe these steps and the characteristic reduction in data that they produce.

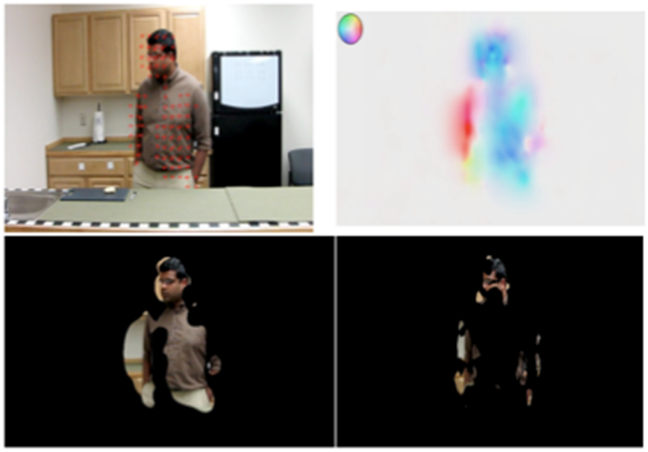

Figure 3.2-3: Processing steps

Figure 3.2-3 shows the processing steps of the proposed method. 1st row: ADL video frame, corresponding optical flow magnitude and colored direction. 2nd row: Activity Area (AA) from flow values, MBAA from changes in flow. It is evident that the MBAA leads to a great reduction in computational cost.

3.2.4 Static Camera Spatiotemporal descriptor

In order to extract meaningful spatiotemporal interest points, we first sample the MBAAs at multiple spatiotemporal scales. KLT tracking is then applied and post-processed via the RANSAC algorithm to extract accurate feature point correspondences. This leads to action trajectories of feature points, around which we compute HOG and HOF descriptors, forming "action cuboids" at four scales (every 8, 16, 24 and 32 pixels) to ensure scale invariance. Histograms of these features are then computed to characterize each action of interest in a BoW framework, capturing the global motion but also localization information (due to the spatiotemporal segmentation).

The extracted temporal trajectories can be tracked throughout their duration; however each action can be broken down into sub-activities. These can be localized in time by applying statistical sequential change detection on the motion data, finding the points in time where velocity changes are abrupt. In order to detect meaningful changes in activity, we consider the HOF descriptors of the tracked interest points. In particular, we assume that the tracked points' HOF descriptors initially follow a multidimensional Gaussian distribution f0, while after a change in the activity (or sub-activity) has occurred, the tracked point velocities will follow a new distribution f1 that is not yet known. The initial distribution is estimated from the data in an initial window of frames w0, while the current one is found from the latest w1 frames. This leads to the test statistic:

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

Statistical sequential change detection is based on the Cumulative Sum of successive test statistics as proposed by Page in [15]:

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

This leads to the temporal segmentation of action trajectories into sub-activity trajectories, for which we extract spatiotemporal features that are more meaningful as they are directly related to the activity taking place since they originate from changes in the tracked points' HOF statistics.

3.2.5 Bag of Features for Recognition

The spatiotemporally detected interest points and their descriptors are introduced into a Bag of Features framework for action recognition. As a first step, a vocabulary needs to be constructed by clustering the extracted descriptors. We tested different encoding methods, namely simple K-means clustering, hierarchical K-means clustering and GMM-based clustering. The cluster centers are separated and video sequences are quantized either by hard binning or by using Fisher transform technique. Former results are used for constructing a Chi-Square Kernel, while latter ones are passed through a linear kernel in order to be introduced to train a SVM classifier. After testing all clustering solutions mentioned above, we found that a GMM vocabulary with Fisher encoding passed to a linear SVM will lead to the most efficient system for recognition. Extensive experiments took place with benchmark videos, as well as realistic videos filmed for the purposes of Dem@Care and achieved results comparable to or better than, the state of the art, as detailed in the section that follows.

3.2.6 Experiments for Recognition of ADLs

Experiments for recognizing activities of daily living were first conducted on benchmark datasets filmed from the University of Rochester (URADL). These videos feature actors from the lab performing daily activities like eating a banana, making a phone call, drinking water etc. However, they are quite limited in their range of action, not moving much more than their face and arms when performing these activities (in some cases they also turn around, e.g. to open the refrigerator). Variability is present, nonetheless, in the people carrying out these actions, who are from a variety of ethnicities, age groups and of both genders.

In order to obtain a better picture of our algorithm's effectiveness in realistic environments, we conducted recordings of elderly people with and without dementia (ranging from mild dementia to Alzheimer's), performing a set of activities of daily life at the Greek Association for Alzheimer's and Related Disorders **(**http://www.alzheimer-hellas.gr/english.php**).** Details on these recordings are presented here and are also made available on the Dem@Care website: http://www.demcare.eu/news/73-multi-sensor-recording-at-the-greek-alzheimer-s-a**.** In particular, 32 individuals were recorded from each group (health, with dementia), performing 11 activities of daily life, namely Clean Up (CU), Drink Beverage (DB), End Phonecall (EP), Enter Room (ER), Eat Snack (ES), Hand Shake (HS), Prepare Snack (PS), Read Paper (RP), Serve Beverage (SB), Start Phonecall (SP), Talk to Visitor (TV). All participants read and signed appropriate consent forms and in most cases were accompanied by family members or other carers, especially in cases of more advanced dementia. These recordings took place with the Kinect RGB camera, although motion and health sensors were also placed on the individuals, for lifestyle recording purposes. Audio recordings of these individuals also took place and are analyzed and discussed in the appropriate sections of this deliverable. The resulting dataset is much more challenging than URADL, due to the higher variability of the subjects themselves, the fact that they move around in a more realistic home-

like environment which is quite unconstrained. A few examples are shown in the figure below, where the faces of the participants are hidden for privacy purposes.

Figure 3.2-4: Dem@Care recordings of ADL at the Greek Association for Alzheimer's and Related Disorders

For the URADL dataset we tested clustering via k-means, hierarchical k-means with Lp non-linear kernel and a GMM Fisher vector. The best results were acquired using a GMM vocabulary combined with Fisher encoding, as the tables below show:

Table 3.2-1 Evaluation results of clustering via k-mean on URADL dataset

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										

Table 3.2-2 Evaluation results of clustering via hierarchical k-mean with Lp non-linear kernel on URADL dataset

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										

Table 3.2-3 Evaluation results of clustering via hierarchical k-mean with GMM Fisher vector on URADL dataset

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										

It should also be noted that without using RANSAC to eliminate outlier trajectory correspondences, the average accuracy is decreased to 86.7%, validating our use of RANSAC. Results comparing our method with SoA approaches are shown in Table 3.2-4:

Table 3.2-4 Results comparing our method with SoA approaches

	Our	[3.2.6]	**[3.2.1],**k-means, chi square kernel	**[3.2.1]**GMM Fisher
Average accuracy	91.33%	89.33%	92%	92.67%
Computational time	15 hr 50 min 23 sec	-	23 hrs 1 min 15 sec	-

We note that our method achieves results comparable to the SoA at a lower computational cost, while the performance of other methods (e.g. [3.2.1]) varies, depending on the encoding used.

Even better results were produced when applying our method to the more challenging Dem@Care dataset. Since GMM with Fisher encoding gave the best results for URADL, we only employ this combination on the Dem@Care videos. The tables below show that our method now surpasses the SoA results of [3.2.1] by around 3.37%, despite the more challenging nature of the data.

Table 3.2-5 Evaluation results of the proposed algorithm on Dem@Care dataset

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 3.2-6 Evaluation results of SoA ([3.2.1]) on 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											

3.2.7 Conclusion

The method presented here for the recognition of activities of daily living from static cameras is shown to achieve accurate recognition results for realistic and challenging videos. It has the advantage over SoA methods of a low computational cost, while producing accurate results. The use of MBAAs significantly reduces the computational cost, while the extracted trajectories are temporally segmented based on changes in their statistical nature in order to derive meaningful sub-trajectories that are related to the changes in the person's motion.

3.3 Relative Dense Tracklets for Human Action Recognition

3.3.1 Introduction

The main objective of Dem@Care WP4 is to analysis of daily activities of the people with dementia in their domestic environment. To do this, the system should be able to automatically recognise actions such as food preparation, walking, housekeeping, exercise or sleeping.

Recent studies on human action recognition have shown that bag-of-word approaches reach high action recognition accuracy [3.3.16], [3.3.19], [3.3.24]. Unfortunately, these approaches have problems to discriminate similar actions, ignoring spatial information of features. A common way to overcome these limitations is to use either spatio-temporal grids [3.3.16] or multi-scale pyramids [3.3.17]. However, these methods are still limited in terms of a detailed description and provide only a coarse representation.

To differ from these ideas, we propose novel descriptors based on a dynamic coordinate system. Our suitable design descriptors introduce important spatial information to the bag-of-words model enhancing its discriminative power. Our main idea is that action recognition ought to be performed using a dynamic coordinate system corresponding to an object of interest. Computing relative tracklets, we are able to keep spatial information in a bag-of-words framework. We propose to use a head position as a center of our dynamic coordinate system, providing description invariant to changes in camera viewpoint. Our novel descriptors improve the discriminative power of features and help to distinguish similar features detected at different positions (e.g. to distinguish similar features appearing on hands and feet). We perform an extensive evaluation on three datasets: popular KTH dataset, challenging ADL dataset and our locally collected Hospital dataset. Consistently, performed experiments show that our representation enhances the discriminative power of features and bag-of-words model, bringing significant improvements in action recognition performance.

The main contributions of our work are summarized as follows:

We offer a novel action recognition approach based on a dynamic coordinate system. We propose to compute relative tracklets, introducing spatial information to the bag-of-words model. The tracklets computation is based on their relative positions according to the central point of our dynamic coordinate system. As this central point, we choose the center of a head to provide camera invariant description.

We report experimental results on three action recognition datasets (KTH, ADL and our collected Hospital dataset), showing that our representation enhances the discriminative power of features and bag-of-words model, bringing significant improvements in action recognition performance.

3.3.2 Related works

Over the last few years, many different action recognition techniques have been proposed. However, due to appearance variations of both people and actions, camera view point changes, occlusions, noise, and enormous amount of video data, action recognition still remains a challenging problem.

Existing techniques can be divided into four categories. The first group of techniques uses silhouette or body contour information to represent an action [3.3.1], [3.3.12], [3.3.18]. Such techniques usually require precise segmentation, which is often difficult to achieve, especially in real-world videos. The second category of techniques uses local spatio-temporal features [3.3.7], [3.3.13], [3.3.15], [3.3.20], [3.3.26], [3.3.29]. The local spatio-temporal features are able to capture both visual and motion appearance. They are robust to viewpoint and scale changes, they are easy to implement and fast to process. Moreover, they do not require object localization and in addition they are robust to background clutter. Over the last few years, many different local interest point detectors (like Harris3D [3.3.15], Cuboid [3.3.5], Hessian [3.3.34] or Dense sampling [3.3.33]) and many spatio-temporal descriptors (like HOG [3.3.16], HOG3D [3.3.13], HOF [3.3.16], Cuboid [3.3.5] or ESURF [3.3.34]) have been proposed. One of the most commonly used descriptors in the literature showing a high performance over the various datasets [3.3.3], [3.3.33] are: Histogram of Oriented Gradients (HOG) and Histogram of Oriented Flow (HOF) descriptors [3.3.16]. The former describes the local visual appearance and the latter characterizes the local motion appearance of an interest point. The third category contains methods analysing motion trajectories [3.3.9], [3.3.23], [3.3.28], [3.3.32]. This group of techniques usually requires tracking of feature points or objects [3.3.6], [3.3.21], [3.3.27], [3.3.35]. Recent techniques, based on feature tracking, have shown high action recognition rate, especially when combining trajectories with local spatio-temporal features. Becha et al. [3.3.9] have proposed to track corner points using HOG tracker, and then represent feature trajectories by angle descriptors. Raptis et al. [3.3.28] have proposed spatio-temporal feature descriptors (named average of gradient orientation and average of optical flow) that capture the local structure of an image around trajectories tracked over time. Messing et al. [3.3.23] have proposed to track Harris feature points using KLT tracker, and then represent trajectories by temporal velocity histories. Recently, especially dense trajectories have drawn a lot of attention and have shown to obtain high performance for action recognition in videos. Wang et al. [3.3.32] have proposed to use dense short trajectories together with HOG, HOF and MBH (Motion Binary Histograms) features. Wu et al. [3.3.35] have proposed to use Langrangian particle trajectories which are dense trajectories obtained by advecting optical flow over time. Raptis et al. [3.3.27] have proposed to extract salient spatio-temporal structures by forming clusters of dense optical flow trajectories. Then, the assembly of these clusters into an action class is governed by a graphical model.

Most of the recent techniques, based on local spatio-temporal features and trajectories, use the bag-of-words model. The bag-of-words model has shown to achieve high recognition rate across various datasets [3.3.16], [3.3.19], [3.3.24]. It simplifies the structure of 3D video data assuming conditional independence across spatial and temporal domains. It encodes global statistics of features computing histogram of feature occurrences in a video. However, the bag-of-words model has limitations. The main drawback of this technique is that it ignores important spatial position of features. A common way to overcome this limitation is to use either spatio-temporal grids [3.3.16] or multi-scale pyramids [3.3.17]. However, these methods provide only a coarse representation, hence they are still limited in terms of a detailed description.

In contrary, we design a novel approach based on short relative dense tracklets, local spatio-temporal features and bag-of-words model. We propose novel descriptors based on a dynamic coordinate system, which introduce important spatial information to the bag-of-words model. Our novel descriptors improve the discriminative power of features and bag-of-words model, and help to distinguish similar features detected at different positions (e.g. to distinguish similar features appearing on hands and feet).

3.3.3 Dense Multi-Scale Tracklet Extraction

The amount of data retrieved from a video content usually depends on the action-video parameters such as: a length of the action taking place and a video resolution. As certain daily living actions like walking or sitting could only last a few seconds, information provided by commonly used tracking algorithms such as KLT and SIFT might not be enough for recognizing these actions. Similarly to [3.3.32], we cope with this problem by employing dense tracklets extracted on multiple spatial scales. For each scale, we sample feature points on a grid with a step size of W pixels and track densely sampled feature points using optical flow and median filtering. Using dense tracklets, we are able to distinguish similar and short actions. Moreover, limiting the length of tracklets to L frames, we avoid a drifting problem and enhance the discriminative properties of tracklets. As tracklets themselves do not contain spatial-temporal information, we propose to introduce relative positions of trajectories, computed using a dynamic coordinate system. The central point of this system is selected using a head position, which is computed by applying our robust head detection algorithm.

Figure 3.3-1: Samples of estimated head positions for the KTH (first row) and ADL (second row) datasets

3.3.4 Head detection

Head detection is of particular interest in our recognition framework, thus we have to ensure robust localization of this body part. As head is a common pattern, we needed to combine several techniques for estimating head position. Cues provided by object detectors (people, head, and face) are combined with motion information (background subtraction) and object tracking results. We employ: (1) histogram of oriented gradients for people detection [3.3.4], (2) LBP patterns for head detection [3.3.25], and (3) haar-like features for face detection [3.3.4]. Refining object detection results by background subtraction step, we produce preliminary detection-based tracking results. Then, detection results are used as an input to the tracking algorithm (applied for forward and backward tracking in a video) [3.3.10] to overcome missed detections.

Both detection-based tracking and TLD tracking algorithm [3.3.10] provide multiple hypothesis along which we select the most likely one. This selection is based on our probability framework *P*, which smooths trajectories by replacing rapid object displacements with the interpolated results. The final position of the head is obtained by maximizing the trajectory-depended probability:

$$\mathcal{P}(l_h|t_{h,i}) = \sum_{z\in\{f,b\}} \mathcal{P}(l_z \mid t_{h,i} \propto t_{z,j})\ \mathcal{P}(t_{h,i} \propto t_{z,j})$$

where l_h is a head location and $t_{h,i}$ is their corresponding trajectory. α describes proportional variance of trajectory $t_{h,i}$ w.r.t. trajectory of other body part $t_{z,j}$. f and b refer to the face detection and the full body detection, respectively. Sample head positions estimated by this method are shown in Figure 3.3-1.

3.3.5 Combined Multi-Scale Tracklet (CMST) Descriptors

In this section, we introduce our novel descriptor, which contains both shape characteristics of a tracklet and relative positions of tracklet elements according to the central point of the dynamic coordinate system. We focus on home care applications, thus we use head as a reference point to provide camera invariant description. Our novel descriptors improve the discriminative power of features and bag-of-words model, and help to distinguish similar features detected at different positions (e.g. to distinguish similar features appearing on hands and feet). We define our CMST descriptor as:

$$\Phi = \left[(\vartheta_X)^T \quad (\vartheta_Y)^T \quad (X_{TR} - X_H)^T \quad (Y_{TR} - Y_H)^T\right]^T$$

where the first two elements of the vector $((V_X)^T, (V_Y)^T)$ are refereed to as Shape Multi-Scale Tracklet (SMST) descriptor, which represents shape characteristics of a tracklet based on the displacement vector descriptor [3.3.32], and the two remaining parts correspond to our Relative Multi-Scale Tracklet (RMST) descriptor.

Shape Multi-Scale Tracklet (SMST) Descriptor: Encoding a local motion pattern of a given tracklet $t_i = \{(x_1, y_1), ..., (x_L, y_L)\}$ of length L, we compute displacement vectors θ_X and θ_Y :

$$\theta_X = \Delta(X - \overline{X})$$
$$\theta_Y = \Delta(Y - \overline{Y})$$

where $X = [x_1 , x_2 , ..., x_L]^T$ and $Y = [y_1 , y_2 , ..., y_L]^T$. Symbols $\bar{X}$ and $\bar{Y}$ represent mean of the vector X and Y, respectively. Then, we normalize displacement vectors by the sum of

$$\vartheta_X = \frac{\theta_X}{\sum_{i=1}^{L} \sqrt{\theta_{X_i}^2 + \theta_{Y_i}^2}}$$

$$\vartheta_Y = \frac{\theta_Y}{\sum_{i=1}^{L} \sqrt{\theta_{X_i}^2 + \theta_{Y_i}^2}}$$

their magnitudes:

where θ_{Xi} and θ_{Yi} represent i-th elements of the vector X and Y , respectively. Finally, we obtain our shape characteristic tracklet representation by defining a vector ψ, which is the result of the concatenation of the vector θ_X and θ_Y :

$$\psi = \left[(\vartheta_X)^T \quad (\vartheta_Y)^T\right]^T$$

Relative Multi-Scale Tracklet (RMST) Descriptor: Encoding a local motion pattern of a given tracklet $t_i = [(x_j , y_j), ..., (x_{j+L} , y_{j+L})]$ (where L is the length of the tracklet and j is the frame number, where the tracklet occurred for the first time) with respect to the head trajectory $t_h = [(x' , y_k), ..., (x' , y_m)]$ (where $k \leq j$ and $j + L \leq m$),
we define the RMST descriptor by:

$$\phi = \left[(X_{t_i} - X_{t_h})^T \quad (Y_{t_i} - Y_{t_h})^T\right]^T$$

Our CMST descriptors introduce the relative positions of features to the bag-of-words approach. Our novel descriptors improve the discriminative power of features and bag-of-words model, and help to distinguish similar features detected at different positions. Fusing the discriminative power of both SMST and RMST descriptors, we significantly improve action recognition accuracy. Our final descriptor (CMST) allows classifier to recognize an action even in the case when the estimation of the head is not perfect or head detection is missing. This is obtained by the discriminative power of the SMST descriptor.

3.3.6 Action Recognition using CMST features

Additionally, to increase the discriminative power of tracklets, we compute the HOG (Histogram of Oriented Gradients) and HOF (Histogram of Oriented Flow) features along space-time neighbourhood of each tracklet [3.3.32]. The former feature describes the local visual appearance and the latter characterizes the local motion appearance of a tracklet.

The tracking algorithm, used to compute the SMST and HOG-HOF descriptors, was selected based on its use in the literature, and provide a good baseline for comparison with the state-of-the-art techniques. However, our action representation method can be also used together

with any other tracking algorithm.

To represent videos, we apply the bag-of-words model for each feature class (SMST-RMST, HOG-HOF) independently. We construct visual vocabularies from training videos clustering computed features. Then, we assign each feature to its closest visual world. The concatenated histograms of visual word occurrences over video forms becomes the final representation.

To classify a new video sequence, we use multi-class non-linear Support Vector Machines (SVM). We apply a $\chi 2$ distance to compare two n-bins histograms $H_i = [H_i(1), ..., H_i(n)]^T$ and $H_j = [H_j(1), ..., H_j(n)]^T$ This distance is then converted into SVM multi-channel $\chi 2$ kernel using a multi-channel generalized Gaussian kernel

3.3.7 Experiments

We perform an extensive set of experiments on multiple datasets to demonstrate the effectiveness of the proposed descriptors. We evaluate our approach on three datasets for human action recognition: popular KTH dataset, challenging ADL dataset and locally collected Hospital dataset. Sample images from video sequences of these datasets are presented in Figure 3.3-2.

Implementation details

We compute HOG and HOF descriptors on a spatio-temporal grid of size nx × ny × nt , where: nx = 2, ny = 2 and nt = 3. For each individual cell of the grid, we compute a 8-bins histogram of orientation for the HOG and 9-bins histogram for the HOF. We normalize both descriptors with the L2 norm.During the quantization process of calculated features, we use the k-means clustering technique and nearest neighbour algorithm. To compute the bag-of-words representation, features are quantized to the codebook size of 1000, which has shown empirically to give good results. As a metric to calculate a distance between features and visual words, we use the L2 norm.

In all our experiments, we apply the cross-validation technique to both gauge the generalizability of the proposed approach, and select the most discriminative parameters.We use the Leave-One-Out Cross-Validation (LOOCV) technique, where videos of one person are used as the validation data, and the remaining videos as the training data. This is done repeatedly so that videos of each person are used once as the validation data

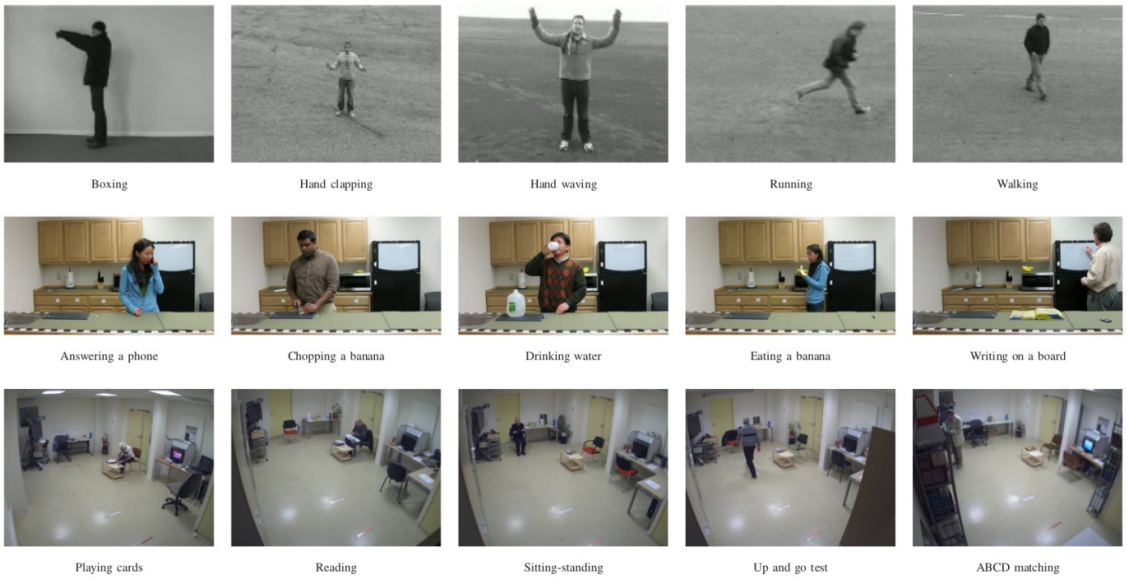

Figure 3.3-2: Sample frames of KTH(first row), ADL (second row), and Hospital (third row)

KTH Dataset

The KTH dataset [3.3.31] does not contain real home care videos. However, we have decided to evaluate our approach on it due to its popularity and possibility to compare our approach with most of the state-of-the-art techniques.

The KTH dataset contains six types of human actions: walking, jogging, running, boxing, hand waving and hand clapping. Each action is performed several times by 25 different subjects in four different scenarios: outdoors (s1), outdoors with scale variation (s2), outdoors with different clothes (s3) and indoors (s4). In total, the dataset contains 599 video files. All sequences were recorded with 25 fps frame rate.

The dataset contains a set of challenges like: scale changes, illumination variations, shadows, different scenarios, cloth variations, inter and intra action class speed variations and low resolution (160×120 pixels spatial resolution).

