SURREY UNIVERSITY

Video Processing and Background Subtraction for Change Detection and Activity Recognition

Konstantinos AVGERINAKIS

A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy in the
Centre for Vision, Speech and Signal Processing
Department of Electronic Engineering

March 2015
Declaration of Authorship

I, Konstantinos Avgerinakis, declare that this thesis titled, 'Video Processing and Background Subtraction for Change Detection and Activity Recognition' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:  

Date:  

---
"As you set out for Ithaka hope the voyage is a long one, full of adventure, full of discovery. Laistrygonians and Cyclops, angry Poseidon—don’t be afraid of them: you’ll never find things like that on your way as long as you keep your thoughts raised high, as long as a rare excitement stirs your spirit and your body. Laistrygonians and Cyclops, wild Poseidon—you won’t encounter them unless you bring them along inside your soul, unless your soul sets them up in front of you.

Hope the voyage is a long one. May there be many a summer morning when, with what pleasure, what joy, you come into harbors seen for the first time; may you stop at Phoenician trading stations to buy fine things, mother of pearl and coral, amber and ebony, sensual perfume of every kind—as many sensual perfumes as you can; and may you visit many Egyptian cities to gather stores of knowledge from their scholars.

Keep Ithaka always in your mind. Arriving there is what you are destined for. But do not hurry the journey at all. Better if it lasts for years, so you are old by the time you reach the island, wealthy with all you have gained on the way, not expecting Ithaka to make you rich.

Ithaca gave you the marvelous journey. Without her you would not have set out. She has nothing left to give you now.

And if you find her poor, Ithaka won’t have fooled you. Wise as you will have become, so full of experience, you will have understood by then what these Ithakas mean.”

C. P. Kavafy
Abstract
Faculty of Engineering and Physical Sciences
Department of Electronic Engineering
Doctor of Philosophy

Video Processing and Background Subtraction for Change Detection and Activity Recognition
by Konstantinos Avgerinakis

The abrupt expansion of the Internet use over the last decade led to an uncontrollable amount of media stored in the Web. Image, video and news information has flooded the pool of data that is at our disposal and advanced data mining techniques need to be developed in order to take full advantage of them. The focus of this thesis is mainly on developing robust video analysis technologies concerned with detecting and recognizing activities in video.

The work aims at developing a compact activity descriptor with low computational cost, which will be robust enough to discriminate easily among diverse activity classes. Additionally, we introduce a motion compensation algorithm which alleviates any issues introduced by moving camera and is used to create motion binary masks, referred to as compensated Activity Areas (cAA), where dense interest points are sampled. Motion and appearance descriptors invariant to scale and illumination changes are then computed around them and a thorough evaluation of their merit is carried out.

The notion of Motion Boundaries Activity Areas (MBAA) is then introduced. The concept differs from cAA in terms of the area they focus on (ie human boundaries), reducing even more the computational cost of the activity descriptor. A novel algorithm that computes human trajectories, referred to as ‘optimal trajectories’, with variable temporal scale is introduced. It is based on the Statistical Sequential Change Detection (SSCD) algorithm, which allows dynamic segmentation of trajectories based on their motion pattern and facilitates their classification with better accuracy.

Finally, we introduce an activity detection algorithm, which segments long duration videos in an accurate but computationally efficient manner. We advocate Statistical Sequential Boundary Detection (SSBD) method as a means of analysing motion patterns and report improvement over the State-of-the-Art.
Acknowledgements

First and foremost, I would like to thank my principal supervisor Dr. Josef Kittler for giving me the opportunity to cooperate and develop my work under his supervision. I am also deeply grateful to Dr. Maria Petrou who helped me throughout my initial steps and provide me bright insights when needed.

I am really thankful to Dr. Yiannis Kompatsiaris for trusting and supporting me on completing this painful work throughout all these years, while, special thanks should be also given to Dr. Alexia Briassouli for her plenty time when needed and her contribution on the majority of my work.

Finally, I feel companionate and thankful for all my close ones who supported me and being there when needed...

This work was supported by the following projects:

FP7-214306 JUMAS - Judicial Management by Digital Libraries Semantics

FP7-288199 Dem@Care - Dementia Ambient Care: Multi-Sensing Monitoring for Intelligent Remote Management and Decision Support.