Table 3.3-1 KTH dataset: Evaluation of SMST, RMST and CMST descriptors

		recognition rate (%)				
	method	s1	s2	s3	s4	overall
Official Split	SMST	98.15%	88.89%	88.89%	90.74%	91.67%
	RMST	96.30%	88.89%	87.04%	92.60%	91.21%
	CMST	**98.15%**	**92.59%**	**92.59%**	**96.30%**	**94.91%**
LOOCV	SMST	98.00%	92.00%	93.29%	94.67%	94.49%
	RMST	98.67%	89.33%	95.30%	96.67%	94.99%
	CMST	**99.33%**	**93.33%**	**97.32%**	**98.67%**	**97.16%**

There are two commonly used experimental setups to evaluate an approach on the KTH dataset: splitting-based scheme and LOOCV technique. Therefore, to compare our approach with all of them, we evaluate our approach on both experimental setups.
LOOCV evaluation scheme: We follow the recent evaluations [3.3.8], [3.3.35], [3.3.36], [3.3.38] on the KTH dataset using LOOCV scheme. The experimental results are presented in Table 3.3-1. Comparison of our approach with state-of-the-art methods using LOOCV technique is presented in Table 3.3-2. The detailed comparison for each scenario separately is presented in Table5. We observe that overall and for each scenario independently, our CMST descriptors outperform SMST descriptors, achieving 97.16%, 99.33%, 93.33%, 97.32% and 98.67%, respectively. We also observe that over all and for scenarios s1, s3 and s4, our RMST descriptors outperform SMST descriptors. Although continuous scale changes in scenario s2 cause time to time inaccurate and missing head estimations (what results in slightly lower accuracy of RMST descriptors compared to SMST features), our CMST descriptors still improve action recognition accuracy and outperform both SMST and RMST features. The results clearly show that our representation enhances the discriminative power of features and bag-of-words model, and outperforms state-of-the-art techniques. HOG-HOF features do not improve action recognition accuracy on this dataset (the accuracy 97.16% is already very high).
Splitting-based evaluation scheme: We also follow the original experimental setup, where video samples are divided into two parts: the training set and testing set. The testing set consists of 9 subjects (2, 3, 5, 6, 7, 8, 9, 10 and 22) and the training set consists of 16 remaining subjects. The results from the experiments are presented in Table 3.3-2. Comparison of our approach with state-of-the-art methods in the literature, using splitting-based evaluation scheme, is presented in Table 3.3-2. Overall, our approach obtains 94.91% recognition rate. Also in this case, we observe that the CMST descriptors improve action recognition accuracy both overall and for each scenario independently. The results clearly show that our representation enhances the discriminative power of features and bag-of-words model, and outperforms state-of-the-art techniques.

Table 3.3-2 : KTH dataset: Comparison of our approach with state-of-the-art methods in the literature using both official splitting-based evaluation scheme and LOOCV technique.

Official Split			LOOCV		
Method	Year	Accuracy (%)	Method	Year	Accuracy (%)
Laptev *et al.* [16]	2008	91.8%	Liu *et al.* [19]	2009	93.8%
Yuan *et al.* [37]	2009	93.3%	Ryoo *et al.* [29]	2009	93.8%
Zhang *et al.* [38]	2012	94.1%	Wu *et al.* [36]	2011	94.5%
Wang *et al.* [32]	2011	94.2%	Kim *et al.* [11]	2007	95.33%
Gilbert *et al.* [7]	2011	94.5%	Zhang *et al.* [38]	2012	95.5%
Kovashka *et al.* [14]	2010	94.53%	Wu *et al.* [35]	2011	95.7%
Becha *et al.* [9]	2012	94.67%	Lin *et al.* [8]	2011	95.77%
Our method		**94.91%**	**Our method**		**97.16%**

Table 3.3-3: KTH dataset: Comparison of our approach with state-of-the-art methods in the literature for each scenario separately using LOOCV technique.

method	recognition rate (%) s1	s2	s3	s4	avg.
Wu *et al.* [36]	96.7%	91.3%	93.3%	96.7%	94.5%
Lin *et al.* [8]	98.83%	94.00%	94.78%	95.48%	95.77%
Our method	**99.33%**	**93.33%**	**97.32%**	**98.67%**	**97.16%**

Table 3.3-4: KTH dataset: Comparison of our approach with state-of-the-art methods in the literature for each scenario separately using LOOCV technique

Method	Recognition Rate (%)
SMST	76.67%
RMST	78.67%
CMST	**88.00%**
CMST + HOG-HOF	**92.00%**

ADL Dataset

The ADL (University of Rochester Activities of Daily Living) dataset [3.3.23] contains ten types of human activities of daily living, selected to be useful for an assisted cognition task. The full list of activities is: answering a phone, dialling a phone, looking up a phone number in a telephone directory, writing a phone number on a whiteboard, drinking a glass of water, eating snack chips, peeling a banana, eating a banana, chopping a banana, and eating food with silverware. Each action is performed three times by five different people. In total, the dataset contains 150 video sequences recorded with 30 fps frame rate and 1280 × 720 pixels spatial resolution. The videos were down-sampled to the 640 × 360 pixels spatial resolution.

The dataset contains a set of challenges like: different shapes, sizes, genders and ethnicities of people, and difficulty to separate activities on the basis of a single source of information (e.g. eating banana and eating snack or answering a phone and dialling a phone). Results from the experiments are presented in Table 3.3-4. Comparison of our approach with state-of-the-art methods in the literature using LOOCV technique is presented in Table 3.3-5. We observe that SMST descriptors overall achieves 76.67% and our RMST descriptors improve recognition rate up to 78.67%. Moreover, our CMST descriptors improve action recognition rate up to 88.0%, which means that our descriptors improve accuracy by 11.33%. We also fuse HOG-HOF features with our CMST descriptors and achieve 92.00% recognition rate, which means that our approach improves the accuracy by 15.33% compared to SMST descriptors. All these results clearly show that our representation enhances the discriminative

power of features and bag-of-words model, bringing significant improvements in action recognition performance.

Table 3.3-5: ADL dataset: Comparison of our approach with state-of-the-art methods in the literature using LOOCV tech-nique.

Method	Year	Recognition Rate (%)
Matikainen *et al.* [22]	2010	70%
Satkin *et al.* [30]	2010	80%
Banabbas *et al.* [2]	2010	81%
Raptis *et al.* [28]	2010	82.67%
Messing *et al.* [23]	2009	89%
Our method		**92.00%**

Table 3.3-6 ADL dataset: Evaluation of SMST, RMST,CMST, and CMST with HOG-HOF descriptors usingLOOCV technique.

Method	Recognition Rate (%)
SMST	76.67%
RMST	78.67%
CMST	**88.00%**
CMST + HOG-HOF	**92.00%**

Hospital Dataset

Most of the existing public action recognition datasets can be divided into a few categories: (a) low resolution videos of relatively simple actions (like Weizmann and KTH datasets) which do not include object interactions, (b) video sequences from broadcast television channels, YouTube, and personal cameras (like UCF Sports, YouTube, and UCF50 datasets) where often a person is not fully visible, videos are recorded in a significant distance from people, videos are often pixelated, blurred, and contain significant camera motion and background clutter, (c) video samples from movies (like Hollywood and Hollywood2 datasets) where often only parts of people and actions are visible, and camera view point is constantly moving, and (d) videos of activities of daily living (like ADL dataset) where the camera is set in front of the actor and background does not significantly change between videos. Therefore, a new dataset is needed for recognition of realistic human activities of daily living.

We have locally collected a dataset, created with the help of medical scientists. The new dataset contains 8 types of real human activities of daily living. The full list of activities is: (a) playing cards, (b) matching ABCD sheets of paper, (c) reading, (d) sitting down and standing up, (e) turning back, (f) standing up and moving ahead, and (g) walking back and forth (2

activities). These activities were selected and annotated by medical doctors.

The experiments have been approved by the national official committee, the Committee for the Protection of Patients in Biomedical Research. Once people have been selected and have agreed (with their relatives) to participate in the studies, videos were recorded during regular consultations of patients at hospital. The videos were recorded over a period of several months, for every recording slight changes were made to the positioning of the camera and objects in the room. As a result, we have obtained a dataset of 55 patients recorded at 640 × 480 pixels spatial resolution.

Our dataset contains a set of challenges like: different shapes, sizes, genders and ethnicities of people, occlusions, and multiple people (sometimes both patient and doctor are visible).

Our proposed CMST descriptors combined with HOG-HOF features achieve high action recognition rate (92.96% accuracy) improving the recognition rate by 6.67% compared to the SMST descriptors. Experiments on this dataset again confirm the effectiveness of our method and show that our representation enhances the discriminative power of features and bag-of-words model, bringing significant improvements in action recognition performance.

3.3.8 Conclusion and future work

We proposed a novel action recognition approach based on a dynamic coordinate system. Our approach employs head detection for computing relative tracklets. These relative tracklets enhance the discriminative power of features and bag-of-words model introducing important information on their spatial position. The proposed approach was evaluated on three benchmark datasets for human action recognition. Obtained results clearly show that our approach improves action recognition performance and outperforms existing state-of-the-art techniques. In future work, we intend to examine more efficient learning algorithms (like Multiple Kernel Learning) to combine features from the bag-of-words model. We also intend to examine different human body parts as reference points for computing our relative tracklets.

Within Dem@Care context, the proposed method has been tested in an off-line manner as it depends on slow pre-processing steps, such as head detection, optical flow etc. For future work, we will study fast and robust version of these pre-processing algorithms to be able to satisfy the online requirement of WP4.

This work has been accepted for publication in the 10th IEEE International Conference on Automatic Face and Gesture Recognition (April 2013).

4 Activities monitoring

4.1 Object recognition in egocentric vision for activity monitoring: a saliency-based approach

4.1.1 Introduction

Identifying human activities becomes a key problem to be solved, since it represents the basis to generate semantic video indices that allow medical staff to efficiently navigate over the video footage. Traditionally, the detection of human activities has been addressed by analyzing human motion patterns. More precisely, various approaches have successfully made use of the motion patterns associated to spatio-temporal interest points (STIP) in the video ([4.1.28], [4.1.10]). In addition, the study of ego-motion has also resulted in successful approaches for first-person camera videos analysis [4.1.17]. However, in the particular case of egocentric view, we claim that an action can be effectively defined as a sequence of manipulated or observed objects, usually known as 'active' objects or 'objects of interest'. This assumption generally holds for video showing many household activities and, in particular, for the intended IADL scenario. In that sense, the literature already shows examples in which the outputs of object detectors become the features for later action recognition systems in egocentric video: in [4.1.25], [4.1.11].

In contrast to the well-known sliding window approaches for object detection and recognition ([4.1.14], [4.1.18]), and due to the specific nature of the first-person view contents, we aim to drive the object recognition process using visual saliency. Under the particular scenario of egocentric video, there is usually a strong differentiation between active (manipulated or observed by the user wearing the camera) and passive objects (associated to background) and, therefore, spatial, temporal and geometric cues can be found in the video content that may help to identify the active elements in the scene. Incorporation of visual saliency in video content understanding is a recent trend. The application of saliency modelling for object recognition on video serves for identifying areas where objects of interest are located. Then, features in these areas can be extracted for object description. Several works in the literature have shown the utility of human gaze tracking in the analysis of egocentric video content and, in particular, in the activity recognition task ([4.1.12], [4.1.23]).

In this section, we present our object recognition system that relies on visual saliency-maps to provide more precise object representations, that are robust against background clutter and, therefore, improve the precision of the object recognition task. We further propose to incorporate the saliency maps into the well known Bag-of- Words (BoW) [4.1.6] paradigm for object recognition. The benefits of this approach are multiple: a) the computation of saliency maps is generic (category-independent) and therefore a common step for any object detector, b) compared to sliding window approaches ([4.1.7], [4.1.14]),by looking at the salient area we can avoid much of the computationally overhead due to the scanning process and therefore use more complex non-linear classifiers, c) since the saliency maps are automatically computed in both training and test data, our method does not need bounding boxes for training, what dramatically reduces the human resources devoted to the database annotation.

We consider two differentiated scenarios of application. The first one is a constrained scenario in which all the subjects perform actions in the same room (the Lab environment) and, therefore, interact with the same objects. This task can be seen as a specific object recognition problem since there is not intra-class variation between instances of a category other than this caused by the strong ego-motion, changes on the viewpoint, illumination, occlusions, etc. The second scenario, on the contrary, is unconstrained, and corresponds to recordings made at different locations. In this case users interact with various instances of the same objects: e.g. in a home environment, a patient performs daily activities using his/her own utensils and devices, that probably differ from those ones available in another home. The second scenario is therefore much more difficult than the first one, due to the large intra-class variability as well as to the limited amount of training data (a few instances of each object category).

Throughout this study, we assess our method in both scenarios, showing its strengths and weaknesses in comparison to other methods in the literature.

4.1.2 Building objective saliency maps for egocentric videos

Spatial, Geometric and Temporal Saliency Approaches

In order to drive the video analysis to the regions that are potentially interesting to human observers we need to model visual saliency on the basis of video signal features. In this work, we have considered three basic approaches to generate saliency maps, each of them built using a particular source of information: spatial, geometric and temporal. In the following paragraphs, we will briefly describe the method that gives place to each map.

Spatial saliency: S_s: proposed in [4.1.4], it is based on various color contrast descriptors that are computed on the HSV color space, due to its closeness to human perception of color. In particular, 7 local contrasts are computed, namely:

Contrast of Saturation: A contrast occurs when low and highly saturated color regions are close.

Contrast of Intensity: A contrast is visible when dark and bright colors co-exist.

Contrast of Hue: A hue angle difference on the color wheel may generate a contrast.

Contrast of Opponents: Colors located at the hue wheel opposite sides create very high contrast.

Contrast of Warm and Cold Colors: Warm colors – red, orange and yellow – are visually attractive.

Dominance of Warm Colors: Warm colors are always visually attractive even if no contrast is present in the surrounding.

Dominance of Brightness and Saturation: Highly bright and saturated regions have more chances of attracting the attention, regardless of the hue value.

The spatial saliency value $S_s(i)$ for each pixel i in a frame is computed by averaging the outputs associated to the 7 color contrasts.

Temporal saliency S_t: this saliency models the attraction of attention to motion singularities in a scene. The visual attention is not grabbed by the motion itself, but by the residual motion for each pixel, e.g. the difference between the estimated motion

for each pixel and the predicted camera motion based on a global parametrization. The very general process of computing a temporal saliency map is as follows: first, for each frame in the video, a dense motion map $v(i)$ that contains the motion vectors in each pixel i in the image is computed using the optical flow technique described in [4.1.10]. Then, a 3x3 affine matrix A that models the global motion associated to the camera movements is computed. For that end, the well known robust estimation method RANSAC [4.1.15] has been used in order to successfully handle the presence of outliers (e.g. areas of the image associated to objects that move differently than the camera). Furthermore, since the central area of each frame constitutes the most likely region where moving objects appear, this region is not considered for the affine matrix estimation, thus reducing the proportion of outliers. Next, the residual motion $r(i)$ is computed by compensating the camera motion:

$$r(i) = v(i) - A\,x$$

where x stands for the spatial coordinates of each pixel i, $x = (x,y,1)^T$. Finally, the values of the temporal saliency map $S_t(i)$ are computed by filtering the amount of residual motion in the frame. The authors of [4.1.4] reported that the human eye cannot follow objects with a velocity higher than 80°/s [4.1.8]. According to these psychovisual constraints, a post-processing filter was proposed in [4.1.4] that decreased the saliency when motion was too strong. Applying this filtering stage to our first-person camera videos was however too restrictive due to the strong camera motion so that we have preferred to consider a simpler filtering stage that normalizes and computes the saliency map as follows:

$$S_t(i) = \min\left(\frac{|||r(i)||}{K}, 1\right)$$

where K has been heuristically computed depending on image dimensions (H,W) , as $K = \max(H,W)/10$.

Geometric saliency S_g: There are two major observations about saliency in egocentric video: on the one hand, some studies on general purpose video confirm the so-called center bias hypothesis, that is the attraction of human gaze by the geometrical center of an image ([4.1.1], [4.1.4]). On the other hand, in videos recorded with wearable cameras, the camera is usually set-up to point specific areas of interest: e.g. the gaze fixation if the camera is located in glasses, or an area just in front of the human body where the hands usually manipulate objects, in case it is located on the body. Generally, central geometric saliency is dependent on the wearable camera position and might be shifted in image plane [4.1.1]. In the present research, we work on datasets with either eye-centered or body-centered camera, thus using the center-bias hypothesis. Hence, following the approach in [4.1.4], the geometric saliency map $S_g(i)=N((x_0,y_0),(s_x,s_y))$ is computed as 2D Gaussian located at the screen center with a spread $s_x=s_y=5°$. However, this attraction may change with the camera motion. This is explained by the anticipation phenomenon [4.1.19]. Indeed, the observer of video content produced by a wearable video camera tries to anticipate the actions of the actor. The action anticipation is performed according to the actor body motion which is expressed by the camera motion. Hence we propose to simulate this phenomenon by

moving the 2D Gaussian centered on initial *geometric saliency point* in the direction of the camera motion projected in the image plane. A rough approximation of this projection is the motion of image center computed with the global motion estimation model previously described.

Results on the basic approaches are shown in Figure 4.1-1(columns "Spatial-"Temporal"-"Geometric"). As one can notice from the figures, spatial and temporal saliency maps show more precise localization of the objects of interest whereas the geometric approach provides a coarse approximation of the visual saliency. However, saliency information appears more scattered or disaggregated for the first two approaches, being more compact and therefore robust for the geometric technique. For an object recognition task, we consider that the perfect saliency map is a trade-off between precision and compactness, requirement that, based on the examples, is not completely fulfilled by any of the basic approaches. Hence, to overcome this issue, we propose two extensions: a) to incorporate a post-processing step on the spatial and temporal techniques that provides more compact saliency representations and, b) to investigate fusion schemes that successfully combine the three approaches taking advantage of their precision and compactness, respectively.

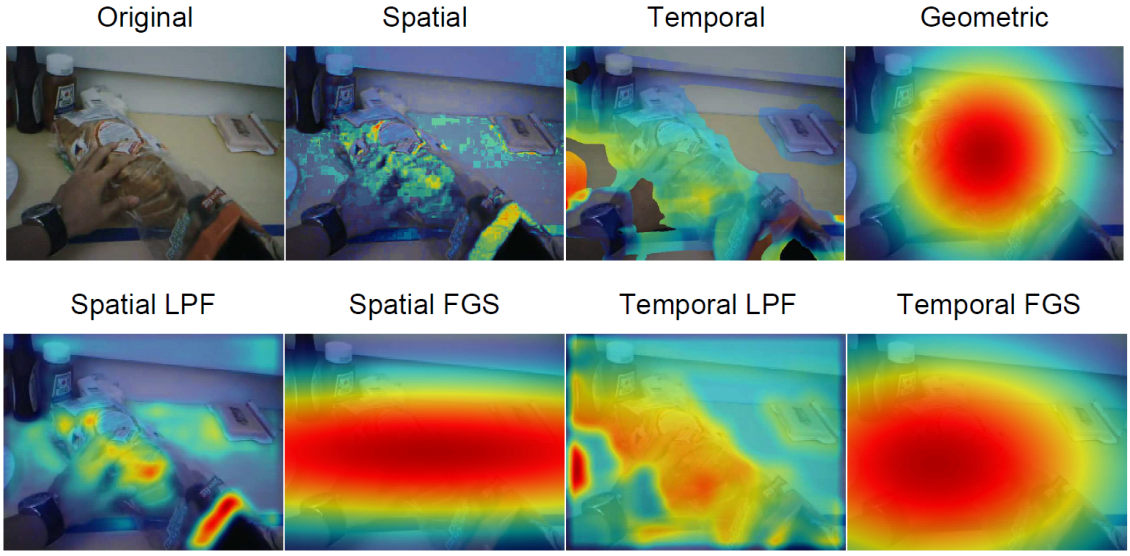

Figure 4.1-1: Results of various saliency maps for one frame in GTEA dataset.

Figure 4.1-1 shows the results of various saliency maps for one frame in GTEA dataset. In this figure, the three basic techniques spatial, temporal and geometric are shown. In addition, for spatial and temporal maps, two types of post-processing are also included (LPF and FGS).

Post-processing: Setting-up suitable saliency maps for object recognition

As already mentioned, we propose to use an additional postprocessing stage to obtain more compact representations for the spatial and temporal saliency. In particular, we have evaluated two methods: a) a very simple spatial low-pass filtering using a Gaussian mask (LPF), and b) a method that fits a Gaussian Surface (FGS) on the original map. The LPF approach, shown in Figure 4.1-1, simply provides a smooth version of the original saliency maps. However, if the standard deviation of the spatial Gaussian is large enough, results may fulfill our requirements

of compactness. For the second approach, given the original saliency mask S, we propose to fit a Gaussian surface of the form:

$$G(x,y) = A\, e^{\left[-\frac{1}{2}\left(\frac{x-x_g}{\sigma_x}+\frac{y-y_g}{\sigma_y}\right)\right]}$$

where $\theta=(A, x_g, y_g, s_x, s_y)$ are the parameters to be estimated in the fitting process. In practice, we minimize the square error between the two maps

$$e^2 = \sum_{x,y} [S(x,y) - G(x,y)]^2$$

In the experimental section we will assess the performance of both post-processing approaches.

Fusion strategies for saliency maps

Once the basic spatial, temporal and geometric saliency maps has been introduced, we aim to evaluate how their combination into spatio-temporal-geometric saliency masks S_{stg} might improve the representation of the area of interest in the image. For that end, several fusion strategies have been proposed and evaluated in this work. Again, although most of them have been already proposed in [4.1.2] in a video quality assessment task, we briefly describe their computation:

Multiplication (Mult): a multiplicative fusion strategy model as:

$$S_{stg}^{mult}(i) = S_s(i) \times S_t(i) \times S_g(i)$$

Mean: the average of the three methods as:

$$S_{stg}^{mean}(i) = \frac{1}{3}\left(S_s(i) + S_t(i) + S_g(i)\right)$$

Square: the squared Minkovsky pooling reinforced by multiplicative pooling:

$$S_{stg}^{sq}(i) = S_s(i) \times S_t(i) \times S_g(i) + \frac{1}{3}\left(S_s^2(i) + S_t^2(i) + S_g^2(i)\right)$$

Max: maximum pooling:

$$S_{stg}^{max}(i) = max\left(S_s(i), S_t(i), S_g(i)\right)$$

Log: logarithmic combination model:

$$S_{stg}^{log}(i) = \frac{1}{3}\left(\log\left(1 + S_s(i)\right) + \log\left(1 + S_t(i)\right) + \log\left(1 + S_g(i)\right)\right)$$

A visual example of the fusion strategies is shown in Figure 4.1-2. In addition, all of them will be evaluated in the experimental section (see 4.1.4).

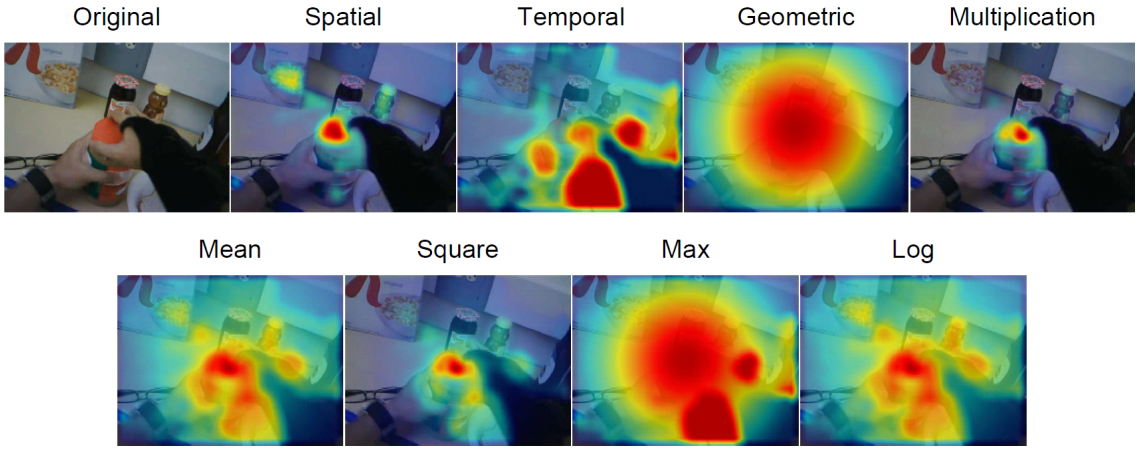

Figure 4.1-2: Results of various fusion strategies for computing spatio-temporal-geometric saliency maps.

4.1.3 Influence of saliency maps in object recognition

Low-level feature extraction and description

In this section we will describe our approach for object recognition in first-person camera videos using saliency masks. As we have already mentioned in the introduction, we aim to detect the region of interest (ROI) of each frame so that we can effectively build more precise image representations. The processing pipeline of our approach is included in Figure 4.1-3. We build our model on the well-known Bag-of-Words (BoW) paradigm [4.1.6], and propose to add saliency masks as a way to improve the spatial precision of the original Bag-of-Words approach. For each frame in a video sequence, we extract a set of N local descriptors using a dense grid of local circular patches [4.1.27]. Based on some experiments, we have set the radius of the circular patches to 25px, and the step size between each local patch of 6px, thus leading to a high degree of overlapping between neighboring local regions. Next, each local patch $n=1..N$ is described using a 64-dimensional SURF descriptor d_n [4.1.3] which has shown similar performance than the SIFT descriptor [4.1.31] in our experiments, whereas it is of half the dimension. Each descriptor d_n is then assigned to the most similar word $j=1..V$ in a visual vocabulary by following a vector-quantization process. The visual vocabulary, computed using a k-means algorithm over a large set of descriptors in the training dataset (about 1M descriptors in our case), has a size of V visual words. As we will show in the evaluation section, we have experimented with visual vocabularies of different sizes V. In parallel, our system generates a saliency map S of the frame with the same dimensions of the image and values in the range [0,1] (the higher the more salient is a pixel).

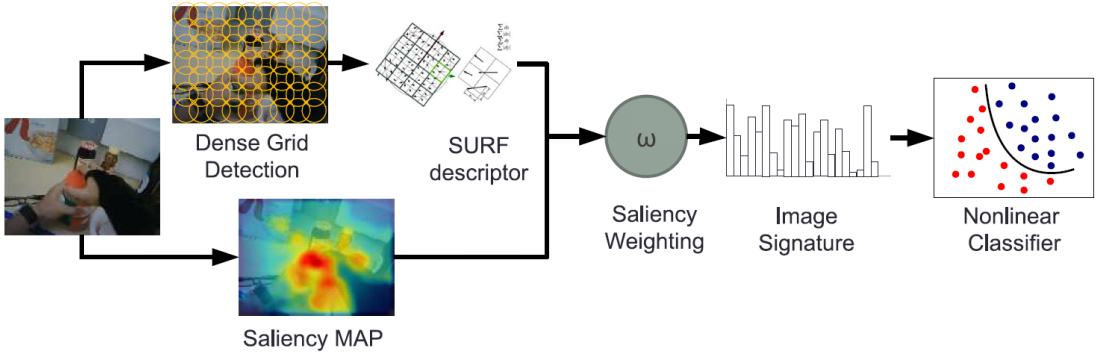

Figure 4.1-3: Object recognition pipeline

Object recognition with Saliency Weighting

In the traditional Bag-of-Visual-Words approach [4.1.6], the final image signature H is the statistical distribution of the image descriptors according to the codebook. This is made by first assigning each local descriptor to a visual word in the vocabulary and then computing a histogram of word occurrences by counting the times that a visual word appears in an image. Instead of doing this hard assignment, we propose to apply what we call *saliency weighting*, a sort of soft-assignment based on saliency maps. With saliency weighting, the contribution of each image descriptor is defined by the maximum saliency value found under the circular region W_n associated to the index n. In other words, descriptors over salient areas will get more weight in the image signature than descriptors over non-salient areas. Therefore, the image signature is a V-dimensional vector H that can be computed as follows:

$$H_j = \sum_{n=1}^{N} \alpha_n w_{nj}$$

where the term $w_{nj} = 1$ if the descriptor or region n is quantized to the visual word j in the vocabulary, and the weight α_n is defined as:

$$\alpha_n = \max(S(s)), \quad s \in \Omega_n$$

where Ω_n represents the set of pixels contained in the n^{th} circular region of the dense grid, and $S(s)$ is a saliency map. Finally, the histogram H is L1-normalized in order to produce the final image signature.

It is worth stressing the difference between our weighted histogram with hard-assignments and the histogram with soft assignments previously proposed in the literature [4.1.16]. In that work, given a descriptor, a similarity measure is computed with respect to all the words in the vocabulary so that various bins of the histogram can be incremented according to these similarities. On the contrary, our method is assigning each descriptor to just one word in the vocabulary but then is weighting its contribution to the histogram using the saliency map information. In fact, if necessary, our method might be combined with the one in [4.1.6].

Once each image is represented by its weighted histogram of visual words, we use a non-linear classifier to detect the presence of a category in the image. In particular, we have employed a SVM classifier [4.1.5] with a χ^2 kernel, which has shown good performance in visual recognition tasks working with normalized histograms as those ones used in the BoW paradigm [4.1.26].

4.1.4 Experiments and results

In this section, we assess our model in various challenging datasets with egocentric videos. As we have already, mentioned we aim to recognize objects under two different scenarios: constrained, in which all videos contain the same instances of the involved object categories, and the unconstrained, in which each video shows a different environment with varying instances of the object categories.

Datasets

We have assessed our approach with three publicly available egocentric video datasets.

The first one is the GTEA Gaze dataset [4.1.12], which consists of 17 standard definition (640x480) video sequences, captured at a frame rate of 15 frames per second, and performed by 14 different subjects using Tobii eye-tracking glasses. Due to the lack of object annotations in this dataset, we have extracted and annotated 595 frames from the videos so that we can easily perform our tests over a set of still images. The whole dataset has been divided into two sets, namely: a) the training set (294 frames), and b) the test set (300 frames). Furthermore, we aimed to detect 15 object categories in this database. Due to its limited size, we have used this dataset to compare various system configurations.

The second dataset is the GTEA dataset [4.1.13] for Object Recognition. This dataset, recorded at 30 frames per second in 1280x720 definition, contains 7 types of daily activities, each performed by 4 different subjects. In this case, the camera is mounted on a cap worn by the subject. Weak annotations are already available for this dataset. They identify active objects on each frame belonging to 16 object categories, but do not include the object location. Since all the users have been recorded in the same room interacting with the same objects, we have evaluated our constrained scenario using this dataset. For that end, we have followed the same setup described in [4.1.25], using the users 2-4 for training the algorithms and the user 1 for testing.

The third dataset used in the experiments is the ADL dataset [4.1.25], that contains videos captured by a chest-mounted GoPro camera on users performing various daily activities at their homes. The high definition videos (1280x960) are captured at rate of 30 frames per second and with 170 degrees of viewing angle. In total, 27,064 frames have been accurately annotated providing bounding boxes for objects belonging to 44 categories. In our experiments, we have just considered those objects labeled as 'active' (those being interacted or observed by the users) for both training and testing purposes. This dataset is more challenging than the other two since both the environment and the object instances are completely different for each user, thus leading to an unconstrained scenario. However, we have evaluated both scenarios with this dataset: the constrained one by randomly dividing the whole set of frames into a training and test set (50/50%), and the unconstrained, by doing so at the video/user level.

The final dataset has been recorded for the sake of the Dem@care project. We are still in the case of a constrained scenario since all the videos have been recorded in the same environment: CHU Nice. In these videos the patients are asked to execute different tasks written on a list, involving the manipulation of objects. The most important ones have been to chosen for creating a final taxonomy of 21 categories listed below:

Basket, bills, cards, checkes, comptes, enveloppes, inst, kettle, mapinst, map, pen, phone, pillbox, plasticglass, remote, tablet, teabag, teabox, TV, and wateringcan.

As for the ADL dataset, the videos are recorded with a GoPro camera.

The annotation has been performed by different people in UB1 using a homemade software. It is important to stress out the fact that only objects of interest have been annotated. Defining an object of interest is a non trivial and especially very subjective task however the basic principles include

The directly manipulated objects

The non manipulated objects but interacted with (reading a book) or shown

The non manipulable objects but looked at (a map on the wall)

In total 13184 objects have been annotated and used to train the categories.

Table 4.1-1 gives the number of annotated frames for each objects.

Table 4.1-1: The number of annotated frames for each objects

Categories	Number of annotated instances
Basket	182
bills	282
cards	24
checkes	1575
comptes	802
enveloppes	401
inst	2677
kettle	509
map	239
medinst	162
pen	471
phone	1428
pillbox	1101
plasticglass	220
remote	371

tablet	1728
teabag	359
teabox	93
TV	326
wateringcan	168

This dataset is evaluated by dividing the annotated videos by half: 50% for training and 50% for testing. Dividing by videos and not by frames is a voluntary choice in order not to train on images which are almost exactly similar to testing ones.

Setting-up the final model

In this section we compare various system configurations on the GTEA Gaze dataset. The objective is then to select the final system setup that provides the best performance, which will be compared with other state-of-the-art methods in the two envisaged scenarios.

Evaluating the basic approaches for saliency maps

We have firstly evaluated our basic approaches for generating the saliency maps. In addition, we have included two reference methods in the comparison:

Basic BoW(B-BoW): the Bag ofWords approach that generates image signatures considering whole images. This method becomes the basic reference and allows us to evaluate the improvement achieved by our saliency masks.

BoW with Ideal Masks (I-BoW): this approach makes use of the ideal ground truth masks provided in the annotation. Since it evaluates our approach when the saliency masks correspond with the ground-truth, it constitutes the theoretical limit in its performance. It is worth noting how this ideal binary masks are used both on training and testing, thus incorporating the annotations in the whole recognition process, but omitting the aforementioned weighting scheme in the histograms computation.

The results of this study in the GTEA Gaze dataset are presented in Figure 4.1-4 (a), which shows the Average Precision (AP) achieved by each approach at various vocabulary sizes. As one can notice from the results, for almost every technique, the performance improves until a vocabulary size of words, after which it stabilizes. Hence, from now on, we will either remove larger vocabulary sizes from our experiments or simply consider the optimal vocabulary size of 4000 as the final approach.

Comparing the approaches, as we expected, the I-BoW constitutes the theoretical upper bound of the method. This is logic due to the use of the ground-truth bounding boxes that, although they do not correspond to the tight silhouette of the object of interest, always ensure correct localization. Furthermore, two of the basic techniques to compute the saliency masks (geometric and temporal) already achieve slightly better results than the reference B-BoW. This is a nice consequence of the use of saliency masks, even when not specific post-processing is applied to the maps. Furthermore, the fact that the geometric saliency map is the one that achieves the best results, let us conclude that compactness is even more important than localization precision for an object recognition task.

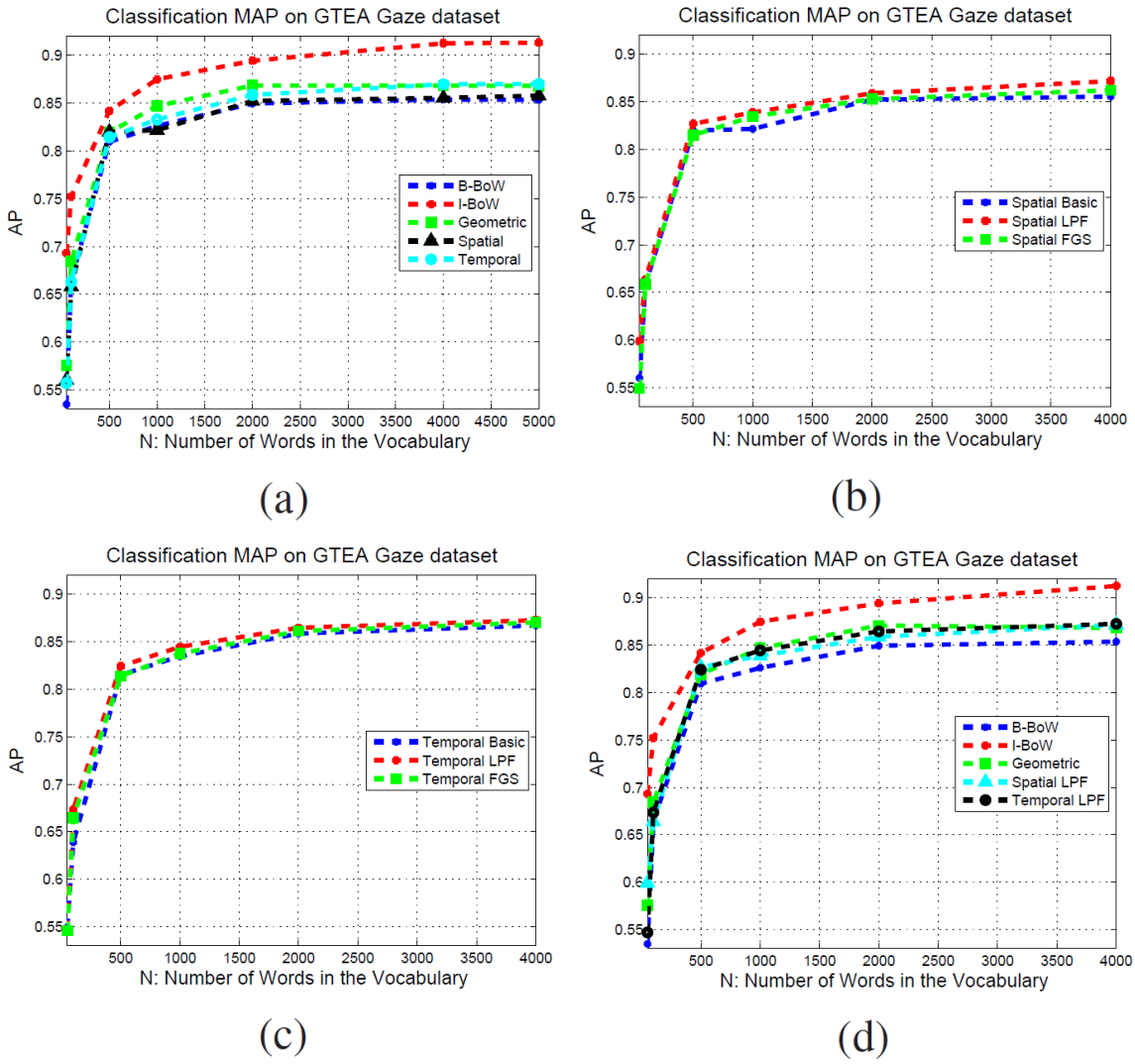

Figure 4.1-4: A comparison of various configurations in the GTEA Gaze dataset and various vocabulary sizes.

Figure 4.1-4 shows a comparison of various configurations in the GTEA Gaze dataset and various vocabulary sizes. In this figure, (a) is the results of the basic saliency techniques in comparison with the two references; (b) is the results achieved by two post-processing techniques for the spatial saliency; (c) is results achieved by two post-processing techniques for the temporal saliency; (d) is a comparison between the best post-processing option (LPF) and the reference methods.

Techniques for saliency map post-processing

In this section, we present the evaluation of the post-processing techniques described in section 4.1.2. As we have already claimed, direct outputs from some saliency detectors might not be optimal for an object recognition task due to the lack of compactness.

Since the Geometric technique already provided compact and Gaussian-shaped saliency masks, we have applied the post-processing stage to the spatial and temporal techniques. Figure 4.1-4(b) and Figure 4.1-4 (c) respectively compare the results obtained in the GTEA Gaze dataset by the basic spatial and temporal saliency, and the two post-processing methods: Low Pass Filtering (LPF) and Fitting of a Gaussian Surface (FGS). The improvements on the results, although not very notable, demonstrate that post-processing is important to adequate the saliency maps to the particular problem of object recognition. Furthermore, the computational cost of the LPF method, the one that achieves the best performance, is almost negligible when compared to other steps of the processing pipeline.

In addition, Figure 4.1-4 (d) shows a comparison between the LPF approach and the two reference methods. With the post-processing stage, now all the saliency methods outperform the reference BBoW and achieve closer results to the theoretical limit I-BoW. Hence, from now on, LPF post-processing will be incorporated to every version of our approach.

4.1.5 Fusion techniques for saliency maps

We have also evaluated the fusion approaches described in section 4.1.2. Results of this experiment are shown in Figure 4.1-5. All the fusion strategies achieve better results than the basic approaches except for the multiplicative technique. The rationale behind is that this strategy is too restrictive and requires all basic saliency maps to show significant values in order to consider a pixel as salient. The square fusion strategy obtains particularly good performance on this dataset, outperforming both the basic saliency approaches and the rest of the fusion strategies. In particular, by using this approach we are achieving absolute gains with respect to the reference B-BoW of a 3.1% and 2.7%, for a vocabulary of size 1000 and 4000, respectively. Hence, we will consider this fusion strategy as the final choice for our object recognition system in ego-centric videos.

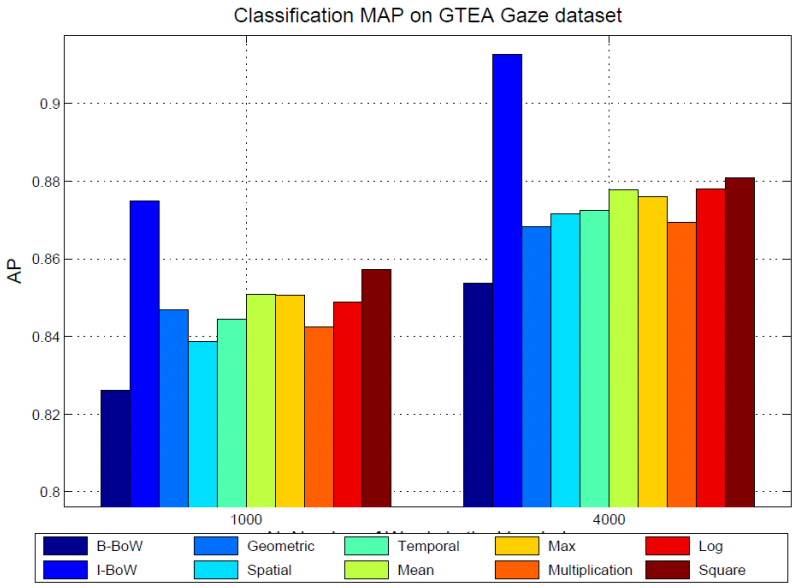

Figure 4.1-5: Classification results of various strategies for fusing spatio-temporal saliency maps.

Figure 4.1-5 shows the classification results of various strategies for fusing spatio-temporal saliency maps. In this figure, values are given at two different vocabulary sizes. Basic and reference methods are also included for comparison.

Algorithm	Cons. mAP ± std	Uncons. mAP ± std
B-BoW	0.585 ± 0.258	0.113 ± 0.152
I-BoW	0.621 ± 0.250	0.191 ± 0.258
DPM [16]	0.341 ± 0.254	0.129 ± 0.194
Proposal	0.602 ± 0.260	0.125 ± 0.167

Table 4.1-2: mAP and standard deviation on ADL dataset under the constrained and unconstrained scenarios.

Experiments under the constrained scenario

As we mentioned before, the constrained scenario is that one in which all the subjects wearing cameras are recorded in the same environment and interacting with the same object instances.

Results for the ADL dataset under the constrained scenario are shown in the first column of Table 4.1-2 in terms of mAP (mean Average Precision), and its standard deviation (category deviation). It is worth noting that we show only the results of those objects considered as 'active' in the dataset ground-truth annotations, e.g. those objects that are either manipulated or observed by the main actor in the ego-centric video. We consider these objects as the main source of information for detecting an action, so that the rest of the visual information (background) is less relevant and only useful for horizontal tasks as context identification.

As we have already mentioned, to simulate the constrained environment, we have randomly divided the whole set of frames into a training and test set (50/50%) without taking into

account the video to which each frame belongs. In this dataset, we are comparing the performance of our approach with the reference method B−BoW, the ideal case I−BoW, and the Discriminatively Trained Part-Based Model (DPM) [4.1.14], which was the approach used by the authors of the dataset [4.1.25] to address the object recognition task.

Furthermore, in Figure 4.1-6 we include detailed per-category performance. Base on these results, we can draw the following conclusions:

> Our proposal outperforms the reference B-BoW by guiding the recognition process to the salient areas of each frame. This result is consistent along almost all the categories in the dataset, and supports the idea that using visual saliency generates more accurate object representations and reduces the effect of clutter.
>
> The approach using ideal masks is, as expected, the one yielding the best performance. However, a deeper by category analysis shows remarkable conclusions: in general, providing an accurate localization of the object (I-BoW) helps the recognition process and improves the performance. This observation is particularly noticeable for relatively small objects such as the ones belonging to the categories 'food/snack', 'knife/spoon/fork', 'milk/juice' or TV. However, when the objects are too small, such as the instances of 'comb', 'dentfloss' or 'pills', we have observed that the ground truth bounding boxes, restricted to the object and lacking any information about object context, give not enough information to successfully detect its presence. In contrast, due to the fact that the saliency maps usually cover more area in the image (object, hands, even spatial neighboring context), our proposal achieves notably better results than the I-BoW. In addition, the reference B-BoW also achieves better results than I-BoW for these classes, although its performance is still below our approach.
>
> The performance of the DPM is poor when compared to any BoW method. From our point of view, the rationale behind is that this method has been designed to get good generalizations of object categories, what prevents from taking advantage of the high visual similarity between training and test samples in the constrained scenario. Hence, we believe that its relative performance with respect to our approach should drastically improve in the unconstrained scenario.

In addition, we have also evaluated our approach in the GTEA dataset. This dataset represents the constrained scenario in a more realistic way, due to the fact that we can take training and test samples from different videos. Hence, we have followed the same evaluation setup proposed by the authors [4.1.13]. In particular, we have developed a multiclass classifier so that each image is considered to contain just one object of interest. Our proposal achieves a global classification accuracy of 36.8% in this dataset, which compares well with the 35% obtained by the authors of the dataset [4.1.13] when they matched the highest detection score to the ground truth annotations.

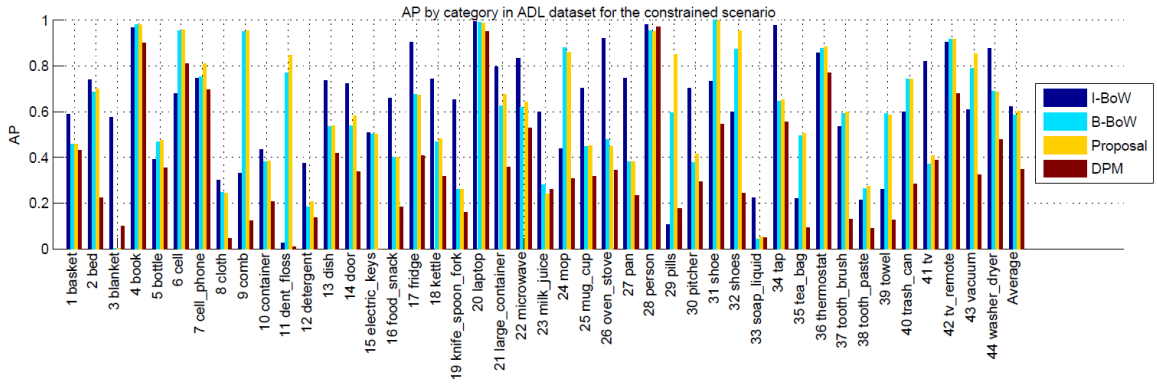

Figure 4.1-6: Per-category results (AP) for the constrained scenario achieved by various methods in the ADL dataset

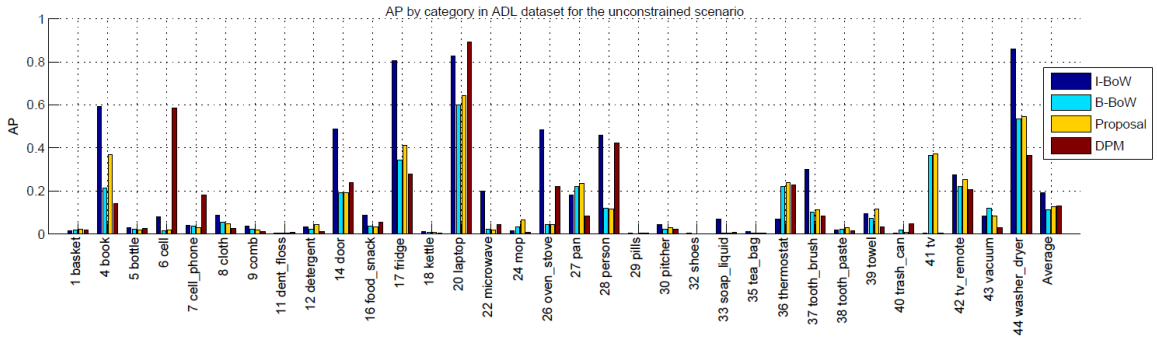

Figure 4.1-7: Per-category results (AP) for the unconstrained scenario achieved by various methods in the ADL dataset.

Figure 4.1-7 shows Per-category results (AP) for the unconstrained scenario achieved by various methods in the ADL dataset. Some categories cannot be computed in this scenarios due to the lack of samples in training/test sets.