A-0120-RT-GC MEDUSA - Multi SEnsor Data Fusion Grid for Urban Situational Awareness
Contents

Declaration of Authorship i

Abstract iii

Acknowledgements iv

Contents v

List of Figures viii

List of Tables xiii

Abbreviations xv

1 Introduction 1
1.1 Activities of Daily Living (ADL) 2
1.2 Motivation 5
1.3 Contributions 6
  1.3.1 Computationally Efficient Dense Sampling 6
  1.3.2 A Hybrid Global-Local Descriptor 7
  1.3.3 A Time Invariant Trajectory 7
  1.3.4 Fisher encoding for Improved Recognition 7
  1.3.5 Fast activity localisation 8
  1.3.6 ADL Recognition Dataset 8
1.4 Structure of thesis 9

2 Literature Review 10
2.1 Activity representation 10
  2.1.1 Global activity representation 11
  2.1.2 Local activity representation 14
2.2 Activity detection 17
2.3 Activity evaluation benchmark datasets 19
  2.3.1 KTH action dataset 20
  2.3.2 UCF action dataset 22
  2.3.3 Hollywood1 action dataset 22
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.4.3.1</td>
<td>KTH results</td>
<td>78</td>
</tr>
<tr>
<td>4.4.3.2</td>
<td>URADL results</td>
<td>80</td>
</tr>
<tr>
<td>4.4.3.3</td>
<td>KiT results</td>
<td>82</td>
</tr>
<tr>
<td>4.4.4</td>
<td>DemCare results</td>
<td>82</td>
</tr>
<tr>
<td>4.4.5</td>
<td>CHUN results</td>
<td>85</td>
</tr>
<tr>
<td>4.5</td>
<td>Discussion</td>
<td>85</td>
</tr>
<tr>
<td>5</td>
<td>Fast activity detection via Statistical Sequential Change Detection (SSCD) using CUSUM</td>
<td>89</td>
</tr>
<tr>
<td>5.3.1</td>
<td>Sliding window activity detection</td>
<td>95</td>
</tr>
<tr>
<td>5.3.2</td>
<td>Statistical Sequential Boundary Detection</td>
<td>97</td>
</tr>
<tr>
<td>5.3.2.1</td>
<td>Practical implementation of CUSUM</td>
<td>101</td>
</tr>
<tr>
<td>5.4.1</td>
<td>Spatial localisation</td>
<td>104</td>
</tr>
<tr>
<td>5.4.2</td>
<td>Temporal localisation</td>
<td>105</td>
</tr>
<tr>
<td>5.4.2.1</td>
<td>DemCare action dataset</td>
<td>105</td>
</tr>
<tr>
<td>5.4.2.2</td>
<td>CHUN action dataset</td>
<td>110</td>
</tr>
<tr>
<td>5.5</td>
<td>Conclusion</td>
<td>115</td>
</tr>
<tr>
<td>6</td>
<td>Conclusions and Future Work</td>
<td>117</td>
</tr>
<tr>
<td>6.1</td>
<td>Conclusions</td>
<td>117</td>
</tr>
<tr>
<td>6.2</td>
<td>Future work</td>
<td>118</td>
</tr>
</tbody>
</table>

A Original publications and references

B Clustering algorithms for Visual Vocabulary

B.1 K-means clustering

B.2 Gaussian Mixture Model (GMM) clustering
List of Figures

1.1 Youtube and public video competitors data growth over the last decade [1]. 2
1.2 Worldwide population growth rate [2]. 3