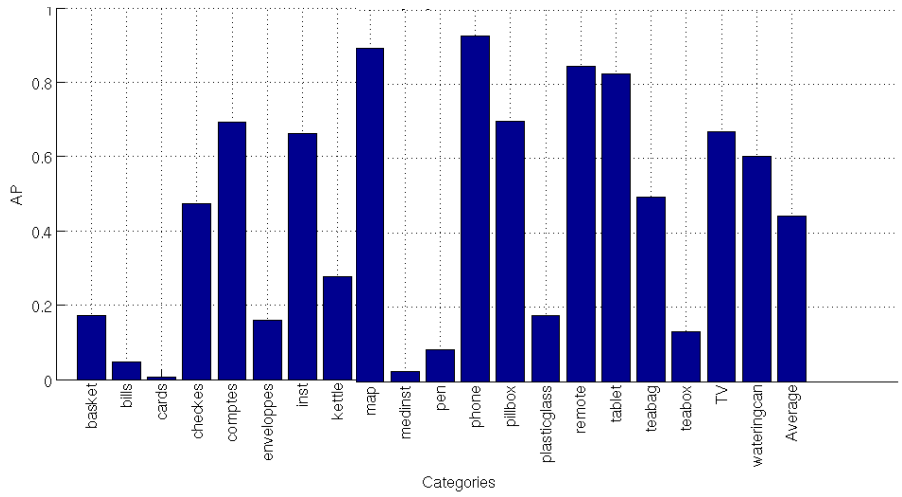

Figure 4.1-8: Per-category results (AP) on the Dem@Care Dataset

The results on Dem@Care dataset are given in figure Figure 4.1-8. Only our saliency based BOW approach has been benchmarked so far.

On this figure can be found the average precision for every categories along with the mean average precision (MAP, bar on the top right). On one hand, one can notice the AP for more than half the categories is rather high (>0.6) meaning the classification of active objects performs well. On the other hand, some categories give a low score for the AP such as medinst, pen, cards. One reason for some categories such as cards is because of the too low number of actual frames annotated (only 24 for cards right now).Another reason we found explaining low APs for some categories is the fact that the classifier recognizes an instance in an image while this instance is not considered as active.

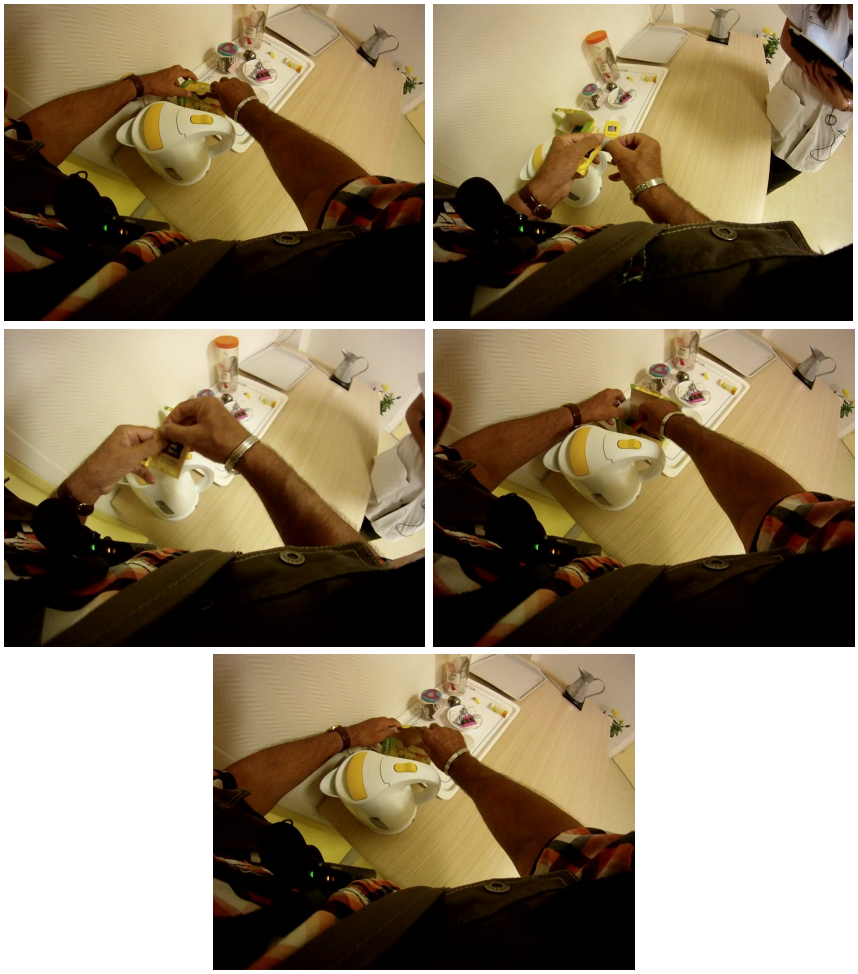

Figure 4.1-9: Five wrong classification of the object « kettle ». Even if the kettle is in the pictures, it is not an active object.

A good example is shown by Figure 4.1-9 showing five wrong classification of the object "kettle": On this figure, it can be seen that the object "kettle" is indeed present in the five returned images from the final classifier however the classification is returned as wrong. Indeed the problem here is that the kettle is not defined as an active object even though it is present on the image. As explained earlier, the task of classifying active objects is a very complex and subjective one. Here for example the AP would be greatly improved if the kettle was an active object.

Experiments under the unconstrained scenario

The unconstrained scenario corresponds to the challenging situation in which users perform their activities at several locations, thus interacting with heterogeneous instances of the object categories. Consequently, the large intra-class variation jointly with the reduced number of object instances, are expected to lead to poor generalization in recognition process.

In our experiments, we have used the videos corresponding to half of the subjects {2, 3, 5, 7, 8, 12, 13, 14, 17, 18} for training, and the remainder videos for test.

Average results of this study are shown in the second column of Table 3.1-1, whereas Figure 4.1-7 shows detailed per-category AP. We next draw the main conclusions of this experiment:

As expected due to the challenging nature of this scenario, the performance is drastically lower for all the automatic approaches (from AP ~ 0.6 to AP ~ 0.10). This illustrates how challenging is the problem of object recognition when just a few instance are available for each object.

Furthermore, the I-BoW, that uses ground-truth masks in test, now notably outperforms any automatic approach. This fact stresses the importance of a good previous localization of the object of interest for its localization.

Our proposal again outperforms the basic reference (B-BoW). The improvement is once more consistent along almost all the categories.

The DPM now achieves competitive results, even slightly superior to the ones of our proposal. As we previously stated, this technique learns object models with a high degree of generalization, which is better suited for this unconstrained rather than for the constrained scenario.

A study of the computational time

In Table 4.1-2, we show a comparison between the average execution times of our proposal and the DPM to run one category object detector in a test frame. We included results using a single threading (S.T.) and multi-threading in a 2.10GHz computer with 4 cores, and hyper-threading enabled.

For our proposal, the execution time comprises the generation of the saliency maps, the SURF feature extraction process, the computation of the weighted histograms, and the classification using a SVM with kernel. It is worth noting that some of the computations for the spatial saliency map are implemented in GPU so that they cannot be translated to S.T. case (spatial saliency takes about 0.05 sec per frame in the GPU). The rest of the calculus is made with the CPU under the aforementioned circumstances.

For the DPM, we run the implementation in [4.1.14], made in Matlab with optimized c routines for all the steps in the process that require most of the execution time.

As we can see in the tables, our approach shows much lower computational times in comparison with DPM. From our point of view, the rationale behind is the fact that using the saliency maps, we avoid the heavy scanning process of a sliding window approach as the DPM.

Furthermore, it is also worth noting that, since the saliency maps are automatically computed in both training and test data, our method does not need bounding boxes for training, what dramatically reduces the human resources devoted to the database annotation when compared to the DPM.

Table 4.1-3: Test execution times of our approach compared with the DPM implementation in [4.1.14].

Algorithm	S.T.	M.T.
DPM [16]	60.4s	10.9s
Proposal	15.7s	4.1s

Table 4.1-3 shows the test execution times of our approach compared with the DPM implementation in [4.1.14]. In this figure, single threading (S.T.) and multi-threading (M.T.) execution time are shown.

4.1.6 Conclusions

Throughout this work we came up with a method for object recognition in egocentric videos. Our proposal aims to drive the recognition process using visual saliency. In particular, spatial, temporal and geometric cues found in egocentric videos are exploited to improve the object recognition, generating more precise representations of the area of interest in a frame, as well as enhancing the robustness against cluttered backgrounds.

We have also evaluated several fusion strategies to generate spatiotemporal-geometric saliency maps from their basic constituents, as well as some post-processing techniques that improve the compactness, a property that has turned out to be very important for object recognition.

In addition, rather than simply performing foreground/background segmentation to restrict the recognition process to the areas of interest, we have proposed a soft application of saliency that controls the influence of pixels in the final object representation based on their saliency. We have combined saliency with the well known Bag of Words paradigm by proposing a saliency weighting method to compute image signatures.

Having in mind the context of this work, which is the automatic analysis of videos for the diagnosis, assessment, maintenance and promotion of self-independence of people with dementia, we have assessed our model in two particular scenarios of interest: a) a constrained scenario in all the subjects perform actions in the same room and, therefore, interact with the same object instances, and b) an unconstrained scenario that corresponds to recordings made at different locations, so that users interact with various instances of the same objects.

Our experiments have shown that this method outperforms the basic BoW model and achieves closer results to a hypothetical case in which optimal foreground masks are available in test. Furthermore, our approach compares well, and outperforms DPM and the full method in [4.1.13] under the constrained scenario.

Furthermore, the computational time is less than half of the DPM one. However, the notable decrease in performance in case of an unconstrained scenario reveals that our method needs further development. Indeed, in an unconstrained scenario the variability of object instances intra-category requires drastically new recognition approaches. Here we are in the case of "concept recognition". As we know from e.g. TRECVID challenge [4.1.24] concept recognition is a complex and open research problem.

4.2 Description based approach for Activity Recognition of older People using an RGB-D Camera

4.2.1 Introduction

Activity recognition approaches can be categorized according to the input features they use and the reasoning methods they apply for [4.2.1]. Concerning the abstraction level of input features used to construct activities, pixel-based (low-level) and object-based (high-level) approaches are the two main approaches. Pixel based approaches are using for instance colors or textures like in [4.2.2] while the other category builds an abstraction of the low-level data as objects including inherent properties (e.g., speed, trajectory) [4.2.3].

The reasoning methods applied on activity recognition can be classified into three main categories: classification methods (e.g., SVM) [4.2.4], probabilistic graphical models (e.g., HMM) [4.2.5] and semantic models (e.g., description based models) [4.2.6]. Sadanand et al. [4.2.7] have proposed a classification method using for activity recognition, where each action (e.g., boxing, diving) is represented by a set of examples on different scales, viewpoint and time-resolution. A set of detectors is used for each action class, and the output of all action detectors is then combined using a Support-Vector Machine approach. This method outperforms most of state of the art methods on benchmark datasets. However, classification methods and probabilistic graphical models are generally based on low-level data (e.g., pixel-based, feature-based) and on a training procedure involving a large dataset to be able to generalize among the activities performed on different scenarios. It is difficult to foresee the behavior/performance of this algorithm when applied for a different environment from the one of the training.

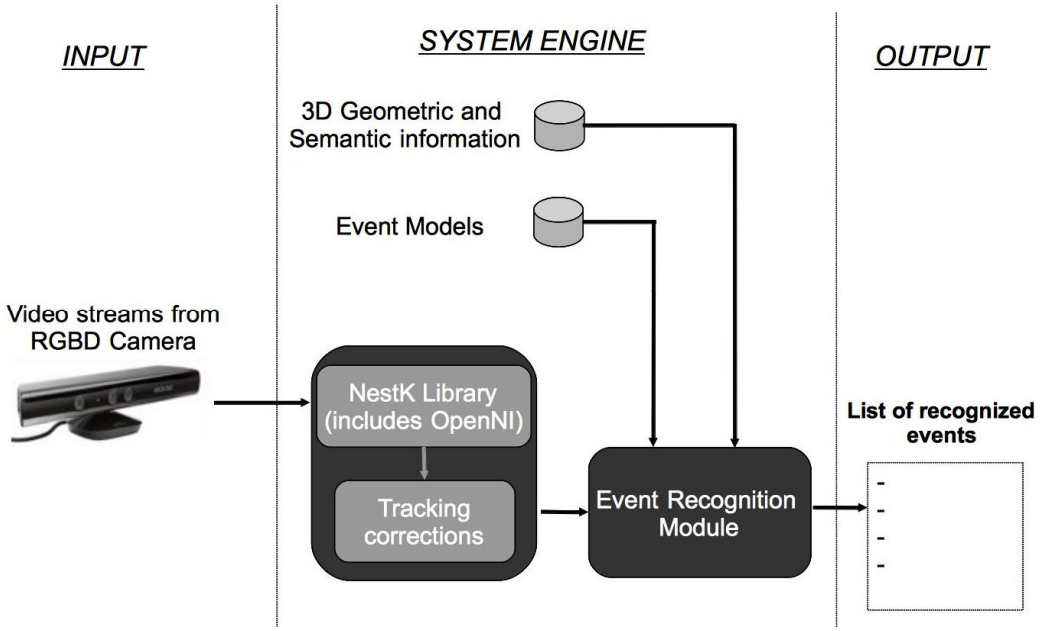

Figure 4.2-1: System architecture

The use of semantic models is an alternative approach as it does not require learning but a set of activity models provided by domain expert, for instance based on logics or grammars rules. An example of such approach has been described by [4.2.8]. They evaluate the detection of complex activities by constructing a tree composed of the related sub activities. The major limitation relies on how to assign, for example, a sub activity to one of two complex activities when it cannot be part of both, in the presence of high level noise.

Most of the work presented previously uses RGB cameras. Nevertheless, the use of RGB-D cameras is growing in the domain of activity recognition as recently they have become more affordable, they can provide real 3D information of the scene and ease the deployment of the system to new environment. Banerjee et al. [4.2.9] have developed a system for fall detection in hospital rooms using RGB-D camera and a fuzzy inference system. The system infers facts using approximate descriptions of the world.

Pramerdorfer [4.2.10] has evaluated RGB-D camera (Kinect, Microsoft) concerning its suitability and robustness for people and fall detection systems, with respect to particular conditions like distance from camera, illumination or clothing materials and color. For instance, it has been shown that clothing colors can be an issue for people detection and tracking and need to be considered.

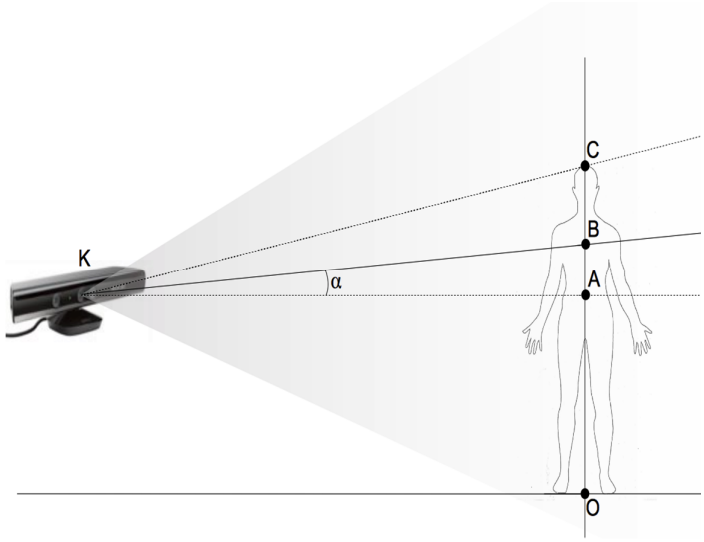

Figure 4.2-2: Height computation

Our contributions in this work are two-fold:

An evaluation framework for mid to long-term activity recognition using hierarchical model-based approach combined with a RGB-D camera,

A set of people detection and tracking techniques is proposed to improve the robustness of the proposed approach:

An alternative method to re-compute the height of a person in case of a partial loss of body area detection cause by the issues described by Pramerdorfer [4.2.10] and also observed in our activity dataset.

The use of a re-identification algorithm [4.2.11] to improve Person tracking in cases where the person leaves the scene and come back.

4.2.2 Proposed approach

The proposed system is presented in two main subsections. The first subsection describes the different issues encountered with people detection and tracking and the proposed solutions. The activity detection module is presented in the second subsection, with the detailed description of the information needed by the system. For more information on the system architecture, see Figure 4.2-1.

Vision component

The first layer of the vision component performs people detection and tracking based on the open source framework OpenNI, through NestK library. The second layer is composed of the proposed solution to cope with poor estimation of person height false positive detections of people, and maintenance of people identification.

Height Computation and Posture Recognition

In some cases, infrared rays are absorbed when the tracked person is wearing black clothes. The consequence is that some parts of the body (generally lower body parts) are excluded from the 3D bounding box, which leads to a smaller estimation of height for the tracked person. This is an important issue since the posture is herein mainly inferred from the estimated height of the person. To cope with the presented problem, we propose to compute the height of a person based on the top point of the person (highest point of the person's point cloud), the angle of the camera from the horizontal position and the distance between the base of the device and the ground (see Figure 4.2-2). So, the system computes the new height H as follows:

$$H = \| OC \|$$

$$= \| OA + AB + BC \|$$

$$= OA \cdot \cos(OA, AB) + AB \cdot \cos(AB, BC) + BC \cdot \cos(BC, OA)$$

where
:

OA : the a priori known distance between Kinect and the ground

$AB = \sin(\alpha) \cdot \| KA \|$

BC : the distance from Kinect to the top point of the person (vertical coordinate of this point)

$\cos(u, v) \in \{-1, 1\}$, where $u, v \in \{OA, AB, BC\}$, depending on the Kinect position and angle.

This last formula is valid for most of the common situations (any value for α or any height for Kinect position).

The computed height is used to detect whether the person is sitting or standing based on a thresholding method. We take into account the average height for sitting and the person is considered standing when they height value is above the sitting average height.

Object Detection Filtering.

Objects, furniture and walls are sometimes detected as a person by the OpenNI tracking algorithm, therefore generating unreliable activities. To avoid it, we implement a filter which removes from the list of detected People, those which are not inside the expected range size of a human being.

Reidentification.

In order to detect activities, the system needs knowledge of past activities related to a person. Nevertheless, the detection and tracking algorithm creates sometimes a new identifier for a person that has already been tracked earlier. This happens for instance when a person leaves and comes back in the field of view of the camera. To fix this issue, the system uses a reidentification algorithm that computes highly discriminative signature based on covariance matrix [4.2.11]. This process enables the system to avoid considering a single person as two different individuals and to keep a continuous tracking on him/her. This algorithm provides the desired robustness to people tracking although it poses a restriction on the application of the proposed approach for a real-time situation.

Activity Recognition Framework

The description of activity models is defined using a declarative language [4.2.12][4.2.6]. This language is affordable by expert since it uses a proper structure and explicit key words. Activity Models are composed of six components:

Physical Objects: objects involved in the recognition of the activity modelled (e.g., person or spatial zone),

Components: sub-activities that the model is composed of,

Forbidden Components: activities that should not occur in case of the activity model is recognized,

Constraints: conditions that the physical objects and/or the components should hold,

Alert: importance level of the scenario model in terms of priority,

Action: in association with the Alert type, specific action which would be performed when an activity of the model is recognized (e.g. send a SMS to a carer),

The main types of physical objects of a model are: person, zones and equipments. A person is a object dynamically detected, and it has a set of attributes (e.g., x-y-z 3D coordinates, width, height, and depth). Zones and equipments are static object which are *a priori* defined to the processing the activity, and they refer to contextual information on the scene (e.g., spatial zones of interest which contains semantic information in regard to the activity to detect).. Constraints define conditions that physical object property(ies) should meet, or components

should hold. They could be non-temporal, such as spatial and appearance constraints, or temporal, such as, Person_in_zone1 before Person_in_zone2. Temporal constraints are defined using Allen\D5s interval algebra (e.g., BEFORE, MEET, and AND) [4.2.13]. This implies to know a priori to the processing the activity we want to recognize and the contextual information on the scene (e.g., spatial zones of interest which contains semantic information in regard to the activity to detect). Activity models are hierarchically categorized according to their complexity (ascending order):

Primitive State: instantaneous value of a property of a physical object (e.g., Sitting or Inside_zone_couch)

Composite State: composition of two or more primitive states

Primitive: change in a value of a physical object property (e.g., Person changes from Sitting to Standing posture)

Composite: composition of two previous models.

Here is an example of the definition of a complex model with its sub events :

```
Composite (Sitting_in_couch,
        PhysicalObjects ((p1 : Person), (z1 : Zone))
        Components ((c1 : Person_sitting (p1))
                        (c2 : Person_inside_zone_couch (p1, z1)))
        Constraints ((c1 and c2))
        Alarm (URGENT))
```

```
PrimitiveState(Person_inside_zone_couch,
        PhysicalObjects ((p1 : Person), (z1 : Zone))
        Constraints ((p1 -> Position in z1 -> Vertices)
                        (z1 -> name = zone_couch))
Alarm (NOTURGENT))
```

To summarize, the extraction of complex events from video sequences is performed by a combination of the RGB-D data stream, the corresponding tracking information (delivered mainly by the libraries NestK and OpenNI), the contextual information (zones or objects of interests) and the event models.

4.2.3 Evaluation

The evaluation of the activity description based approach using RGB-D camera is divided into four main parts:

A posture recognition evaluation to assess the improvement brought by the proposed techniques on the height computation,

A performance comparison between the proposed activity recognition system and the system using only NestK people tracking functionalities (without the proposed improvements)

A performance comparison with a system following the same description based approach but using a RGB camera (AXIS, Model P1346). The reference system uses image segmentation algorithm proposed by [4.2.14] and the people tracking algorithm proposed by [4.2.15].

A complementary evaluation of the assessed activity duration compared to the real activity duration provided by the ground truth.

Dataset

The proposed system has been evaluated at monitoring the physical tests of participants of a medical protocol for Alzheimer's disease study. Participants are asked to perform a set of physical activities and daily living activities as basis to a clinical evaluation of their executive functions. The protocol is divided into three scenarios: directed activities, semi-directed activities and undirected activities. Scenario 01 (S1) or Directed activities is intended to assess kinematic parameters about the participant's gait profile (e.g., static and dynamic balance test, walking test). During this scenario an assessor stays with the participant inside the room and asks him/her to perform mainly four physical activities within 10 minutes (divided in sub activities).The RGB-D camera recordings are acquired at a frame rate of 10 frames per seconds with an angle of view of 57 degrees horizontally and 43 degrees vertically (because of the motorized tilt). These activities are briefly described as follows:

Balance test: the participant should keep balance while performing exercises (e.g., standing with feet side by side or standing on one or the other foot)

Walking test: the assessor asks the participant to walk through the room, following a straight path from one side of the room to another (go attempt, four meters), and then to return (return attempt, four meters)

Repeated transfer test: The assessor asks the participant to make the first posture transfer (from sitting to standing posture) without using help of his/her arms. The examiner will then ask the participant to repeat the same action five times in a row

Up & go test: participants start from the sitting position, and at the assessor's signal he/she needs to stand up, to walk a three meters path, to make
a U-turn in the center of the room, return and sit down again
.

Performance Evaluation

The system is evaluated compared to activity annotation provided by domain expert. The following indices are computed:

Precision = TP / (TP + FP),

Recall = TP / (TP + FN),

F-Score = 2 Precision * Recall / (Precision + Recall)

where:

TP: true positive, FP: false positive, FN: false negative. The evaluation is performed by taking into account the number of detected activities.

4.2.4 Results and Discussion

The following results refer to 30 videos with average time length of 6.9 minutes. Table 4.2-1 presents the activity recognition system performance at posture recognition. A recall of 100% is achieved (out of 190 activities, 107 for standing posture and 83 for sitting posture).

Table 4.2-1: Posture recognition performance

	Recall (%)	Precision (%)	F-Score (%)
Sitting	100	75.5	86.0
Standing	100	89.2	94.3
Total	**100**	**82.6**	**90.5**

Table 4.2-2 shows the differences obtained for complex activities recognition with and without the proposed improvements for the vision component. Results obtained directly from NestK people detection output are presented on the left, while the results obtained from the proposed system are on the right (with improvements).