2.1 Examples of model based approaches centered on human body parts detection and tracking. Top row basketball pose is taken from [3], while medium and bottom row poses are taken from [4]. 12
2.2 Motion energy (MEI) and Motion History Images (MHI) for a dance pirouette taken from [5]. 12
2.3 Spatio-temporal volumes as described in [6] illustrating actions taken from the Weizzman dataset are depicted in the left picture, while 3D voxel models, proposed in [7] are depicted in the right picture. 13
2.4 The motion descriptor proposed in [8]. (a) Original image, (b) Optical flow, (c) Separating the x and y components of optical flow vectors, (d) Half wave rectification of each component to produce 4 separate channels, (e) Final blurry motion channel. 14
2.5 Spatio-Temporal Interest Point (STIP) or Harris3D detector introduced on [9]. 15
2.6 Correlations between salient features by using a KLT tracker. The left picture have reproduced from [10], while the right pictures is taken from [11]. 17
2.7 Spatio-temporal pyramids for a weak order BoVW system in [12] 17
2.8 Sliding central frame temporal localisation taken from [13]. 19
2.9 Examples from KTH action dataset [14]. 22
2.10 Examples from UCF sports action dataset [15]. 22
2.11 Examples from hollywood (HOHA) action dataset [12]. 23
2.12 Examples from URADL action dataset [11]. 24
2.13 KIT action dataset video samples [16]. On the top row we depict samples from counter-top setup, while bottom row has samples from room setup. 25
2.14 DemCare action dataset video samples [17]. Faces have been blurred for ethics purposes. 28
2.15 CHUN action dataset video samples 28

3.1 Activity representation using motion compensation on OF, multiscale dense sampling on cAA regions and HOGHOF trajectory descriptor around interest points. 33
3.2 Concatenated compensated OF (magnitude) inside (blue line) and outside (green line) cAAs from 18 UCF sports videos. Compensated OF in cAAs is higher and more irregular than outside cAAs. 37
3.3 The plot is used for KS-test visualisation and shows the similarity between the empirical-cdf of the centered and scaled compensated OF inside (blue line) and outside (green line) cAA and the cdf of the standard normal distribution (red). The RMSD between the Gaussian model of the compensated OF and the compensated OF distribution inside cAA is higher than outside it, showing that the compensated OF follows a Gaussian distribution for static pixels (All cdfs have been shifted to x-axis so that the standard normal cdf could be centered to zero when cumulative probability equals to 0.5).

3.4 Optical flow (top), compensated optical flow (middle row) and compensated Activity Areas (bottom row). Dominant motion is successfully removed from the foreground, while some noise is still retained. cAA masks successfully isolate regions of human motion.

3.5 Trajectory extraction inside cAA and HOGHOF+Traj representation.

3.6 Training visual descriptors are clustered using K-Means or GMM algorithm and based on the extracted means (or cluster centers accordingly) appropriate encoding is performed (VLAD, Fisher, Chi-Square). Training encoded data are used to train a SVM classifier, while test encoded data evaluate its discriminative power.

3.7 Average Accuracy over all classes when cAA applied against dense sampling when applied on KTH action dataset. HOGHOF descriptors are used on the left figure, while HOGHOF+Traj in the right one.

3.8 Average Accuracy over all classes for each encoding scheme. Comparison among HOGHOF and HOGHOF+Traj is presented in each case.

3.9 Average Accuracy over all classes when cAA applied against dense sampling on UCF sports action dataset. HOGHOF+Traj is depicted on the left figure, while HOGHOF is used on the right one.

3.10 HOGHOF and HOGHOF+Traj comparisons for each encoding scheme.

3.11 Average Accuracy over all classes when cAA applied against dense sampling on HOHA action dataset. HOGHOF descriptors results are depicted on the left figure, while HOGHOF+Traj on the right one.

3.12 Average Accuracy over all classes when Activity Area applied against dense sampling. HOGHOF+Traj descriptors were used on the left figure, while HOGHOF was used in the right one.

3.13 HOHA bars.

3.14 Average Accuracy over all classes when cAA applied against dense sampling on YouTube action dataset. HOGHOF descriptors results are depicted on the left figure, while HOGHOF+Traj on the right one.

3.15 Average Accuracy over all classes when Activity Area applied against dense sampling. HOGHOF+Traj descriptors were used on the left figure, while HOGHOF was used in the right one.
4.1 Block diagram for activity of daily living method overview. Orange blocks denote the novel steps on the recognition algorithm. MBAAs are extracted based on motion information and dense interest points are sampled in them. Appearance (HOG) and motion (HOF) descriptors are extracted at several scales around the interest points, which are tracked to form a trajectory and the final spatio-temporal descriptor. Trajectory length is defined based on the motion pattern (HOF) statistics using SSCD.

4.2 Block diagram for ADL visual features representation. OF estimates are used to extract MBAAs, dense sampling within MBAAs on four spatial scales results in interest points that are tracked by KLT and construct trajectories. Outliers are removed from trajectories by the application of RANSAC and HOGHOF features are derived on multiple scales around the tracked points. CUSUM SSCD is applied to the resulting trajectories’ HOFs to detect changes in motion and form final ADL representation.

4.3 K-S test comparing empirical cdf with Gaussian approximation cdf for pixels inside MBAAs(top) and pixels outside MBAAs(bottom).

4.4 Optical flow data distribution of the two regions (i.e. inside and outside MBAA). It is obvious that our data can easily be distinguished as they assume values in different range.

4.5 The kurtosis of OF gradients corresponding to a changing motion (top Fig.) is higher than in regions outside an MBAA (bottom Fig.).

4.6 MBAAs are much smaller than AAs but are located in regions of changing motion that are informative about the activity taking place.

4.7 Averaging for constructing the spatio-temporal descriptor of fixed length from input data of varying temporal extents.

4.8 Optimal trajectories depicted for several activities. Answer phone, eat snack, eat banana are depicted for URADL in the top row, while handwave, jog and run are depicted from KTH in the bottom. Green lines denote the extra extension from fixed trajectory with temporal window of 15 frames.

4.9 Block diagram for ADL recognition. Training data is used to construct a vocabulary, via K-means or GMM clustering. Test data is similarly encoded. This is followed by hard binning or Fisher vector encoding respectively. Chi-square kernel and/or Hellinger kernels are used in an SVM scheme for recognition.

4.10 Optimal trajectories (blue lines) & hybrid descriptors (green rectangles) for KTH action dataset. Box, handclap, handwave, jog, run and walk are depicted from left to right.

4.11 Optimal trajectories (red lines) & hybrid descriptors (green rectangles) for URADL action dataset.

4.12 Characteristic pictures for KIT action dataset.

4.13 Optimal trajectories (red lines) & hybrid descriptors (green rectangles) for DemCare action dataset.

4.14 Optimal trajectories (yellow-reddish lines) & hybrid descriptors (green rectangles) for CHUN action dataset.
5.1 A person enters a smart home, where the presence of MBAAs indicate the start of the activity sequence, which ends when the trajectories are terminated. Spatio-temporal sliding windows would need to process the whole sequence using 2 multi-scale overlapping windows, while SSBD builds a reference pattern and collects test data from new frames to detect changes in their pattern online. SSBD thus reduces the computational cost by avoiding the sequential application of SVMs for activity detection, classifying instead the subsequences only after a change has been found.

5.2 Block diagram of the ADL representation schema. Kurtosis analysis on OF values results in Motion Boundary Activity Areas (MBAAs), where dense sampling is performed. Trajectory and local histograms (HOGHOF) are constructed around interest points in the MBAAs on multiple scales, leading to the final ADL descriptor.

5.3 Block diagram of ADL detection and recognition. Training pre-segmented samples are used to build visual vocabularies (i.e. cluster centers or means) and all data is encoded into a single size feature vector based on the VLAD/Fisher framework. \( C(C - 1)/2 \) multi-class SVM classifiers are built and a sliding window filters the unsegmented test data, in order to detect and recognise the ADLs and ambiguous temporal segments that occur within them.

5.4 Activity Areas on the left and trajectories on the right of each frame depict the start and end of a video segment.

5.5 Ambiguous status is demonstrated among two discriminant ADLs (pay bill after preparing drugs).

5.6 SVDD on a toy dataset depicting the hypersphere around data when a Gaussian with different widths \( (\sigma = 1, 5, 15) \) and different values for \( C \) (\( C = 0.1 \) and \( C = 25 \)) are applied. Support vectors are indicated by the solid circles, the solid white line is the description boundary.

5.7 SSBD analysis when a change on the activity pattern occurs (Top), and when is not (Bottom).

5.8 Characteristic video frames for enter room (ER), handshake (HS) and serve beverage (SB) ADLs in the top row. Drink Beverage (DB), prepare snack (PS) and read paper (RP) in the bottom row, taken from the DemCare dataset from left to right.