Table 4.2-2: Event recognition performance with the proposed vision component improvements. Total number of events to detect: 150 (1 event of each category per video)

	Only NestK		NestK + Improvements	
Event category	Recall (%)	Precision (%)	Recall (%)	Precision (%)
Balance test	90.0	96.4	100	100
Walking test (go attempt)	93.3	93.3	100	90.9
Walking test (return attempt)	**73.3**	95.7	**90**	100
Repeated transfer test	96.7	**60.4**	100	**90.9**
Up & go test	80.0	85.7	93.3	90.3
Total	**86.7**	**82.8**	**96.6**	**94.2**
Global F-Score	**84.7**		**95.4**	

The observed gain of performance of the proposed approach is approximately of 10% for precision, recall and F-Score. On improved version, a recall of approximately 97% is obtained on the overall activities (true positive rate) while the precision is close to 94%. This fact means that the system recognizes most of the activities from the video sequence (around 3% missed) with an acceptable amount of false positive activities. For the repeated transfer test, we highlight that the improvement of the height computation of the person has improved the precision of the detection of this activity, directly related to posture (from 60.4% to 90.9%). Concerning the return attempt of the Walking test, its detection is mainly improved by the use of the re-identification algorithm and the improvement of the recognition of the go attempt.

Table 4.2-3 compares the results obtained with a RGB-D camera (on the right) and with a RGB camera (on the left). RGB-D camera results refer to the improved version on Table 4.2-2. The proposed approach has a higher recall (less false negatives) than RGB camera one, but a lower precision (more false positives). This means that the system using real 3D information obtain a lower rate of wrongly detected activities. In total, RGB-D camera improves the global recognition of the system of around 2.5% (F-Score).

Table 4.2-3: Comparison between the event recognition performances of the system using RGB and RGB-D cameras

Camera	RGB camera		RGB-D camera	
Event category	Recall (%)	Precision (%)	Recall (%)	Precision (%)
Balance test	95.8	95.8	100	100
Walking test (go attempt)	91.7	100	100	90.9
Walking test (return attempt)	87.5	95.5	90	100
Repeated transfer test	75.0	100	100	90.9
Up & go test	91.7	100	93.3	90.3
Total	**88.3**	**98.3**	**96.6**	**94.2**
Global F-Score	**93.0**		**95.4**	

The previous tables have presented the evaluation of the system in terms of activity frequency. Table 4.2-4 shows an evaluation of the proposed approach in terms of assessed duration of a given event compared to the real event duration annotated on the ground truth. The recall value of the assessed duration is close to the one obtained in terms of event frequency, matching the real duration of event. On the other hand, the precision is lower. This lower value is due to the fact that RGB-D camera (Kinect) field of view do not cover the whole scene where the physical tests have been undertaken. Therefore, in the current system, the time spent by the person performing a physical test outside the field of view (Walking tests and Up & Go test) is not taken into account.

Table 4.2-4 Comparison between the event recognition performances of the system using RGB and RGB-D cameras in terms of assessed duration of a given activity

	Recall (%)	Precision (%)
Balance test	99.9	70.8
Walking test (go attempt)	55.2	94.1
Walking test (return attempt)	60.1	62.6
Repeated transfer test	86.6	94.6
Up & go test	79.8	86.2
Total	**94.5**	**73.6**
Global F-Score	**82.8**	

4.2.5 Conclusion

We have presented an activity recognition framework using RGB-D camera based on hierarchical descriptive models. While RGB cameras can be calibrated to obtain a 3D estimation of the scene, the use of RGB-D camera provides real 3D information of the scene, which tends to be more reliable. Besides the affordability of the nowadays RGB-D cameras and the robustness of their 3D information, the use of a description based language allows us

to easily adapt the event models to new environments. Moreover, the proposed improvements for the vision components such as the re-identification algorithm enable the proposed approach to achieve the desired robustness. Finally, while most of the computer vision algorithms are developed to work on short video clips and for short terms activities, we are more focused on detecting mid to long term activities (e.g., walking test) for long term monitoring (e.g., weeks or months) in order to track people habits directly at home and therefore detect any health state change.

The presented framework is applicable for both short term and long term activity recognition. The short term activity recognition (*e.g.,* primitive states, primitive events) encompass posture recognition, person localisation in contextual zones, and person interaction with contextual objects.

In the context of Dem@Care project, the proposed activity recognition framework is able to run online, and it has been integrated into the first pilot of the project. Although we have presented results for long term activities (complex activities), these results are only illustrative of the framework performance. Long term activity recognition is going to be presented in details in WP5, and it will be based on the combination of the short term activities detected in WP4.

This work is under review for publication in the International Conference on Computer Vision Systems 2013.

We plan to evaluate the fusion of multiple RGB-D cameras to cope with the restricted field of view of a single RGB-D camera when compared to an ambient camera.

5 Life-logging

5.1 Introduction

Lifelogging is the capture and analysis of any data sources for the purposes of recording the events and patterns of a person's life. It is an important aspect of the Dem@Care system for a number of reasons, covering the clinical and technical elements of the project.

Lifelogging is already widely used in medical and therapeutic applications [5.2.1] such as reminiscence therapy and memory reinforcement. In reminiscence therapy, participants are shown items from earlier periods in their life, which helps to evoke memories and provides cognitive stimulus, as well as having social benefits like encouraging conversation and story-telling.

The aim of task T4.4 is to develop the technology necessary to build a lifelog and to make it searchable and browsable by means of constructing an automatic index. This index will use the daily activities of the participant as its basic unit. Thus, in this task, research is concentrate on two challenges:

Event Segmentation: the identification of each episode of behaviour, i.e., a continuous period of time where the participant is engaged in a single activity. By identifying the points in time where the primary activity changes (i.e., the boundary between events) we can delineate the periods of time where a single event started and stopped.

Event Identification: having determined the time periods where a single activity is underway, these periods need to be classified, or labelled, with the identity of the activity taking place (e.g., walking, sleeping, meal-time, on a bus, etc.).

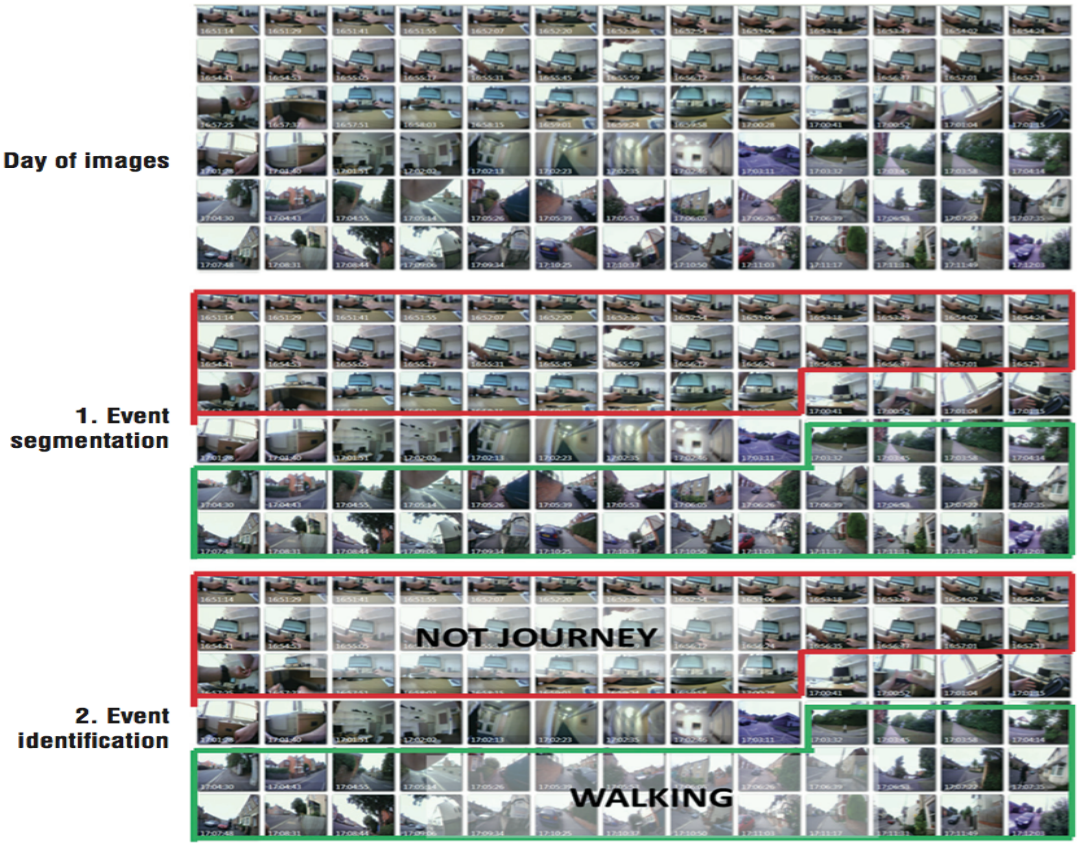

Figure 5.1-1 Segmentation and classification of lifelog events

This two-step process is illustrated in Figure 5.1-1 where a stream of photos is first segmented into distinct sets using an event segmentation algorithm, and then each set of photos is labelled with the name of the event or the activity that is taking place.

The aim of automatic event detection is to determine boundaries that signify a transition between different activities of the subject. For example if the subject was working in front of his computer and then goes to a meeting, or was watching TV and then goes to prepare a meal, we believe it will be desirable to automatically detect the boundary between the segment of images of him working at the computer, and the segment of images of him being at a meeting. In essence the aim is to detect moments of change, whether they be visual, sensory, or otherwise.

We use the terms Event Segmentation and Event Identification in lifelogging (instead of the more specific Activity Recognition used elsewhere in this deliverable) because in lifelogging we are concerned primarily with classifying a given instance in time in order to identify the type of behaviour and/or event underway.

5.2 Motivations

The memory reinforcement aspects of lifelogging relate to the participants reviewing their lifelogs on a daily and weekly basis, for example, describing the events and activities encapsulated there.

As well as the benefits to the PwD, the lifelog could also be used by the clinician and the care-giver to verify that instructions and directions are being followed appropriately. For example, the lifelog could be used to ascertain that eating activities are occurring at regular intervals, appropriate to the dietary requirements of the PwD. Similar functionality could be used to monitor sleep, exercise, social contact and so on.

In the following sections, we describe the background theory and the current state of the art for life-logging, data capture, event segmentation and event identification. We then describe how lifelogging will be implemented in Dem@Care, showing the work already developed and stating our plans for future development and research.

5.3 Lifelog data capture

The capture of data for lifelogging has been undertaken by many researchers for some time. Steve Mann is a pioneer who tried to capture what he saw through video cameras mounted on his head and these have evolved from head-mounted cameras to discreet recorders built into eyeglasses [5.3.1]. Microsoft Research in Cambridge have used the SenseCam to capture everyday life and have evidence that these images can improve peoples' memory abilities [5.3.2]. In MIT, an experiment was carried out using Bluetooth-enabled mobile telephones to measure information context in order to identify the deep social patterns in user activities [5.3.3]. In [5.3.4], the authors presented a memory re-finding use of lifelogging which is called "iRemember". In their research, they recorded audio clips as the main information used to navigate memory. In [5.3.5], this kind of technology is also employed to provide real-time transportation information to individuals with mild cognitive disabilities and improve efficiency and safety as well. Mobile phones and other kinds of digital devices are very popular nowadays and form a large computing resource and a ubiquitous infrastructure for our digital life. The DietSense project [5.3.6] at UCLA makes use of a mobile phone with a camera embedded to capture pictures automatically. The images collected as the log of a wearer's mealtimes are used to analyze the diet intake in order to give feedback and to improve diet choices. The WayMarkr project [5.3.7] also makes use of a mobile phone affixed to a strap to take pictures automatically. Furthermore, social dynamics are studied in [5.3.3] by using mobile Bluetooth as the measure in lifelogging.

Data capture technology is still under active development. For instance, Google glass is a wearable computer with a head-mounted display that displays information in a smartphone-like hands-free format [5.3.8]. The user can interact with the glass via natural language voice commands. For instance, the wearer can ask the glass to take a picture or record a video. This device supposes to be in the market soon. Another lifelogging tool that came to the market recently is a wearable camera called Memoto [5.3.9]. Memoto is a small camera that takes a photo every 30 seconds and automatically uploads it to an online service.

When capturing data, the data set can be annotated, which is helpful especially for the use of supervised methods. Each observation in an annotated data set uses a set of predefined annotated variables. These observations can be created by assigning a label to a set of values.

It is possible to do the annotation while capturing the data by asking the participant to perform a set of activities, and then label the data based on the performed activity. This method has been used in tons of studies, where a researcher supervises and observes the participant during the collecting process. For example, the authors in [5.3.10] did a study to detect activities of daily living using accelerometer data. The participants were asked to perform 20 different activities. Five accelerometers were placed at the upper arm, lower arm, hip, thigh and ankle. Data from the accelerometers was labelled during the collection process based on the performed activity, and then used to train a number of classifiers including the C4.5 Decision Tree, Decision Tables, Naive Bayes and nearest neighbour classifier.

Another approach for annotating the data set is to use manual annotation protocols. In this approach a vocabulary for annotation will be created, and then the user can look over at the data, using data annotation & processing tool, and annotate the metadata based on the controlled vocabulary. For instance, a study uses manual annotation protocol was done by Kerr et.al [5.2.1], to assess sedentary behaviour relaying on accelerometers data and SenseCam images. The data was collected by 40 users, and images were coded later for sitting and standing posture and 12 activity types. The coded image data were then compared to the accelerometer data. The authors reported that manual coding of the images was time-consuming and coding errors can occur.

Although they are successful in solving some design considerations, the algorithms for detecting contexts and situations lacks flexibility; this means the systems cannot adapt to the semantics of contexts. Context information is not fully used to receive more flexible approaches of context classification and recognition for labelling the semantic meaning of the user events.

The size and scope of this research shows that there is a very active community in lifelogging, exploring a range of techniques and using a variety of lifelogging devices. Yet lifelogging research needs to address more than just the data capture technology; it needs to also investigate and create new techniques for the analysis of lifelogs and to provide search, browsing, and navigation through the lifelogs. Thus indexing and retrieval are just as important as the lifelog capture devices.

In order to manage accumulated lifelogs we need good information management tools, and much related work has been done in multimedia retrieval where low-level feature-based multimedia queries using image features such as colour, texture, edges and other attributes have been studied extensively. However, there is no means to reflect the coincidence between features extracted from visual data and the interpretation that they have for the user in a given situation [5.3.11].

5.4 Event Segmentation and motivations

The authors in [5.4.1] describe a visual diary of lifelog images constructed by clustering images based on low-level image features such as colour spatiogram and block-based correlation between images. Their experiments also incorporate additional sensor data from accelerometers. These features allow the clustering of images into events, allowing the user to review their day by event, rather than a stream of images.

The study in [5.4.2] showed a method for detecting event boundaries, based on an adaptation of Hearst's TextTiling algorithm, in which images (or blocks of images) are compared to their

neighbours to determine their dissimilarity. The system identifies boundaries where the dissimilarity has exceeded a threshold. The paper investigates the optimal size for the block of images, the optimal distance metric to measure dissimilarity and the optimal threshold for successfully detecting boundaries.

In [5.4.3], the authors examine event-boundary detection using multimodal data. The data consists of images, accelerometer data, light sensor values, temperature and recorded audio. They also experiment with the fusion of these different data sets. From their results they were able to identify three main types of activity boundaries: 1) a change of activities within the same location 2) change of location 3) engagement in social interaction. They show that sensor accuracy is related to the type of activity boundary. For example, recorded audio is significantly better at identifying changes in social interaction, but not activity changes in the same location.

The authors in [5.4.4] developed a novelty detection algorithm based on identifying deviations from the wearer's normal behaviour. Daily activities are logged in the usual fashion. Sequences of similar images can be matched over days and weeks (e.g., the wearer's daily commute to work). Novelty is detected when a sequence of images cannot be matched to a previously recorded sequence. Assuming that the adjoining sequences do match previously recorded sequences, the novel sequence can be seen as a temporary deviation from the norm, and so the event can be emphasized as significant.

5.4.1 Event identification

Bridging the gaps between different levels of semantics is a challenge for researchers in content-based information retrieval. High-level features refer to features that are semantically meaningful for the end user. While low-level features are never readable by the end user, high-level features can express the semantics of media in a more acceptable way as concepts, such as 'indoor', 'outdoor', 'vegetation', 'computer screen', etc. These features can provide a meaningful link between low-level features, and user expectations. The extraction of high-level features demands filling the gap between low-level features and high-level features, which is called the semantic gap in multimedia retrieval.

Semantic concepts are usually automatically detected in a mathematical way by mapping low-level features to high-level features. The state-of-the-art approach is to apply discriminative machine learning algorithms such as Support Vector Machines (SVMs) to decide the most likely concepts given the extracted features [5.4.5]. Compared to a discriminative model which is more task-oriented, generative statistical models such as Markov model try to analyze the joint probability of variables, which are also proposed in concept annotations [5.4.6]. Both generative and discriminative approaches have their own pros and cons. A generative model is a full probabilistic model of all variables whereas a discriminative model has limited modeling capability. This is because a discriminative model provides a model only for the target variable(s) conditional on the observed variables hence cannot generally explain the more complex dynamics underlying the generation of data for a given class. However, discriminative models are often easier to learn and perform faster than generative models. Besides, it has been shown that discriminative classifiers often get better classification performance than generative classifiers with large training volume (usually including positive and negative samples).

A limitation for building classifiers is for them to reveal the higher-level semantics of images when they have multiple concepts with high correlation. The concepts involved in lifelogging cover numerous aspects of our daily lives and the choice of concepts is very broad.

Although individuals may have different contexts and personal characteristics, the common understanding of concepts that is already socially constructed and allows people to communicate according to [5.4.7] and [5.4.8], also makes it possible for users to choose suitable concepts relevant to activities.

In state-of-the-art everyday concept detection and validation [5.4.9], concepts are suggested by several SenseCam users after they have reviewed several days' worth of their own lifelogged events. The set of concepts used are those that can be detected with an accuracy rate above a particular threshold.

To find a set of candidate concepts related to each activity in a set of everyday activities, the studies in [5.4.10] [5.4.11] [5.4.12] carried out user experiments on concept selection where candidate concepts related to each of the activities above were pooled based on user investigation.

Byrne et al. used low-level features of SenseCam images to define high-level semantic concepts such as eating, road, sky, office, etc [5.4.13]. 27 semantic concepts have been defined and used as a source for improving the segmentation task. The everyday concept detection is composed of: supervised learning, visual feature extraction and feature and classifier fusion. The validation was done by 9 participants who manually judged the accuracy of the detection on a subset of 95,507 lifelog images. The results showed on average a precision of 57% for positive matches and 93% for negative matches within such a collection. The authors see these results as encouraging ones, and they suggested that automatic concept detection methods translate well to the domain of visual lifelogs.

Doherty [5.4.14] relied on SenseCam sensor readings and low-level features of images to create clusters of distinct activities throughout a day. The MPEG7 visual descriptors of color layout, scalable color, edge histogram, and color structure information for each image is extracted to give an indication of what image features can represent this image. SenseCam sensor readings (including accelerometer, ambient temperature, light level, and passive infrared detector) are then associated with each image based on time. The values of sensor readings and images features are normalized to ensure that they are all on the same scale for comparison. The adjacent image/sensor values are then compared against each other to determine how dissimilar they are. When the dissimilarity value is higher than a threshold value, a boundary for a new activity is considered.

To improve the content-based retrieval, Byrne et al. combined contextual sources, namely GPS measurements and MAC addresses of nearby Bluetooth devices, to the content sources to add more accuracy when defining and retrieving activities [5.4.15]. The feasibility of using GPS data and Bluetooth MAC addresses to improve retrieval of similar events was successfully tested. GPS offers a means of determining the location at which an activity occurs, while Bluetooth MAC addresses infer the presence of specific individuals in an event. Similar study was done by Kikhia et al. [5.4.16] that relies on GPS data and Bluetooth MAC addresses in segmenting the day into distinct activities based on visited places and met people. Known places to the user, such as home and working place, were defined using polygons based on GPS coordinates, while Bluetooth MAC addresses are used to detect the

presence of people. The system compares the logged GPS and Bluetooth MAC addresses during the day with the pre-defined data and then lists the day as activities based on places and persons. SenseCam images that correspond in time to each activity are associated with that activity. Periods of time when there is no known context, namely the user is in unknown place and no known person nearby, are also presented with the corresponding images giving the user the opportunity to review and adjust the data.

Accelerometer data has been used in many studies to provide recognition of everyday activities such as walking, running, sitting and lying [5.4.17] [5.4.18]. Many researchers utilized Machine-learning techniques to segment the day into activities based on Accelerometer data. Bao and Intille [5.4.19] placed five accelerometers at the upper arm, lower arm, hip, thigh and ankle. Features derived from both the time and frequency domain was extracted from the raw accelerometer data and used to train a number of classifiers including the C4.5 decision tree, decision tables, naive Bayes and nearest neighbor classifier. The authors succeeded in classifying 20 different activities with an accuracy of 86% using the decision tree classifier. Preece et al. [5.4.20] did a similar study to recognize activities including walking, going up and down the stairs, running, hopping on left or right leg and jumping based on wearable accelerometers. The highest activity recognition accuracy for a single sensor (97%) was achieved using ankle-mounted accelerometer. The authors in [5.4.20] also relied on features extracted from the raw accelerometer data namely the FFT component feature set.

Nugent et al. used Dempster-Shafer theory to assess the impact of sensor reliability on the classification of ADLs [5.4.21]. The authors used a smart environment equipped with binary sensors, which only present two possible values as outputs namely ‘0’ or ‘1’. The aim was to infer activities of making a hot drink (including making coffee and making tea) or making a cold drink. 7 binary sensors were included in the experiment namely Fridge Sensor, Cupboard Sensor, Coffee Sensor, Tea Sensor, Sugar Sensor, Water sensor and Kettle sensor. Dataset was collected and tested over a period of 4 weeks on 58 instances of people preparing either a cold drink or a hot dink (either tea or coffee). The result following analysis of the initial data have validated the conceptual approach and have shown the ability of the evidential networks to correctly classify 100% of all of the drink preparation experiments. The authors presented the evidential networks of making a cold drink and making a hot drink as it is shown in Figure 5.4-1.

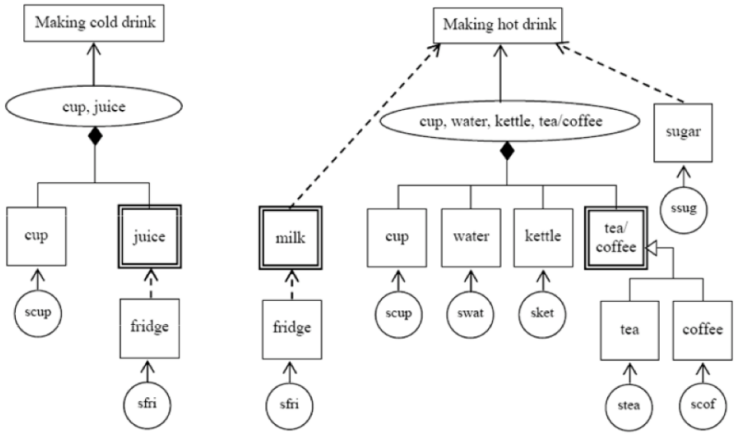

Figure 5.4-1: Examples of evidential networks for making a cold drink and making a hot drink

Susan McKeever in her PhD thesis extended Dempster-Shafer theory with temporal and quality knowledge [5.4.22]. Temporal data was added by defining absolute time of some activities when possible. For instance, breakfast is in the morning and dinner is at night. In addition, the duration of each activity is set to increase the probability in detecting the right situation. The quality knowledge is added if the sensor has beforehand uncertainty in giving accurate results. For example, a manufacturer's accuracy of 95% for a temperature sensor will result in all belief assignments reduced by 95% of their original value (with the remaining 0.05 assigned to uncertainty). McKeever tested the extended version of Dempster-Shafer on two datasets. The first dataset was the same one that was used in [5.4.23], which is collected in a department over 28 day period with 14 digital sensors. The second dataset was collected in a lab environment over 5 days using 3 main sensors. The sensors are: a computer activity sensor to monitor the user on her desk, a calendar sensor to collect information about the user's scheduled and a location sensor to locate the user. Six situations were annotated by the user during collecting the latter dataset: busy at computer, busy reading at desk, coffee break, lunch break, informal break and at meeting. The results showed that adding temporal and quality enhancements to the evidence decision network increased the recognition accuracy. Even though Dempster-Shafer is a potential solution when training data is unavailable and the relationship between sensors and situations is detectable, McKeever indicated that using the belief theory is not suitable when there is complex relationship from sensors to situations, which is only discernible via training data.

Moutacalli et al. relied on sequential pattern mining to analyze data collected by sensors in a smart home [5.4.23]. The aim was to discover the frequent activities of the home that the user usually does. The authors defined a list of all the activities that the user usually performs, as well as their component actions, in order to choose which activity the user is actually performing. For instance, the activity preparing coffee can be composed of the actions: take cup, pour coffee, add milk and add sugar. For each activity, the time that this activity usually happens and the names of sensors that are triggered during the activity are also defined. When the values of some sensors change significantly, the list of the activities is checked based on the time and the triggered sensors, to detect what activity is mostly happening at the moment. The current time helps in ignoring some activities immediately. For example, the activity

taking a shower might be segmented into two intervals: from 8:30 A.M. to 10:00 A.M. and from 7:00 P.M. to 10:00 P.M. If the current time is 1:00 pm, the activity taking a shower will not be considered. This segmentation diminished the activity search time by more than 70%. The validation has been done using a dataset collected in 28 days using 14 sensors. There were seven distinct activities to detect: leave house, use toilet, take shower, go to bed, prepare breakfast, prepare dinner and get drink. The average accuracy of detecting activities was around 85%.

5.5 Event Segmentation Models

In this section, we describe the mathematical models we are investigating for their use in the Event Segmentation and Identification task. This task is a critical element of Lifelogging as it provides the base unit (the "Event") of the lifelog index. A precise and accurate model is required.

We are investigating a number of methods for this task of creating an event segmentation model. Initial work has concentrated on belief network models (and specifically the Dempster-Shafer model (Section 5.5.1)). Research is also underway into the creation of an event segmentation model using machine learning methods, such as clustering algorithms (kNN) and classification algorithms (Support Vector Machines) (Section 5.5.2).

5.5.1 Belief network models

In this section, we describe the work we have done using belief-theory networks to develop an episode classification model based on (possibly) incomplete sensor data.We describe the background theory and derivations of Dempster-Shafer belief theory first. This is then developed and modified to fit the specific requirements, conditions and data that will be available in the Dem@Care system. The belief network model is proposed as a potential model to be used in the event identification task of lifelogging.

Dempster-Shafer theory is a mathematical theory of evidence. It is an attempt to allow more interpretation of what uncertainty is all about [5.5.1]. Dempster-Shafer is widely used in domains where information (evidence) is known to be imperfect and reasoning uncertain [5.5.2]. It allows combining evidences from different sources to arrive at a degree of belief [5.5.3]. While probability theory takes it as something either is or isn't true, Dempster-Shafer theory allows for more nebulous states of a system, such as "unknown".

The Dempster-Shafer Theory (DST) was firstly raised by Dempster [5.5.4] in 1967 and developed by Shafer [5.5.5] in 1976. DST is a synonym of evidence theory and extended to cater for different scenarios. DST can handle uncertainty caused by inaccurate knowledge. It is an expansion of Bayesian Theory. Evidence theory (ET) is a mathematically well-defined theory for handling conflict between different bodies of evidence. It is conceptually the same as Bayesian theory except that it uses epistemic (subjective) uncertainty [5.5.6].

Besides its key features, McKeever [5.4.22] in her PhD thesis identifies the following advantages of DST when applied to situation recognition:

1) Sensors are unreliable; an ability to quantify this lack of reliability and preserve the resulting uncertainty will support the quantification of situation uncertainty;

2) Rules are uncertain, and this uncertainty can be used to contribute to situation uncertainty calculations;

3) The theoretically sound basis for incorporating domain knowledge offers us a way to encode knowledge without relying on training data.

Theory and previous work

BPAF (Basic Probability Assignment Function)

Dempster-Shafer Theory (DST) is based on a universal set U. U is called the Frame of Discernment. The frame of discernment should exhibit 1) mutual exclusion 2) limited elements. The power set
2^U is the set of all subset of U, including empty set ∅. For example: U={a, b}, 2^U∅={a, b, U, ∅}. Hypotheses are defined as any subset of the frame of discernment.

The function $m: 2^U \rightarrow [0,1] m: 2^U \rightarrow [0,1]$ is called a basic belief assignment or mass function, when it satisfies the following conditions. For hypotheses A:

$$m(\emptyset) = 0$$

$$\sum_{A \subseteq U} m(A) = 1$$

These two equations tell us: 1) Belief from evidence cannot be assigned to empty hypotheses. 2) Belief from the evidence assigned to the every possible hypothesis must sum to 1. Uncertainty is expressed by the symbol ∅, which is a hypotheses including all elements in U. $m(A)$ shows a level the evidences supporting hypotheses A.

Dempster combination rules:

A vital part of DST to evaluate evidence is to fuse evidence from different sources. The basic combination rule fusing evidences from two independent sources is achieved by Dempster's combination rules:

$$m(A) = \frac{1}{1-k} \times \sum_{X \cap Y = A} m_1(X) m_2(Y)$$

$$k = \sum_{X \cap Y = \emptyset} m_1(X) m_2(Y)$$

This formula returns a belief value for hypothesis A. This value is proportional to the sum of the products of the mass functions containing that hypothesis. The mass functions m1 and m2 have belief sets X and Y, respectively. The value k is a normalizing factor and is based on the

degree of conflict between two mass functions, i.e., it varies according to the number of belief elements not shared between the sets X and Y.

Murphy's combination rule:

Murphy [5.5.7] proposed an modified version of Dempster's combination rules that will eliminate the problem caused by dominance of a single sensor and enable contradictory evidence to be preserved to a certain extent. When there are N evidence bodies, Murphy's rule first calculates the average of each hypothesis for the evidence. After calculating the averages, it applies the D-S combination rule with the averages N − 1 times

$$\tilde{m}(A) = \frac{1}{1-\bar{k}} \times \sum_{X \cap Y = A} \bar{m}_1(X)\bar{m}_2(Y)$$

$$\bar{k} = \sum_{X \cap Y = \emptyset} \bar{m}_1(X)\bar{m}_2(Y)$$

Here, $\bar{m}_1(X)$ and $\bar{m}_2(Y)$ are the averages of evidence for X and Y, respectively.

Averaging rule:

In Shafer's work [5.5.5], he combined belief functions by averaging all the evidence for each hypothesis (instead of the combination rule), as follows

$$m(A) = \frac{1}{n}\left(m_1(A) + \cdots + m_n(A)\right)$$

Averaging can be used to eliminate the influence of any strongly conflicting single belief [5.5.5]. The use of averaging improves an accurate record of contributing beliefs because no belief is 'lost', but it lacks convergence

Averaging rule provides a less conclusive picture because conflict is not normalised out. Meanwhile, both Dempster's and Murphy's rule are trying to emphasize evidence from sources that are in agreement and discard disagreeing evidence. One of the advantages of averaging rule is less computation. Averaging rule is often used to eliminate problems caused by conflicting sensors in binary sensors.

Sensor discounting:

Shafer defined an evidential operation for discounting sensor evidence [5.5.5]. When an evidence source is known to be less than 100% reliable, a discounting factor between 0 and 1 is applied to the source's beliefs. The impact of the discounting factor on beliefs is represented formally by Lowrance [5.5.8] as follows:

For a discount factor, d, where ($0 \leq d \leq 1$), where Θ represents uncertainty:

$$m_d(A) = \begin{cases} (1-d)m(A) & if\ A \neq \Theta \\ d + (1-d)m_d(\Theta) & if\ A = \Theta \end{cases}$$

Situation DAGs:

The situation Directed Acyclic Graphs captures knowledge about the environment that is relevant to the evidential reasoning process: sensors, sensor quality, abstracted context, inference rules, temporal information and situation hierarchies.

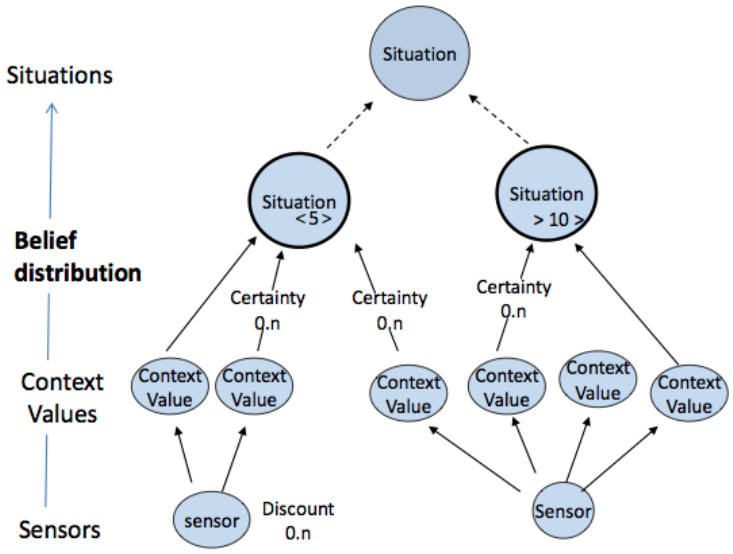

Figure 5.5-1: Example of Situation DAG

Evidence decision network:

We separate the situation recognition process into two steps 1) belief distribution and 2) decision making.

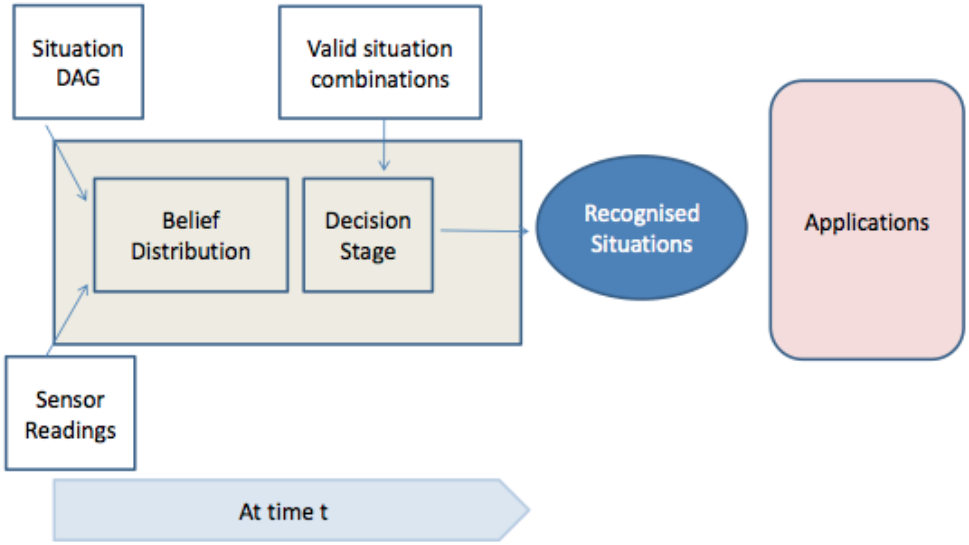

Figure 5.5-2: Structure of Evidence Decision Network

Belief distribution: Belief values are assigned to all nodes throughout the situation DAG. Mass functions are used to distribute mass to context values from raw sensor data. Evidence is fused and propagated to higher levels on the DAG until every node on the DAG has an assigned belief level.

Decision making: Once evidence has been propagated to all nodes in the DAG, a decision process is needed to decide the recognised situations. Once all evidence has been distributed, the decision algorithm is driven by the recognition requirements for the environment:

1. Can situations co-occur or can only one situation occur at a time?

2. If situations can co-occur, what are the invalid situation occurrence combinations?

The situation with the highest belief is occurring when only single situation can be happening at the same time. If several situations are allowed to co-occur, a belief threshold level will be applied. If any invalid co-occurrences of situation exist, the threshold will be reconsidered.

Dempster-Shafer models

Situation analysis and modelling

We use the categories identified in Dem@Care to explain how DS theory works on fusing multi-source data by

- Mass functions.
- Frames of discernment and evidence combination
- Evidence combination.
- Sleeping

The first Dem@Care prototype specification gives five situations which involve sleep, namely PwD just went to sleep, PwD is sleeping, PwD is taking a nap, Pwd just woke up, PwD is awake. There are five hardware sensor devices which measure parameters pertaining to sleep. These are the Kinect, Axis P13, DTI-2, Gear4 and WIMUs. The sensors contribute data to 8 parameters, as shown in the graph diagram below.

The frame of discernment for each of the sensor includes the singleton hypotheses {went to sleep, is sleeping, taking a nap, woke up, awake, other}, theoretically, it includes all possible combination in the frame, for example {went to sleep & other, is sleeping & went to sleep}, or even {went to sleep & is sleeping & taking a nap & woke up & awake & other} which is represented as theta.

Sensor reliability measures how much we can believe the sensor. Assume, we can define the accuracy of overall successful detection rate.

For each possible situation, we give a mass function for each sensor. So that each lowest node satisfy:

$m: 2^U \rightarrow [0,1]$

$$m(\emptyset) = 0$$

$$\sum_{A \subseteq U} m(A) = 1$$

The directed acyclic graph DAG for the sleeping scenario is shown in Figure 5.5-3:

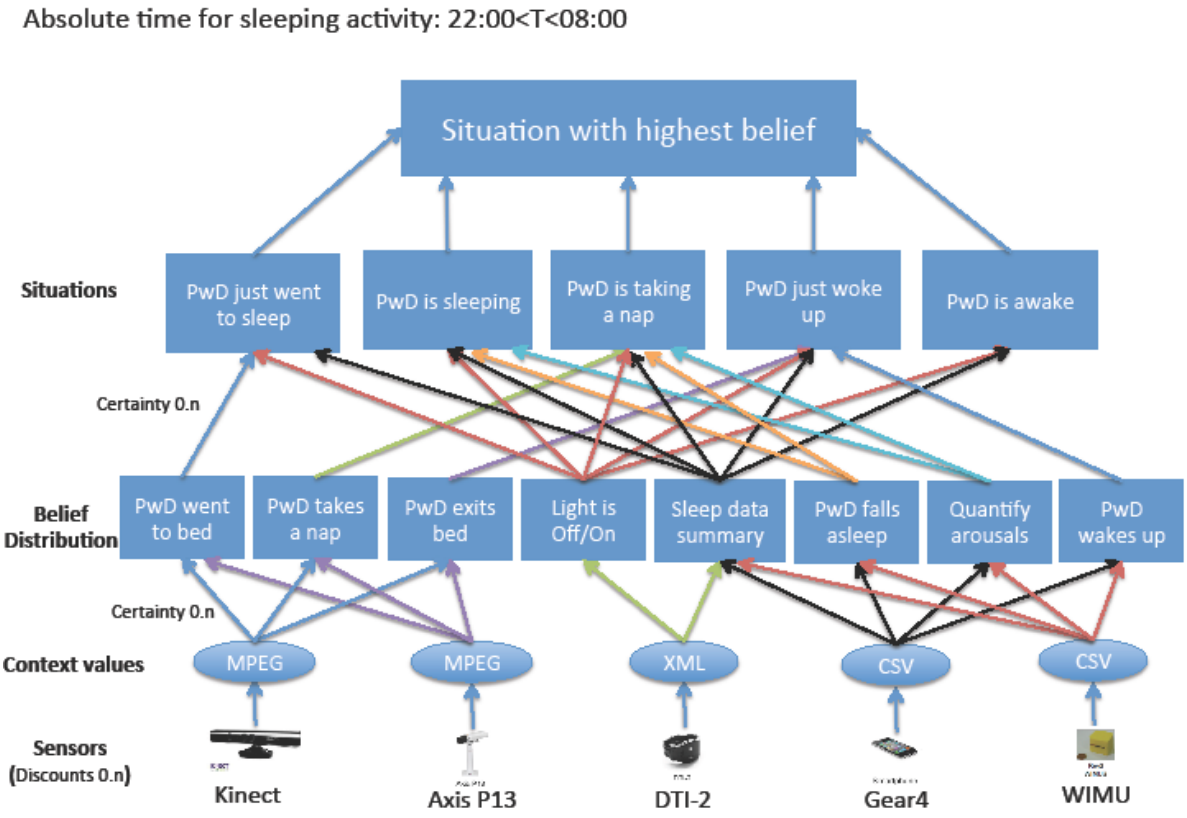

Figure 5.5-3: The directed acyclic graph DAG for the sleeping scenario

Temporal data is added for being sleeping between 22:00 and 08:00. The DAG is drawn based on:

Sensors that are involved in detecting sleeping/awake situations

Beliefs that each sensor can produce

Situations that can be detected based on combining different Beliefs

We can define the mass function based on the DAG. For example, we have three mass functions for Kinect for different movement detected:

m_{k1}(went to bed, didn't go to bed, uncertainty)

m_{k2}(takes nap, doesn't take nap, uncertainty)

m_{k3}(exits bed, doesn't exit bed, uncertainty)

We represent the different beliefs by different letters in the mathematical model:

Went to bed → "a",

Takes a nap → "b",

Exits bed → "c",

Light On → "d",

Sleeping data → "e",

Falls asleep → "f",

Arousals → "g",

Wakes up → "h"

Based on the DAG and the definition above, the mass functions for all sensors will be:

Kinect: $m_{k1}(a_1, a_2, \theta_{k1}), m_{k2}(b_1, b_2, \theta_{k2}), m_{k3}(c_1, c_2, \theta_{k3})$

P13: $m_{p1}(a_1, a_2, \theta_{p1}), m_{p2}(b_1, b_2, \theta_{p2}), m_{p3}(c_1, c_2, \theta_{p3})$

DTI: $m_{d1}(d_1, d_2, \theta_{d1}), m_{d2}(e_1, e, \theta_{d2})$

Gear4: $m_{g1}(e_1, e_2, \theta_{g1}), m_{g2}(f_1, f_2, \theta_{g2}), m_{g3}(g_1, g_2, \theta_{g3}), m_{g4}(h_1, h_2, \theta_{g4})$

WIMU: $m_{w1}(e_1, e_2, \theta_{w1}), m_{w2}(f_1, f_2, \theta_{w2}), m_{w3}(g_1, g_2, \theta_{w3}), m_{w4}(h_1, h_2, \theta_{w4})$

Table 5.5-1 shows the mass functions of all sensors together with the different Beliefs.

Table 5.5-1 The mass functions of all sensors together with the different Beliefs for sleeping scenario.

Beliefs	Kinect	Axis P13	DTI-2	Gear4	WIMU
PwD went to Bed	Mk1(a1,a2, θk1)	Mp1(a1,a2, θp1)			
PwD takes a nap	Mk2(b1,b2, θk2)	Mp2(b1,b2, θp2)			
PwD exits bed	Mk3(c1,c2, θk3)	Mp3(c1,c2, θp3)			
Light is On			Md1(d1, d2,θd1)		
Sleep data summary			Md2(e1,e2, θd2)	Mg1(e1,e2, θg1)	Mw1(e1,e2, θw1)

PwD falls asleep				Mg2(f1,f2, θg2)	Mw2(f1,f2, θw2)
Quantify arousals				Mg3(g1,g2, θg3)	Mw3(g1,g2, θw3)
PwD wakes up				Mg4(e1,h2, θg4)	Mw4(e1,h2, θw4)

According to the compatible relationship, we want to merge sensor data from two or three different sources:

For $m = m_1 \oplus m_2$, $m = m_1 \oplus m_2$, we can use basic Dempster-shafer method to fuse the data:

$$m(A) = \frac{1}{1-k} \times \sum_{X \cap Y = A} m_1(X) m_2(Y)$$

$$k = \sum_{X \cap Y = \emptyset} m_1(X) m_2(Y)$$

Or applying Shafer's Averaging combination::

$$m(A) = \frac{1}{n}\left(m_1(A) + \cdots + m_n(A)\right)$$

For two sources of combination using D-S combination:

$$m_1(a_1, a_2, \theta_1) \oplus m_2(a_1', a_2', \theta_1') = \left(\frac{a_1 a_1' + a_1 \theta_1'}{1-k}, \frac{a_2 a_2' + a_2' \theta_1}{1-k}, \frac{\theta_1 \theta_1'}{1-k}\right)$$

$$k = a_1 a_2' + a_2 a_1'$$

We can then have 8 frames of discernments namely:

$$2^{U_1} = \{\text{went to bed, didn't go to bed, uncertainty}\}$$

$$2^{U_2} = \{\text{takes nap, doesn't take nap, uncertainty}\}$$

$$2^{U_3} = \{\text{exits bed, doesn't exit bed, uncertainty}\}$$

$$2^{U_4} = \{\text{Light on, light off, uncertainty}\}$$

$$2^{U_5} = \{\text{sleep data summary, no sleep data summary, uncertainty}\}$$

$$2^{U_6} = \{\text{falls asleep, doesn't fall asleep, uncertainty}\}$$

$$2^{U_7} = \{\text{quantify arousal, not quantify arousal, uncertainty}\}$$

$$2^{U_8} = \{\text{wakes up, doesn't wake up, uncertainty}\}$$

Applying the same combination method again, 8 beliefs will be summarized to 5 situations.

$$2^{U_1'} = \{\text{went to bed, didn't go to bed, uncertainty}\}$$

$$2^{U_2'} = \{\text{sleeping, not sleeping, uncertainty}\}$$

$$2^{U_3'} = \{\text{takes a nap, doesn't take nap, uncertainty}\}$$

$$2^{U_4'} = \{\text{woke up, didn't wake up, uncertainty}\}$$

$$2^{U_5'} = \{\text{awake, not awake, uncertainty}\}$$

2. Eating ADL/IADL

Two sensors (Axis P13 and GoPro) are used to detect six situations, namely PwD is having a meal alone, PwD is having a meal with someone, PwD is having coffee/tea alone, PwD is having coffee/tea with someone, PwD finished eating/coffee, PwD is preparing a meal. The six situations is determined by the merging of 7 parameters produced by the two sensors as shown in the DAG diagram below.

Figure 5.5-4 shows the DAG for Eating scenario:

Absolute time for Breakfast activity: 07:00<T<09:00
Absolute time for Lunch activity: 12:00<T<14:00
Absolute time for Dinner activity: 17:00<T<20:00

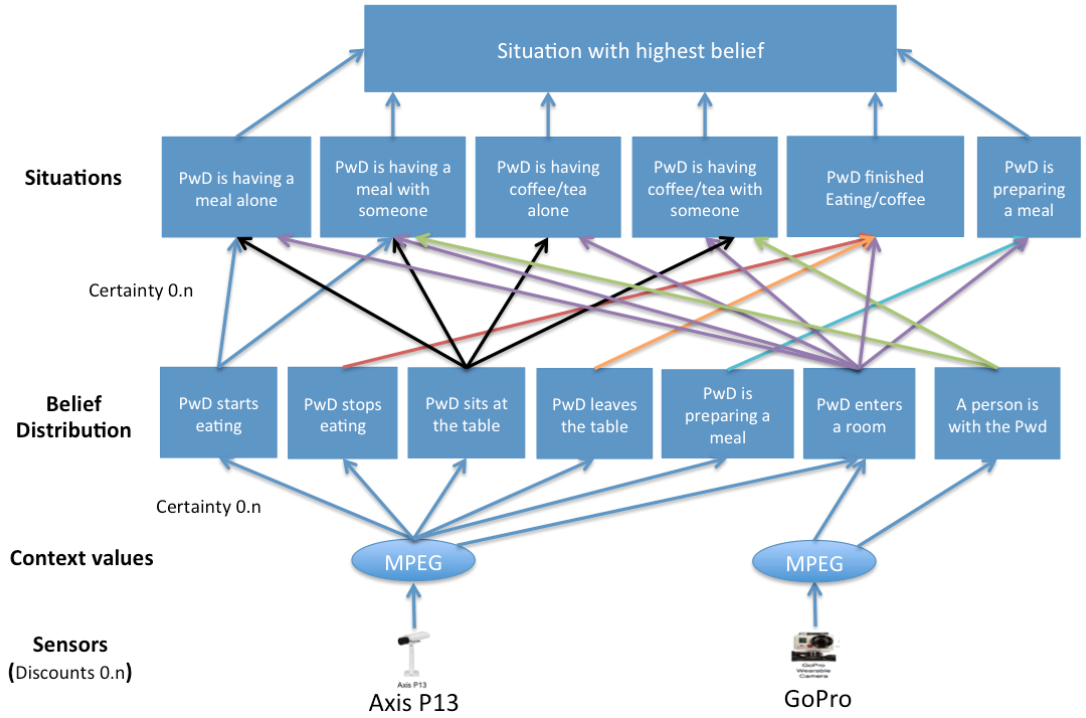

Figure 5.5-4: The DAG for Eating scenario

Temporal data is added for 3 different activities:

Having breakfast between 07:00 and 09:00

Having lunch between 12:00 and 14:00

Having dinner between 17:00 and 20:00

We can define the mass function based on the DAG. For example, we have two mass functions for GoPro for different movement detected:

$$m_{g1}(\text{PwD enters a room, PwD doesn't enter a room, uncertainty})$$

$$m_{g2}(\text{A person is with the PwD, The PwD is alone, uncertainty})$$

We represent the different Beliefs by different letters in the mathematical model:

PwD starts eating → "i",

PwD stops eating → "j",

PwD sits at the table → "k",

PwD leaves the table → "l",

PwD is preparing a meal → "m",

PwD enters a room → "n",

A person is with the PwD → "o"

Based on the DAG and the definition above, the mass functions for the sensors will be:

P13: Mp1 (i1, i2, θp1), Mp2 (j1, j2, θp2), Mp3 (k1, k2, θp3), Mp4 (l1, l2, θp1), Mp5 (m1, m2, θp2), Mp6 (n1, n2, θp3)

GoPro: Mg1 (n1, n2, θg1), Mg2 (o1, o2, θg2)

Table 5.5-2 shows the mass functions of all sensors together with the different Beliefs.