5.9 HOGHOF and HOGHOF+Traj comparison when VLAD(left) and Fisher(right) encoding is applied on DemCare action dataset.

5.10 Overlap performance measure for varying \( W_0 \) when HOGHOF+Traj is combined with Fisher encoding of 64 cluster centers.

5.11 Average accuracy for different overlap ratios when HOGHOF and HOGHOF+Traj applied on DemCare dataset.

5.12 Classification by detection results. Average accuracy for HOGHOF and HOGHOF+Traj descriptors are provided for OV20 on the DemCare dataset.

5.13 Precision Recall curves over all classes for DemCare dataset.

5.14 CHUN dataset sample ADLs. From left to right and top to bottom we have: answer phone, look on map, pay bill, prepare drugs, prepare tea, read paper, water plant and watch TV.

5.15 HOGHOF and HOGHOF+Traj comparison when VLAD(left) and Fisher(right) encoding is applied on the CHUN action dataset.
5.16 Classification by detection results. Average accuracy for HOGHOF and HOGHOF+Traj descriptors are provided for OV20 on the CHUN dataset. 114
5.17 Average accuracy for different overlap ratios when HOGHOF and HOGHOF+Traj and SSBD are applied on the CHUN dataset . . . . . . . . . . . . . . . . 115
5.18 Precision Recall curves over all classes for CHUN dataset. . . . . . . . . . . 116
# List of Tables

2.1 List of human activity video datasets and their properties. .......................... 21
2.2 The set of activities that are observed in DemCare1 and DemCare2 action
datasets and their description ................................................................. 27
3.1 UCF Sports average IoU for spatial localisation. ........................................ 42
3.2 TV Human Interaction IoU for spatial localisation. ...................................... 43
3.3 Confusion matrix of our best result for the KTH videos. Accuracy rates are
recorded as the ratio of the correctly recognized actions to the groundtruth
ones ........................................................................................................... 45
3.4 KTH recognition results: Aggregated average accuracy rates over all classes. 46
3.5 UCF aggregated recognition rates (i.e. percentage of correct classification)
by measuring average accuracy over all classes. ........................................ 47
3.6 Confusion matrix for UCF action dataset when HOGHOF+Traj encoded
with Fisher of 32 cluster centers. Recognition rates are recorded as ratio
of correct classified actions to groundtruth ones. ....................................... 48
3.7 Mean average precision over varying vocabulary sizes for BoVW, Fisher
and VLAD encoding, tested on Hollywood action dataset (HOHA). ............. 50
3.8 Mean average precision over varying vocabulary sizes for BoVW, Fisher
and VLAD encoding, tested on YouTube action dataset. ................................ 52
3.9 Comparison to SoA average accuracy over all classes for the three action
datasets. ................................................................................................... 53
3.10 Time consumption on the three action datasets ........................................... 53
4.1 Mean Kurtosis for 10 videos from the URADL dataset for pixels inside
and outside MBAA (i.e. Noisy). ............................................................... 64
4.2 KTH ADL recognition accuracy with confidence intervals (CI) and pro-
cessing speed in frames per second (fps) for all activities and different
combinations of appearance (HOG), motion (HOF) and trajectory (traj,
SSCD) descriptors, using the Optical Flow (OF) methods of [18] and [19].
Comparison with the SoA methods [20], [12], [21], [22], where applicable. . 79
4.3 URADL recognition accuracy with confidence intervals (CI) and processing
speed in frames per second (fps) for all activities and combinations of
appearance (HOG), motion (HOF) and trajectory (traj, SSCD) descriptors,
using the OF of [18] and [19]. All experiments are on MBAA data,
except the columns [18]_AA. Comparison with the SoA methods Messing:URADL, [21] with HOGHOF, [21] with MBH. ................. 81
4.4 KIT Counter top setup activity recognition accuracy with confidence
intervals and comparison with the SoA [16]. ............................................. 83
4.5 KIT Room Setup activity recognition accuracy with confidence intervals and comparison with the SoA [16]. ........................................ 83
4.6 DemCare recognition accuracy rates over all activity classes for: different combinations of appearance (HOG), motion (HOF), trajectory (traj, SSCD), the Optical Flow (OF) methods of [19] and [18] and comparisons with the SoA Wang:DenseTraj with HO GHOF and [21] with MBH........ 87
4.7 Recognition rates on CHUN action dataset. ................................. 88
5.1 Intersection over Union over all classes for UCF action dataset ........ 105
5.2 Average accuracy over all classes for DemCare action dataset .......... 107
5.3 Confusion matrix for ADL recognition, using average accuracy scores over all classes, on the DemCare dataset .................................................. 107
5.4 The effect of varying $h$ SSBD sensitivity threshold for HO GHOF+Traj combined with Fisher of 64 cluster centers. ............................. 109
5.5 Average accuracy, computational cost, and frame per second algorithm performance for HO GHOF,HO GHOF+Traj without SSBD and HO GHOF+Traj with SSBD. Standard Jaccard coefficient of 20% overlap (OV20) is recorded as average accuracy metric. ........................................ 109
5.6 Average accuracy over all classes for the CHUN dataset .................. 113
5.7 ADL recognition in terms of average accuracy on the CHUN dataset ... 113
5.8 Average accuracy, computational cost, and frame per second algorithm performance for HO GHOF,HO GHOF+Traj without SSBD and HO GHOF+Traj with SSBD. Not only SSBD increases the overall average accuracy rates, but also it decreases the computational cost. .............................. 114
## Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cAA</td>
<td>compensated Activity Areas</td>
</tr>
<tr>
<td>HOG</td>
<td>Histograms of Oriented Gradients</td>
</tr>
<tr>
<td>HOF</td>
<td>Histograms of Oriented Optical Flow</td>
</tr>
<tr>
<td>MBAA</td>
<td>Motion Boundary Activity Areas</td>
</tr>
<tr>
<td>SSCD</td>
<td>Statistical Sequential Change Detection</td>
</tr>
<tr>
<td>SSBD</td>
<td>Statistical Sequential Boundary Detection</td>
</tr>
<tr>
<td>BoVW</td>
<td>Bag of Visual Words</td>
</tr>
<tr>
<td>VLAD</td>
<td>Vector of Locally Aggregated Descriptors</td>
</tr>
<tr>
<td>AAL</td>
<td>Ambient Assisted Living</td>
</tr>
<tr>
<td>ADL</td>
<td>Activity of Daily Living</td>
</tr>
<tr>
<td>PwD</td>
<td>People with Dementia</td>
</tr>
<tr>
<td>SoA</td>
<td>State of the Art</td>
</tr>
</tbody>
</table>
Dedicated to my family…
Chapter 1

Introduction

Contents

1.1 Activities of Daily Living (ADL) ........................................... 2
1.2 Motivation ................................................................. 5
1.3 Contributions .............................................................. 6
   1.3.1 Computationally Efficient Dense Sampling ...................... 6
   1.3.2 A Hybrid Global-Local Descriptor ............................... 7
   1.3.3 A Time Invariant Trajectory .................................... 7
   1.3.4 Fisher encoding for Improved Recognition .................... 7
   1.3.5 Fast activity localisation ....................................... 8
   1.3.6 ADL Recognition Dataset ..................................... 8
1.4 Structure of thesis ....................................................... 9

The rapid growth of the Internet in the past few years has led to a great increase of the multimedia data available on the web. Youtube announced that over 100 hours of video are uploaded to YouTube every minute for 2013 (Figure 1.1), while 12 billion views were recorded for all video sites in 2009. The spiraling increase of multimedia on the web is making the exploitation and management of their information of paramount importance. Concurrently, a great deal of online and offline applications require the detection and recognition of certain events in a set of videos. So we have the spatio-temporal localisation of rugby tackles on a soccer game or the number of handshakes in a news footage, the localisation of different dance movements in a video chorography and the monitoring of shopping malls for the detection of suspicious activity, the adoption of surveillance cameras in elderly or smart homes can all be considered as some characteristic examples that activity recognition can be applied. Recently, human-computer interaction and games remote control has also gained control among the computer vision society with the introduction of cheap and compact depth cameras (e.g. Kinect, Asus). For this
reason, activity detection and recognition methods are being developed, focusing on automatically determining what types of events are occurring in a video, classifying video and so on.