Table 5.5-2: The mass functions of all sensors together with the different Beliefs in the eating scenario.

Beliefs	Axis P13	GoPro
PwD starts eating	Mp1 (i1,i2, θp1)	
PwD stops eating	Mp2 (j1,j2, θp2)	
PwD sits at the table	Mp3 (k1,k2, θp3)	
PwD leaves the table	Mp4 (l1,l2, θp1)	

PwD is preparing a meal	Mp5 (m1,m2, θp2)	
PwD enters a room	Mp6 (n1,n2, θp3)	Mg1 (n1,n2, θg1)
A person is with the PwD		Mg2 (o1,o2, θg2)

We can then have 7 frames of discernments namely:

$$2^{U_1} = \{\text{PwD starts eating, PwD didn't start eating, uncertainty}\}$$

$$2^{U_2} = \{\text{PwD stops eating, PwD didn't stop eating, uncertainty}\}$$

$$2^{U_3} = \{\text{PwD sits at the table, PwD didn't sit at the table, uncertainty}\}$$

$$2^{U_4} = \{\text{PwD leaves the table, PwD didn't leave the table, uncertainty}\}$$

$$2^{U_5} = \{\text{PwD is preparing a meal, PwD is not preparing a meal, uncertainty}\}$$

$$2^{U_6} = \{\text{PwD enters a room PwD doesn't enter a room, uncertainty}\}$$

$$2^{U_7} = \{\text{A person is with the PwD, the PwD is alone, uncertainty}\}$$

Applying the same combination method again, 7 beliefs will be summarized to 6 situations.

$$2^{U_1'} = \{\text{PwD is having a meal alone, PwD is not having a meal alone, uncertainty}\}$$

$$2^{U_2'} = \{\text{having a meal with someone, not having a meal with someone, uncertainty}\}$$

$$2^{U_3'} = \{\text{PwD is having coffee/tea alone, PwD is not having coffee/tea alone, uncertainty}\}$$

$$2^{U_4'} = \{\text{having coffee/tea with someone, not having coffee /tea with someone, uncertainty}\}$$

$$2^{U_5'} = \{\text{PwD finished Eating/coffee, PwD didn't finish Eating/coffee, uncertainty}\}$$

$$2^{U_6'} = \{\text{PwD is preparing a meal, PwD is not preparing a meal, uncertainty}\}$$

3. Exercise

Four sensors (Axis, WIMU, DTI-2, Accelerometers) are used to detect seven situations, namely PwD is walking inside home, PwD is walking outdoors, PwD is not doing an exercise, PwD fell down , PwD is doing an exercise/activity indoor (e.g. cleaning the house), PwD is doing an exercise/activity outdoor (e.g. running), PwD is travelling. The seven situations are determined by the merging of 5 parameters produced by the four sensors as shown in the DAG diagram in Figure 5.5-5.

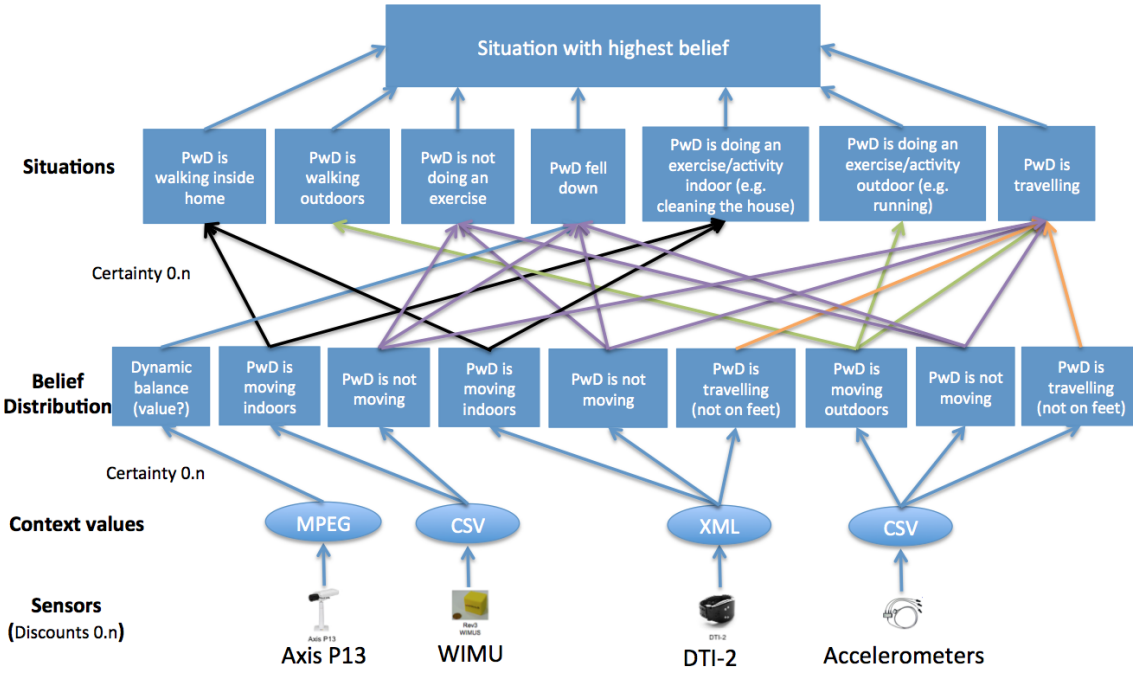

Figure 5.5-5 The DAG for exercise scenario

We can define the mass function based on the DAG. For example, we have two mass functions for WIMU for different movement detected:

$$m_{w1}(\text{PwD is moving indoors, PwD is not moving indoors, uncertainty})$$

$$m_{w2}(\text{PwD is not moving, PwD is moving, uncertainty})$$

We represent the different Beliefs by different letters in the mathematical model:

Dynamic balance → "p"

PwD is moving indoors → "q",

PwD is not moving → "r"

PwD is moving outdoors → "s"

PwD is travelling (not on feet) → "t"

Based on the DAG and the definition above, the mass functions for the sensors will be:

P13: $m_{p1}(p_1, p_2, \theta_{p1})$

WIMU: $m_{w1}(q_1, q_2, \theta_{w1})$, $m_{w2}(r_1, r_2, \theta_{w2})$

DTI: $m_{d1}(q_1, q_2, \theta_{d1})$, $m_{d2}(r_1, r_2, \theta_{d2})$, $m_{d3}(t_1, t_2, \theta_{d3})$

ACC: $m_{a1}(s_1, s_2, \theta_{a1})$, $m_{a2}(r_1, r_2, \theta_{a2})$, $m_{a3}(t_1, t_2, \theta_{a3})$

Table 5.5-3 shows the mass functions of all sensors together with the different Beliefs.

Table 5.5-3 The mass functions of all sensors together with the different Beliefs for exercise scenario.

Beliefs	Axis P13	WIMU	DTI-2	ACC
Dynamic balance	Mp1 (p1, p2, θp1)			
PwD is moving indoors		Mw1(q1,q2, θw1)	Md1 (q1, q2, θd1)	
PwD is not moving		Mw2(r1,r2, θw2)	Md2(r1,r2, θd2)	Ma2(r1,r2, θa2)
PwD is moving outdoors				Ma1(s1,s2, θa1)
PwD is travelling (not on feet)			Md3(t1,t2, θd3)	Ma3(t1,t2, θa3)

We can then have 5 frames of discernments namely:

$$2^{U_1} = \{\text{Dynamic balance, Not Dynamic balance, uncertainty}\}$$

$$2^{U_2} = \{\text{PwD is moving indoors, PwD is not moving indoors, uncertainty}\}$$

$$2^{U_3} = \{\text{PwD is not moving, PwD is moving, uncertainty}\}$$

$$2^{U_4} = \{\text{PwD is moving outdoors, PwD is not moving outdoors, uncertainty}\}$$

$$2^{U_5} = \{\text{PwD is travelling (not on feet), PwD is not travelling (not on feet), uncertainty}\}$$

Applying the same combination method again, 5 beliefs will be summarized to 7 situations.

$$2^{U_1'} = \{\text{PwD is walking inside home, PwD is not walking inside home, uncertainty}\}$$

$$2^{U_2'} = \{\text{PwD is walking outdoors, PwD is not walking outdoors, uncertainty}\}$$

$$2^{U_3'} = \{\text{PwD is not doing an exercise, PwD is doing an exercise, uncertainty}\}$$

$$2^{U_4'} = \{\text{PwD fell down, PwD didn't fall down, uncertainty}\}$$

$$2^{U_5'} = \{\text{PwD is doing an activity indoor, PwD is not doing an activity indoor, uncertainty}\}$$

$$2^{U_6'} = \{\text{doing an activity outdoor, not doing an activity outdoor, uncertainty}\}$$

$$2^{U_7'} = \{\text{PwD is travelling, PwD is not travelling, uncertainty}\}$$

4. Social Activities

Four sensors (GoPro, P13, Sensecam, Microphone) are used to detect four situations, namely PwD called someone, Someone called the Pwd, Pwd had a visitor home, PwD is with someone outside home. The four situations is determined by the merging of 6 parameters produced by the four sensors.

Figure 5.5-6 shows the DAG for Social Activities scenario:

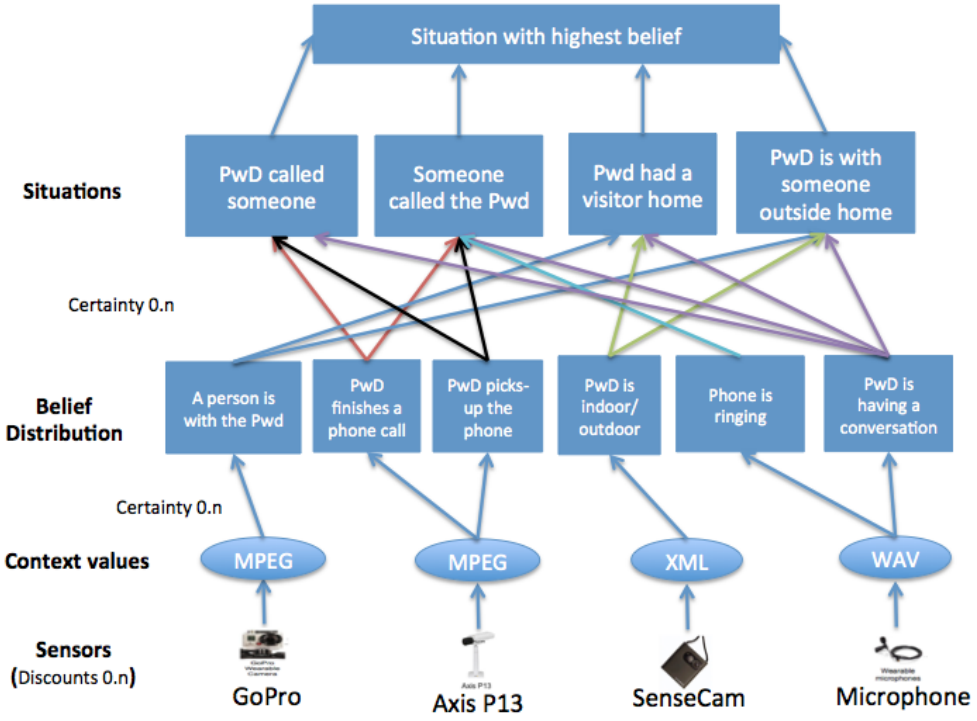

Figure 5.5-6: The DAG for social activities scenario

We can define the mass function based on the DAG. For example, we have two mass functions for P13 for different movement detected:

$$m_{p1}(\text{PwD finishes a phone call, PwD does not finish a phone call, uncertainty})$$

$$m_{p2}(\text{PwD picks up the phone, PwD does not pick up the phone, uncertainty})$$

We represent the different Beliefs by different letters in the mathematical model:

A person is with the Pwd → "u"

PwD finishes a phone call → "v"

PwD picks-up the phone → "w"

PwD is indoor → "x"

Phone is ringing → "y"

PwD is having a conversation → "z"

Based on the DAG and the definition above, the mass functions for the sensors will be:

GoPro: $m_{g1}(u_1, u_2, \theta_{g1})$

P13: $m_{p1}(v_1, v_2, \theta_{p1})$, $m_{p2}(w_1, w_2, \theta_{p2})$

SenseCam: $m_{s1}(x_1, x_2, \theta_{s1})$

Microphone: $m_{m1}(y_1, y_2, \theta_{m1})$, $m_{m2}(z_1, z_2, \theta_{m2})$

Table 5.5-4 shows the mass functions of all sensors together with the different beliefs.

Table 5.5-4 The mass functions of all sensors together with the different beliefs for social activities scenario.

Beliefs	GoPro	Axis P13	SenseCam	Microphone
A person is with the Pwd	Mg1 (u1, u2, θg1)			
PwD finishes a phone call		Mp1 (v1, v2, θp1)		
PwD picks-up the phone		Mp2 (w1, w2, θp2)		
PwD is indoor			Ms1(x1,x2, θs1)	
Phone is ringing				Mm1 (y1, y2, θm1)
PwD is having a conversation				Mm2(z1,z2, θm2)

We can then have 6 frames of discernments namely:

$$2^{U_1} = \{\text{A person is with the Pwd, No one is with the Pwd, uncertainty}\}$$

$$2^{U_2} = \{\text{PwD finishes a phone call, PwD does not finish a phone call, uncertainty}\}$$

$$2^{U_3} = \{\text{PwD picks up the phone, PwD does not pick up the phone, uncertainty}\}$$

$$2^{U_4} = \{\text{PwD is indoor, PwD is not indoor, uncertainty}\}$$

$$2^{U_5} = \{\text{Phone is ringing, Phone is not ringing, uncertainty}\}$$

$$2^{U_6} = \{\text{PwD is having a conversation, PwD is not having a conversation, uncertainty}\}$$

Applying the same combination method again, 6 beliefs will be summarized to 4 situations.

$$2^{U_1'} = \{\text{PwD called someone, PwD didn't make a call, uncertainty}\}$$

$$2^{U_2'} = \{\text{Someone called the Pwd, No one called the Pwd, uncertainty}\}$$

$$2^{U_3'} = \{\text{Pwd had a visitor home, Pwd did not have a visitor home, uncertainty}\}$$

$$2^{U_4'} = \{\text{PwD is with someone outside home, PwD is alone outside home, uncertainty}\}$$

5.Presenting all beliefs

Table 5.5-5 presents all beliefs together with all sensors based on the tables presented above

Table 5.5-5 All beliefs together with all sensors for all scenarios

Beliefs	**Kinect**	**Axis P13**	**DTI-2**	**Gear4**	**WIMU**	**GoPro**	**ACC**	**SenseCam**	**Microphone**
PwD went to Bed	Mk1(a1, a2, θk1)	Mp1(a1, a2, θp1)							
PwD takes a nap	Mk2(b1, b2, θk2)	Mp2(b1, b2, θp2)							
PwD exits bed	Mk3(c1, c2, θk3)	Mp3(c1, c2, θp3)							
Light is On			Md1(d1, d2,θd1)						
Sleep data summary			Md2(e1, e2, θd2)	Mg1(e1, e2, θg1)	Mw1(e1, e2, θw1)				
PwD falls asleep				Mg2(f1,f2, θg2)	Mw2(f1, f2, θw2)				
Quantify arousals				Mg3(g1, g2, θg3)	Mw3(g1, g2, θw3)				
PwD wakes up				Mg4(e1, h2, θg4)	Mw4(e1, h2, θw4)				
PwD starts eating		Mp1 (i1,i2, θp1)							
PwD stops eating		Mp2 (j1,j2, θp2)							
PwD sits at the		Mp3 (k1,k2,							

table		θp3)							
PwD leaves the table		Mp4 (l1,l2, θp1)							
PwD is preparing a meal		Mp5 (m1,m2, θp2)							
PwD enters a room		Mp6 (n1,n2, θp3)				Mg1 (n1,n 2, θg1)			
A person is with the PwD						Mg2 (o1,o 2, θg2)			
Dynamic balance		Mp1 (p1, p2, θp1)							
PwD is moving indoors			Md1 (q1, q2, θd1)		Mw1(q1, q2, θw1)				
PwD is not moving			Md2(r1, r2, θd2)		Mw2(r1, r2, θw2)		Ma2(r1, r2, θa2)		
PwD is moving outdoors							Ma1(s1, s2, θa1)		
PwD is travelling (not on feet)			Md3(t1,t 2, θd3)				Ma3(t1, t2, θa3)		
A person is with the Pwd						Mg1 (u1, u2, θg1)			
PwD finishes a phone call		Mp1 (v1, v2, θp1)							
PwD picks-up the phone		Mp2 (w1, w2, θp2)							
PwD is								Ms1(x1,	

indoor								x2, θs1)	
Phone is ringing									Mm1 (y1, y2, θm1)
PwD is having a conversation									Mm2(z1, z2, θm2)

5.5.2 Event Classification using Machine Learning

In this section we describe how machine-learning methods will be used to create models for event segmentation and event identification. We identify the motivations from a theoretical point of view for investigating and experimenting with particular machine-learning methods.

As described earlier in the chapter, many of the automated event classification systems in lifelogging use machine-learning methods in order to cluster and classify events. For example, k-Nearest Neighbour can be used to classify unlabelled activities by identifying their similarity to some manually labelled cases. Similarly, clustering can be used to collect similar cases into clusters that can then be labelled en-masse by a manual operation.

Models based on decision trees are interesting for us, since they can be used to easily model the sensor data received and allows for the reflection on how particular sensors interact and may show any correlations. We may also be able to identify if data from some sensors are redundant due to their being superseded or subsumed by other sensors. They will also provide an interesting parallel to the belief networks generated using Dempster-Shafer theory.

Having established a suite of models based on different ML techniques, we will evaluate their performance using a number of different datasets. Primarily, we will use the collections of Dem@Care sensor data gathered in Nice and Thessaloniki. Later on, we will use the data from the pilot deployments as it becomes available.

These data collections will be good representations of real-world data and we can expect that, in some of the data instances, there will be some missing sensor data, incomplete or truncated data, and miscalibrated or erroneous data. Such data sets then will allow is to test not only the relative and absolute performances of the models, but also will test the robustness of the systems in how they perform with incomplete or incorrect data.

Our experiments will also examine how well particular algorithms perform with the different types of sensor data: are there correlations to be found between an algorithm's performance and the data on which it is applied. Findings here could have implications for sensor choice and algorithm choice. Furthermore, there may be trade-offs between issues of speed and accuracy where these findings could also influence matters.

5.6 Conclusion

In this section, we explained how life-logging technology will be used within the Dem@Care project to reason about the day of the person. Different sensors might give different indications of the person's current activity, so the aim of the life-logging task is to reason and return the activity with the highest belief. All sensors data will be then aggregated and segmented into activities based on the reasoning results. In conclusion, the day of the person will be organized as lifelogs that are searchable and browsable.

The outcome of the life-logging task will help the participants to review their lifelogs on a daily and weekly basis. As well as the benefits to the PwD, the lifelog could also be used by the clinician and the carers to verify that instructions and directions are being followed appropriately.

6 Conclusions

In this deliverable we reported the results of WP 4 on processing visual data, specifically, the contributions of all the partners (UB1, LTU, DCU, CERTH, and INRIA) on posture recognition, action recognition, activity monitoring, and lifelogging.

For posture estimation, we introduced a new filtering methodology aimed at producing more accurate ego-motion estimation, to be applied on pose estimation using video stream of the wearable camera. Experiments with synthetic data show the capacity of this approach to improve in the accuracy of camera position and orientation estimation. Future work will consider this as an input to location and semantic posture inference.

For action recognition, several complementary approaches are presented.Firstly, for the wearable camera video stream, we proposed new methods for object recognition in egocentric vision, based on visual saliency. A model was proposed that assumes a temporal shift of visual saliency between the person executing different activities, i.e. the Actor (Patient) and the Viewer (doctor) who interprets this content a posteriori. Psychovisual experiments have confirmed this assumption. The concept of saliency was included within object recognition algorithms. It was shown experimentally that this approach provides improvement over the basic BoW model and achieves closer results to a hypothetical case in which optimal foreground masks are available in test.

Secondly, for video from static camera, we presented an algorithm for action recognition segmenting trajectories based on changes in their statistical nature. The algorithm derives meaningful sub-trajectories that are related to the changes in the person's motion. With this segmentation, the proposed algorithm becomes scale and viewpoint invariant. Besides that, this algorithm also uses Motion Boundary Activity Areas (MBAA) to reduce the computational cost while it still maintains high recognition accuracy.

Thirdly, and also for video from static camera, we presented another action recognition algorithm focussing more on accuracy improvement. This algorithm uses a dynamic coordinate system with the head as the origin. The experiments showed that with the dynamic coordinate system, the proposed algorithm outperformed existing state-of-the-art techniques for action recognition.

For activity monitoring, we presented a generic framework using RGB-D camera based on hierarchical descriptive models. The framework performance increases considerably using RGB-D cameras in terms of the recall index. Although the framework can be used to recognise both short term and long term activities, in WP4 we only focus on short term activity recognition. Long term activity recognition is going to be presented in details in WP5.

For lifelogging, we explain how life-logging technology will be used within the Dem@Care project to reason about the day of the person. We also presented the algorithms to aggregate information from different sensors about the person's current activity. The aim of this algorithm is to reason and return the activity with the highest belief using different indicators of different sensors. As a result, the day of the person will be organized as lifelogs that are searchable and browsable.ferences

[2.1.1] H Snoussi and A Mohammad-Djafari, "Particle filtering on riemannian manifolds," in AIP Conference , Vol. Issue 1, p219, 2006, vol. 872.

[2.1.2] Frank Tompkins and Patrick J. Wolfe, “Bayesian filtering on the stiefel manifold,” in IEEE International Workshop on Computational Ad- vances in Multi-Sensor Adaptive Processing, 2007.

[2.1.3] Q Rentmeesters, P. Absil, P. Van Dooren, K. Gallivan, and A. Srivas- tava, “An efficient particle filtering technique on the grassmann mani- fold,” in IEEE International Conference on Acoustics Speech and Sig- nal Processing, 2010.

[2.1.4] Simon S Haykin et al., Kalman filtering and neural networks, Wiley Online Library, 2001.

[2.1.5] D. Simon, “Kalman filtering with state constraints: a survey of linear and nonlinear algorithms,” IET Control Theory & Applications, vol. 4, no. 8, pp. 1303–1318, 2010.

[2.1.6] E. Kraft, “A quaternion-based unscented kalman filter for orientation tracking,” in International Conference on Information Fusion, 2003, vol. 1, pp. 47–54.

[2.1.7] J. Crassidis and F. Markley, “Unscented filtering for spacecraft attitude estimation,” Journal of Guidance, Control, and Dynamics, vol. 26, pp. 536–542, 2003.

[2.1.8] Paul Smith, Tom Drummond, and Kimon Roussopoulos, “Computing map trajectories by representing, propagating and combining pdfs over groups,” in ICCV, 2003, pp. 1275–1282.

[2.1.9] S. Persson and I. Sharf, “Invariant momentum-tracking kalman filter for attitude estimation,” in IEEE Conference on Robotics and Automation, 2012.

[2.1.10] G.S. Chirikjian, Stochastic Models, Information Theory, and Lie Groups, Springer, 2012.

[2.1.11] Y. Wang and G. Chirikjian, “Error propagation on the euclidean group with applications to manipulators kinematics,” IEEE Transactions on Robotics, vol. 22, 2006.

[2.1.12] K. Wolfe, M. Mashner, and G. Chirikjian, “Bayesian fusion on lie groups,” Journal of Algebraic Statistics, vol. 2, pp. 75–97, 2011.

[2.1.13] P. Maybeck, Stochastic Models, Estimation, and Control, Academic Press, 1979.

[2.1.14] Y. Ma, S. Soatto, J. Kosecka, and S. Sastry, An Invitation to 3D Vision: From Images to Geometric Models, Springer, 2004.

[2.1.15] J Selig, “Lie groups and lie algebras in robotics,” Computational Non-commutative Algebra and Applications, pp. 101–125, 2005.

[2.1.16] Chien-Ping Lu, Gregory D. Hager, and Eric Mjolsness, “Fast and globally convergent pose estimation from video images,” IEEE Trans. Pat- tern Anal. Mach. Intell., vol. 22, no. 6, pp. 610–622, 2000.

[2.1.17] F. Moreno-Noguer, V. Lepetit and P. Fua, Accurate Non-Iterative O(n) Solution to the PnP Problem, IEEE International Conference on Computer Vision, Rio de Janeiro, Brazil, October 2007.