The use of multimedia has also increased in healthcare in the past years, with deployments of various sensing and feedback capabilities in numerous applications, ranging from those of a clearly medical orientation to providing support in ambient assisted living situations. In recent years, targeted efforts have been made to deploy multi-modal sensing for the particular use case of monitoring and providing feedback to people with dementia (PwD), who are at a high risk of losing their independence as their condition progresses. Multi-modal sensing can provide emergency detection, but also a clear picture of the health status of PwD if physiological sensors are being used. Additionally, its role in building behavioural and lifestyle profiles can be of great help in detecting subtle changes in lifestyle that may indicate a deterioration of the person’s condition.

### 1.1 Activities of Daily Living (ADL)

The worldwide increase in life expectancy (Figure 1.2) entails age-related health issues, multiplying healthcare costs every year. Technologies that monitor activities of daily living (ADL) can allow a person with chronic degenerative conditions, such as dementia, to remain independent, reducing the burden on family/friends and decreasing healthcare costs. Ambient Assisted Living (AAL) solutions are being developed to help people with chronic degenerative conditions continue living independently for as long as they can. This is achieved in large part by continuous unobtrusive monitoring of activity, lifestyle and behavioral profiling, which ensures their safety in case of an emergency, as well as the detection of gradual changes in their condition. The results of the monitoring and profiling provided by such systems can then be used as feedback both for the people being monitored, as well as for their carers.
Chapter 1. Introduction

Monitoring of daily life can take place with numerous sensors, both ambient and wearable, ranging from ambient video cameras to wearable accelerometers [23], [24], environmental [25], physiological, audio sensors [26], electrooculography [27] and various combinations thereof [28] [29], [30]. The advantage of wearable sensors, e.g. inertial sensors (accelerometers), is that they can be deployed in various environments and can provide information in unconstrained real life conditions [31], while less complicated sensors like accelerometers also entail a lower computational cost. Their drawbacks are that the user may forget to wear them, they may be obtrusive and interfere with daily activities/routines, and they cannot provide a complete picture of the activities taking place. Ambient sensors, on the other hand, are limited to specific locations and environments (e.g. in a person’s home), but have the advantage of providing more activity information and lifestyle profiling. The scope of this work is the accurate recognition of human activities from ambient visual sensors, however these results can be fused with other sensor modalities’ results in future work, for highly accurate, comprehensive, robust monitoring.

Visual sensors can help achieve more detailed and accurate human activity recognition than environmental or physiological sensors [32], as they provide rich information about the activities taking place, leading to high recognition rates, and improving a recognition system’s robustness when fused with other sensor data [33], [34]. In [35], ambient visual sensors are wall-mounted for human identity sensing in smart spaces, while the introduction of color-depth cameras like the Kinect has enabled the more efficient solution of difficult computer vision problems and the more accurate and robust recognition of human activities [36].

The deployment of visual sensors for daily life monitoring can be challenging, as users may refuse the installation of cameras in their home due to privacy concerns. However,
video-based human activity recognition is already taking place in video game consoles that people have in their homes, like the Nintendo Wii and the Microsoft Kinect, which interact with human gestures and even entire human body movements. It should also be emphasised that the automatic activity recognition described in this work does not require viewing or transmitting the raw video data, since it is based on the processing of motion/appearance features, automatically extracted from the video. This further protects privacy, as the videos can remain completely inaccessible to any viewer, without hindering activity recognition. Therefore, it is expected that the presence of these sensors in peoples’ homes, the appropriate education of the end users on the way their data is kept safe and private, as well as on the benefits of the daily activity monitoring provided, will increase acceptance and familiarity with visual sensing in the home. In cases where acceptance for visual sensors in the home is unsurmountable, the human activity recognition proposed in this work can be used, for example, in appropriately designed home-like environments in hospitals. This has already taken place for EU project Dem@Care (www.demcare.eu) in the Centre Hospitalier Universitaire de Nice (CHUN) in Nice, France and the Greek Association for Alzheimer’s Disease and Related Disorders (GAADRD) in Thessaloniki, Greece, leading to benchmark recordings of activities of daily living [17].