[3.1.1] C.C. Gonzalez and M.R. Burke, “The brain uses efference copy information to optimise spatial memory,” Experimental Brain Research, vol. 224, pp. 189–197, 2013.

[3.1.2] C. Prablanc, J.F. Echailler, E. Komilis, and M. Jeannerod, “Optimal response of eye and hand motor systems in pointing at a visual target” Biol. Cybernetics, vol. 35, pp. 113–124, 1979.

[3.1.3] Michael Dorr, Thomas Martinetz, Karl R. Gegenfurtner, and Erhardt Barth, “Variability of eye movements when viewing dynamic natural scenes,” Journal of vision, vol. 10, no. 10, 2010.

[3.1.4] Alireza Fathi, Yin Li, and JamesM. Rehg, “Learning to recognize daily actions using gaze,” in Computer Vision – ECCV 2012, Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, Eds., vol. 7572 of Lecture Notes in Computer Science, pp. 314–327. Springer Berlin Heidelberg, 2012.

[3.1.5] David Wooding, “Eye movements of large populations: Ii. deriving regions of interest, coverage, and similarity using fixation maps,” Behavior Research Methods, vol. 34, pp. 518–528, 2002, 10.3758/BF03195481.

[3.1.6] Andrew T. Duchowski, Eye Tracking Methodology: Theory and Practice, Second Edition, Springer-Verlag London Limited, 2007.

[3.1.7] Donald C. Hood and Marcia A. Finkelstein, “Sensitivity to light,” in Handbook of perception and human performance, Volume 1: Sensory processes and perception, K. R. Boff, L. Kaufman, and J. P. Thomas, Eds., chapter 5, pp. 5–1–5–66. JohnWiley & Sons, New York, NY, 1986.

[3.1.8] O. Le Meur and T. Baccino, “Methods for comparing scanpaths and saliency maps: strengths and weaknesses.,” Behav Res Methods, 2012.

[3.1.9] Sophie Marat, Tien Ho Phuoc, Lionel Granjon, Nathalie Guyader, Denis Pellerin, and Anne Gu´erin-Dugu´e, “Modeling spatio-temporal saliency to predict gaze direction for short videos,” International Journal of Computer Vision, vol. 82, no. 3, pp. 231–243, 2009, D´epartement Images et Signal.

[3.1.10] M. F. Land and M. Hayhoe, “In what ways do eye movements contribute to everyday activities?,” Vision research, vol. 41, no. 25-26, pp. 3559–3565, 2001.

[3.2.1] 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.

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

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

[3.2.4] 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.

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

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

[3.2.7] E. S. Page, “Continuous inspection scheme,” Biometrika, vol. 41, pp. 100–115, 1954.

[3.3.1] M. Ahad, J. Tan, H. Kim, and S. Ishikawa. Motion history image: its variants and applications. MVA, 2010.
[3.3.2]Y. Benabbas, A. Lablack, N. Ihaddadene, and C. Djeraba. Action recognition using direction models of motion. In ICPR, 2010.
[3.3.3] P. Bilinski and F. Bremond. Evaluation of local descriptors for action recognition in videos. In ICVS, 2011.

[3.3.4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In CVPR, 2005.

[3.3.5] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In VS-PETS workshop, in conjunction with ICCV, 2005.

[3.3.6] A. A. Efros, A. C. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In ICCV, 2003.

[3.3.7] A. Gilbert, J. Illingworth, and R. Bowden. Action recognition using mined hierarchical compound features. TPAMI, 2011.

[3.3.8] Z. Jiang, Z. Lin, and L. S. Davis. Recognizing human actions by learning and matching shape-motion prototype trees. TPAMI, 2011.

[3.3.9] M. Kaaniche and F. Bremond. Recognizing gestures by learning local motion signatures of hog descriptors. TPAMI, 2012.

[3.3.10] Z. Kalal, J. Matas, and K. Mikolajczyk. P-n learning: Bootstrapping binary classifiers by structural constraints. In CVPR, 2010.

[3.3.11] T.-K. Kim, S.-F. Wong, and R. Cipolla. Tensor canonical correlation analysis for action classification. In CVPR, 2007.

[3.3.12] T.-S. Kim and Z. Uddin. Silhouette-based Human Activity Recognition Using Independent Component Analysis, Linear Discriminant Analysis and Hidden Markov Model. InTech, 2010.

[3.3.13] A. Klaser, M. Marszalek, and C. Schmid. A spatio-temporal descriptor based on 3d-gradients. In BMVC, 2008.

[3.3.14] A. Kovashka and K. Grauman. Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. In CVPR, 2010.

[3.3.15] I. Laptev. On space-time interest points. IJCV, 2005.

[3.3.16] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In CVPR, 2008.

[3.3.17] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In CVPR, 2006.

[3.3.18] Z. Lin, Z. Jiang, and L. S. Davis. Recognizing actions by shape-motion prototype trees. In ICCV, 2009.

[3.3.19] J. Liu, J. Luo, and M. Shah. Recognizing realistic actions from videos ”in the wild”. In CVPR, 2009.

[3.3.20] J. Liu and M. Shah. Learning human action via information maximization. In CVPR, 2008.

[3.3.21] W.-L. Lu, K. Okuma, and J. J. Little. Tracking and recognizing actions of multiple hockey players using the boosted particle filter. IVC, 2009.

[3.3.22] P. Matikainen, M. Hebert, and R. Sukthankar. Representing pairwise spatial and temporal relations for action recognition. In ECCV, 2010.

[3.3.23] R. Messing, C. Pal, and H. Kautz. Activity recognition using the velocity histories of tracked keypoints. In ICCV, 2009.

[3.3.24] J. C. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. In BMVC, 2006.

[3.3.25] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. TPAMI, 2002.

[3.3.26] K. Rapantzikos, Y. Avrithis, and S. Kollias. Dense saliency-based spatiotemporal feature points for action recognition. In CVPR, 2009.

[3.3.27] M. Raptis, I. Kokkinos, and S. Soatto. Discovering discriminative action parts from mid-level video representations. In CVPR, 2012.

[3.3.28] M. Raptis and S. Soatto. Tracklet descriptors for action modeling and video analysis. In ECCV, 2010.

[3.3.29] M. S. Ryoo and J. K. Aggarwal. Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. In ICCV, 2009.

[3.3.30] S. Satkin and M. Hebert. Modeling the temporal extent of actions. In ECCV, 2010.

[3.3.31] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local svm approach. In ICPR, 2004.

[3.3.32] H. Wang, A. Klaser, C. Schmid, and L. Cheng-Lin. Action recognition by dense trajectories. In CVPR, 2011.

[3.3.33] H. Wang, M. M. Ullah, A. Klaser, I. Laptev, and C. Schmid. Evaluation of local spatio-temporal features for action recognition. In BMVC, 2009.

[3.3.34] G. Willems, T. Tuytelaars, and L. Gool. An efficient dense and scale-invariant spatio-temporal interest point detector. In ECCV, 2008.

[3.3.35] S. Wu, O. Oreifej, and M. Shah. Action recognition in videos acquired by a moving camera using motion decomposition of lagrangian particle trajectories. In ICCV, 2011.

[3.3.36] X. Wu, D. Xu, L. Duan, and J. Luo. Action recognition using context and appearance distribution features. In CVPR, 2011.

[3.3.37] J. Yuan, Z. Liu, and Y. Wu. Discriminative subvolume search for efficient action detection. In CVPR, 2009.

[3.3.38] Y. Zhang, X. Liu, M.-C. Chang, W. Ge, and T. Chen. Spatio-temporal phrases for activity recognition. In ECCV, 2012.

[4.1.1] Hugo Boujut, Jenny Benois-Pineau, and Remi Megret, “Fusion of multiple visual cues for visual saliency extraction from wearable camera settings with strong motion,” in Computer Vision – ECCV 2012. Workshops and Demonstrations, Andrea Fusiello, Vittorio Murino, and Rita Cucchiara, Eds., vol. 7585 of Lecture Notes in Computer Science, pp. 436–445. Springer Berlin Heidelberg, 2012.

[4.1.2] H. Boujut, J. Benois-Pineau, T. Ahmed, O. Hadar, and P. Bonnet, "A metric for no-reference video quality assessment for hd tv delivery based on saliency maps," in Multimedia and Expo (ICME), 2011 IEEE International Conference on, july 2011, pp. 1 –5.

[4.1.3] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (surf). Comput. Vis. Image Underst., 110:346–359, June 2008.

[4.1.4] O. Brouard, V. Ricordel, and D. Barba. Cartes de Saillance Spatio-Temporelle basées Contrastes de Couleur et Mouvement Relatif. In Compression et representation les signaux audiovisuels, 2009.

[4.1.5] C. Cortes and V. Vapnik. Support-vector networks. Machine Learning, 20:273–297, 1995.

[4.1.6] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In Workshop on Statistical Learning in Computer Vision, ECCV, pages 1–22, 2004.

[4.1.7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In C. Schmid, S. Soatto, and C. Tomasi, editors, International Conference on Computer Vision & Pattern Recognition, volume 2, pages 886–893, INRIA Rhône-Alpes, ZIRST-655, av. de l'Europe, Montbonnot-38334, June 2005.

[4.1.8] S. J. Daly. Engineering observations from spatiovelocity and spatiotemporal visual models. In IS&T/SPIE Conference on Human Vision and Electronic Imaging III, 1 1998.

[4.1.9] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascalnetwork.org/challenges/VOC/voc2012/workshop/index.html.

[4.1.10] G. Farnebäck. Fast and accurate motion estimation using orientation tensors and parametric motion models. In Proceedings of 15th International Conference on Pattern Recognition, volume 1, pages 135–139, Barcelona, Spain, September 2000. IAPR.

[4.1.11] A. Fathi, A. Farhadi, and J. M. Rehg. Understanding egocentric activities. In D. N. Metaxas, L. Quan, A. Sanfeliu, and L. J. V. Gool, editors, International Conference in Computer Vision, ICCV 2011, pages 407–414. IEEE, 2011.

[4.1.12] A. Fathi, Y. Li, and J. M. Rehg. Learning to recognize daily actions using gaze. In Proceedings of the 12th European conference on Computer Vision - Volume Part I, ECCV'12, pages 314–327, Berlin, Heidelberg, 2012. Springer-Verlag.

[4.1.13] A. Fathi, X. Ren, and J. M. Rehg. Learning to recognize objects in egocentric activities. In The 24th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, Colorado Springs, CO, USA, 20-25 June 2011, pages 3281–3288. IEEE, 2011.

[4.1.14] P. F. Felzenszwalb, R. B. Girshick, D. A. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(9):1627–1645, 2010.

[4.1.15] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM, 24:381–395, June 1981.

[4.1.16] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In IEEE Conference on Computer Vision and Pattern Recognition, pages 1 –8, June 2008.

[4.1.17] K. Kitani, T. Okabe, Y. Sato, and A. Sugimoto. Fast unsupervised ego-action learning for first-person sports videos. In Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, pages 3241 –3248, june 2011.

[4.1.18] C. H. Lampert, M. B. Blaschko, and T. Hofmann. Beyond sliding windows: Object localization by efficient subwindow search. In 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008), 24-26 June 2008, Anchorage, Alaska, USA. IEEE Computer Society, 2008.

[4.1.19] M. Land, N. Mennie, and J. Rusted. The roles of vision and eye movements in the control of activities of daily living. Perception, 28:1311–1328, 1999.

[4.1.20] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In IEEE Conference on Computer Vision and Pattern Recognition, 2008. CVPR 2008., pages 1 –8, june 2008.

[4.1.21] D. G. Lowe. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 60:91–110, 2004.

[4.1.22] J. J. Moré and D. C. Sorensen. Computing a trust region step. 4(3):553–572, Sept. 1983.

[4.1.23] K. Ogaki, K. M. Kitani, Y. Sugano, and Y. Sato. Coupling eye-motion and ego-motion features for first-person activity recognition. In 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops,2012, pages 1–7. IEEE, 2012.

[4.1.24] P. Over, G. Awad, M. Michel, J. Fiscus, G. Sanders, B. Shaw, W. Kraaij, A. F. Smeaton, and G. QuÃl'enot. Trecvid 2012 – an overview of the goals, tasks, data, evaluation mechanisms and metrics. In Proceedings of TRECVID 2012. NIST, USA, 2012.

[4.1.25] H. Pirsiavash and D. Ramanan. Detecting activities of daily living in first-person camera views. In 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2012.

[4.1.26] V. Sreekanth, A. Vedaldi, C. V. Jawahar, and A. Zisserman. Generalized RBF feature maps for efficient detection. In Proceedings of the British Machine Vision Conference (BMVC), 2010.

[4.1.27] T. Tuytelaars, C. Lampert, M. Blaschko, and W. Buntine. Unsupervised object discovery: A comparison. International Journal of Computer Vision, 88:284–302, 2010.

[4.1.28] S.-F.Wong and R. Cipolla. Extracting spatiotemporal interest points using global information. In International Conference on Computer Vision, 2007. ICCV 2007., pages 1–8, oct. 2007

[4.2.1]. Lavee, G., Rivlin, E., Rudzsky, M. : Understanding Video Events: A Survey of Methods for Automatic Interpretation of Semantic Occurrences in Video. IEEE Transactions on Systems, Man, and Cybernetics. 39 (5). (2009) 489–504

[4.2.2]. Kellokumpu, V., Zhao, G., Pietikinen, M. : Human Activity Recognition Using a Dynamic Texture Based Method. The British Machine Vision Conference (BMVC 2008), Leeds, UK. (2008)

[4.2.3]. Chen, L., Nugent, C. D., Wang, H. : A Knowledge-Driven Approach to Activity Recognition in Smart Homes. IEEE Transactions on Knowledge and Data Engineering. 24 (6). (2012) 961–974

[4.2.4]Xu, D., Chang, S.-F. : Visual Event Recognition in News Video using Kernel Methods with Multi-Level Temporal Alignment. IEEE Transactions on Pattern Analysis and Machine Intelligence. 30 (11). (2008) 1985–1997

[4.2. 5]. Ogale, A. S., Karapurkar, A., Guerra-Filho, G., Aloimonos, Y. : View Invariant identification of Pose Sequences for Action Recognition. Presented ate the Video Analysis Contect Extraction Workshop (VACE). (2004)

[4.2. 6]. Zaidenberg, S. and Boulay, B. and Bremond, F. : A generic framework for video understanding applied to group behavior recognition. The 9th IEEE International Conference On Advanced Video and Signal Based Surveillance (AVSS 12). (2012)

[4.2. 7]. Sadanand, S., Corso, J. J. : Action Bank : A High-Level Representation of Activity in Video. IEEE Conference on Computer Vision and Pattern Recognition (CVPR).(2012) 1234–1241

[4.2. 8]. Summers-Stay, D., Teo, C. L., Yang, Y., Fermller, C., Aloimonos, Y. : Using a Minimal Action Grammar for Activity Understanding in the Real World. IEEE Conference on Intelligent Robots and Systems. 4104–4111 (2012)

[4.2. 9]. Banerjee, T. and Rantz, M. and Li, M. and Popescu, M. and Stone, E. and Skubic, M. and Scott, S. : Monitoring Hospital Rooms for Safety Using Depth Images.Gerontechnology. AI for Gerontechnology. (2012)

[4.2. 10]. Pramerdorfer, C. : Evaluation of Kinect Sensors for Fall Detection. IASTED International Conference. Signal Processing, Pattern Recognition and Applications (SPPRA 2013).

[4.2. 11]. Bak, S., Chau, D. P., Badie, J., Corvee, E., Bremond, F., Thonnat, M. : Multi-Target Tracking by Discriminative Analysis on Riemannian Manifold. ICIP. (2012)

[4.2. 12]. Vu, T., Brmond, F., Thonnat, M. : Automatic Video Interpretation : A Novel Algorithm for Temporal Scenario Recognition. The Eighteenth International Joint Conference on Artificial Intelligence (IJCAI'03), Acapulco, Mexico, (2003) 9-15

[4.2. 13]. Allen J. F. : Maintaining knowledge about temporal intervals. Communications of the ACM. 26 (11). (1983) 832–843

[4.2. 14]. Nghiem,A. T., Brmond, F., Thonnat, M. : Controlling Background Subtraction Algorithms for Robust Object Detection. The Proceedings of the 3rd International Conference on Imaging for Crime Detection and Prevention, ICDP 09, Kingston University, London, UK, (2009)

[4.2. 15]. Chau, D. P., Bremond, F., Thonnat, M. : A multi-feature tracking algorithm enabling adaptation to context variations. In the Imaging for Crime Detection and Prevention Conference (ICDP 2011), Kingston University, London, UK, (2011)

[5.2.1] Jacqueline Kerr, Simon J. Marshall, Suneeta Godbole, Jacqueline Chen, Amanda Legge, Aiden R. Doherty, Paul Kelly, Melody Oliver, Hannah M. Badland and Charlie Foster. Using the SenseCam to Improve Classifications of Sedentary Behavior in Free-Living Settings. American Journal of Preventive Medicine, Volume 44, Issue 3, Pages 290–296. 2013.

[5.3.1] Mann, S. (1997). Wearable Computing: A first step towards personal imaging. Computer .

[5.3.2] Sellen, A., Fogg, A., Aitken, M., Hodges, S., Rother, C., & Wood, K. (2007). Do life-logging technologies support memory from the past? An experimental study using SenseCam. CHI .

[5.3.3] Eagle, N., & Pentland, A. (2006). Reality Mining: Sensing complex social systems. Personal Ubiquitous Computing .

[5.3.4] Vemuri, S., & Bender, W. (2004). Next-generation personal memory aids. BT Technology Journal .

[5.3.5] Patterson, D., Lao, L., Gajos, K., Collier, M., Livic, N., Olson, K., et al. (2004). Opportunity Knocks: A System to Provide Cognitive Assistance with Transportation Services. UbiComp .

[5.3.6] Reddy, S., Parker, A., Hyman, J., Burke, J., Estrin, D., & Hansen, M. (2007). Image Browsing, Processing, and Clustering for Participatory Sensing: Lessons from a DietSense prototype. Embedded Network Sensors (EmNets07) .

[5.3.7] Bukhin, M., & DelGaudio, M. (2006). WayMarkr: Acquiring Perspective through continuous documentation. Mobile and Ubiquitous Media (MUM 06) .

[5.3.8] Albanesius, Chloe. Google 'Project Glass' Replaces the Smartphone With Glasses". PC Magazine. Retrieved, 2012.

[5.3.9] http://memoto.com/ (12-04-2013)

[5.3.10] L. Bao and S. S. Intille, "Activity recognition from user-annotated acceleration data," Lecture Notes in Computer Science, pp. 1-17, 2004.

[5.3.11] Smeulders, A. W., Worring, M., Santini, S., Gupta, A., & Jain, R. (2000). Content-based Image Retrieval at the end of the early years. IEEE Transactions Pattern Analysis and Machine Intelligence .

[5.4.1] Ó Conaire, C., O'Connor, N., Smeaton, A., & Jones, G. (2007). Organising a Daily Visual Diary Using Multi-Feature Clustering. SPIE Electronic Imaging .

[5.4.2] Doherty, A., & Smeaton, A. (2008). Automatically Segmenting LifeLog Data Into Events. WIAMIS.

[5.4.3] Doherty, A., Smeaton, A., Lee, K., & Ellis, D. P. (2007). Multimodal Segmentation of Lifelog Data. RIAO .

[5.4.4] Aghazadeh, O., Sullivan, J., & Carlsson, S. (In Press). Novelty Detection from an Ego-Centric Perspective.

[5.4.5] Snoek, C. G., Huurnink, B., Hollink, L., Rijke, M. D., Schreiber, G., & Worring, M. (2007). Adding Semantics to Detectors for Video Retrieval. IEEE Transactions on Multimedia.

[5.4.6] Li, J., & Wang, J. Z. (2003). Automatic Linguistic Indexing of Pictures by a statistical modelling approach. IEEE Transactions on Pattern Analysis and Machine Intelligence .

[5.4.7] Lakoff, G. (1990). Women, Fire, and Dangerous Things. University of Chicago Press.

[5.4.8] Huurnink, B., Hofmann, K., & de Rijke, M. (2008). Assessing concept selection for video retrieval. Proceedings Multimedia Information Retrieval (MIR08).

[5.4.9] Byrne, D., Doherty, A. R., Snoek, C. G., Jones, G. J., & Smeaton, A. F. (2010). Everyday Concept Detection in Visual Lifelogs: Validation, Relationships and trends. Multimedia Tools and Applications.

[5.4.10] Wang, Peng (2011). Semantic Interpretation of Events in Lifelogging (PhD Thesis). Dublin City University.

[5.4.11] Wang, P., & Smeaton, A. F. (In Press). Semantically Enhancing Lifelog Events based on Multiple Contexts. Personal and Ubiquitous Computing .

[5.4.12] Wang, P., & Smeaton, A. F. (In Press). Semantics-Based Selection of Everyday Concepts in Visual Lifelogging. International Journal of Multimedia Information Retrieval .

[5.4.13] Byrne D, Doherty A.R., Snoek C.G.M., Jones G.F., and Smeaton A.F. “Validating the Detection of Everyday Concepts in Visual Lifelogs”. SAMT 2008 – 3rd International Conference on Semantic and Digital Media Technologies, Koblenz, Germany, 3-5 D, 2008.

[5.4.14] Doherty, Aiden R. “Providing effective memory retrieval cues through automatic structuring and augmentation of a lifelog of images”. PhD thesis, Dublin City University, 2009.

[5.4.15] Daragh Byrne, Barry Lavelle, Aiden R. Doherty, Gareth J.F. Jones, & Alan F. Smeaton. “Using Bluetooth & GPS Metadata to Measure Event Similarity in SenseCam Images”. Centre for Digital Video Processing (CDVP) & Adaptive Information Cluster (AIC), Dublin City University, Dublin 9, Ireland, 2007.

[5.4.16] Kikhia, Basel. “Supporting Lifestories through Activity Recognition and Digital Reminiscence Recognition and Digital Reminiscence”. Licentiate thesis, Luleå University of Technology, 2011.

[5.4.17] J. Y. Yang, J. S. Wang and Y. P. Chen, "Using acceleration measurements for activity recognition: An effective learning algorithm for constructing neural classifiers," Pattern Recog. Lett., vol. 29, pp. 2213-2220, 2008.

[5.4.18] L. Atallah, B. Lo, R. King and G. Z. Yang, "Sensor positioning for activity recognition using wearable accelerometers," Biomedical Circuits and Systems, IEEE Transactions on, vol. 5, pp. 320-329, 2011.

[5.4.19] L. Bao and S. S. Intille, "Activity recognition from user-annotated acceleration data," Lecture Notes in Computer Science, pp. 1-17, 2004.

[5.4.20] S. J. Preece, J. Y. Goulermas, L. P. J. Kenney, D. Howard, K. Meijer and R. Crompton, "Activity identification using body-mounted sensors—a review of classification techniques," Physiol. Meas., vol. 30, pp. R1, 2009.

[5.4.21] Chris D. Nugent, Xin Hong, Josef Hallberg, Dewar D. Finlay, Kåre Synnes: Assessing the impact of individual sensor reliability within smart living environments. CASE 2008: 685-690. 2008

[5.4.22] Mckeever, Susan. "Recognising Situations Using Extended Dempster-Shafer Theory". PhD thesis, University College Dublin, Ireland 2011.

[5.4.23] Mohamed Tarik Moutacalli, Abdenour Bouzouane, Bruno Bouchard, "Unsupervised Activity Recognition using Temporal Data Mining". In SMART 2012, The First International Conference on Smart Systems, Devices and Technologies, Germany. ISBN: 978-1-61208-225-7. 2012.

[5.5.1] Don Koks, Subhash Challa, An introduction to Bayesian and Dempster-Shafer data fusion. DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION SALISBURY (AUSTRALIA) SYSTEMS SCIENCES LAB, Technical rept. 2003.

[5.5.2] R. R. Murphy. Dempster-Shafer theory for sensor fusion in autonomous mobile robots. Robotics and Automation, IEEE Transactions on, 14(2):197– 206, 1998.

[5.5.3] Shafer, Glenn; A Mathematical Theory of Evidence, Princeton University Press, ISBN 0-608-02508-9, 1976.

[5.5.4] Dempster AP. Upper and lower probabilities induced by a multi-valued mapping [M].Annals of Mathematical Statistics, 1967, 38:325-339.

[5.5.5] Shafer, Glenn; A Mathematical Theory of Evidence, Princeton University Press, 1976, ISBN 0-608-02508-9

[5.5.6] [3]Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian network classifiers. Machine Learning (1997) 131–163

[5.5.7] [5] Catherine K. Murphy. Combining belief functions when evidence con- flicts. Decis. Support Syst., 29(1):1–9, 2000.

[5.5.8] J. D. Lowrance, T. D. Garvey, and T. M. Strat. A framework for evidential- reasoning systems, pages 611–618. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990