In this work we focus, amongst others, on the use of ambient (static) video for remote monitoring of people with chronic degenerative conditions, like dementia and other degenerative disorders, living home alone. For effective video-based monitoring, the recognition of ADLs is central, and accordingly it is a practical focus of this work. Activity recognition from video for assisted living is based on unobtrusive ambient sensors, namely static video cameras, which do not disturb people in their daily life. It is essential to provide highly accurate recognition results to learn activity, lifestyle and behavioural patterns for each human subject, helping their carer remotely monitor the progress of their condition, so as to support them accordingly. Activity recognition in computer vision mainly focuses on extracting information from pre-segmented video sequences, i.e. videos annotated by user a priori (i.e. videos that have been previously segmented into subsequences containing one activity each), which makes it inappropriate for dealing with real scenarios where videos are not segmented beforehand. In practical situations, activity recognition needs to be preceded by activity localisation, which localises in spatio-temporally an action of potential interest in a video. Also, in real life scenarios, scene conditions are challenging and diverse and near real time results may be required, especially for the detection of emergencies.
1.2 Motivation

There are several challenges that activity recognition algorithms are called to deal with and may be attributed to several causes. The most common problem encountered concerns the wide variations that may be observed even among the same type of actions (intra-class). Anthropometric, speed and action/stride length differences usually confuse recognition systems, leading to erroneous results. Temporal variations from one activity to another, which may be induced by different camera recording ratios or different performance styles, render activity recognition more difficult, as the time of change from one activity to the other is unknown. Cluttered and dynamic environments, occlusions during the recording, viewpoint alteration and changes in illumination render the recognition problem more difficult, making it essential to create a robust activity descriptor, which nonetheless, can be computed fast. Videos recorded from a moving camera make the motion analysis more difficult, as two different motion patterns (i.e. background and human motion) make up the entire movement, rendering a motion compensation algorithm a prerequisite for accurate human activity recognition.

A robust activity representation scheme is the stepping stone of a robust recognition system. It is common in the literature to adopt either (1) holistic approaches, where activity information is extracted from the entire person’s silhouette and is aggregated in a common descriptor or (2) local approaches that accumulate visual cues around spatio-temporal interest points and describe them in Bag-of-Words (BoW), Fisher, and VLAD encoding schemes [37], [38]. However, holistic (or global) approaches usually suffer from lack of localisation since they examine all human data at once, while they can be sensitive to changes in scale, occlusions of human parts or other objects, anthropometric differences and camera motion, due to their processing of the entire human data at once. Local methods aim to overcome limitations of global methods by focusing on local patches and volumes to ensure robustness to changes in scale, appearance and camera motion. Currently, these methods achieve the highest accuracy in activity recognition when used in a BoVW framework. Nonetheless, histogram encoding in BoVW methods suffers from lack of spatial information, as the geometric relations of the feature points being examined are not taken into account.

An essential step for activity recognition schemes with local patch-based descriptors is the interest point sampling procedure. Early interest point detection techniques were based on the extension of corner detectors over time, as in [20], where Harris3D was proposed, in [12] where Gabor filters were extended and the determinant of the Hessian matrix used in [39], while spatio-temporal salient points [40] are also encountered in the literature. The disadvantage of these techniques is that they extract a small number of interest points, which has now been shown to be too sparse and insufficient for discriminatively...
describing actions. Recent results in image classification \[41, 42\] have demonstrated that dense sampling increases recognition rates, motivating the activity recognition community to use it for action recognition, as in the current SoA (State-of-the-Art) \[43–45\].

One of the challenges faced for the recognition of activities is their varying temporal length, often caused by anthropometric variations. The same activity may be performed differently by various people, because of their particular kinematics, body types, habits, age and even scene context. Furthermore, as the number of subjects increases, so does the anthropometric variance, leading to changes not only in appearance characteristics of the activity taking place, but also in the time interval that is needed to perform the action. The current SoA does not address the issues related to time scale variance \[21\], resulting in errors arising from a low similarity score among similar activities recorded at a different frame per second ratio.

This thesis addresses many of these challenges, such as camera motion, spatio-temporal localisation, interest point sampling, time variance and activity representation, by incorporating new methods (Section 1.3), while building upon the SoA algorithms.

### 1.3 Contributions

#### 1.3.1 Computationally Efficient Dense Sampling

As mentioned in Section 1.2, the use of densely sampled interest points has led to excellent recognition rates, but at a high computational cost, which makes adequate sampling over time crucial, so that the sampled points are neither too sparse, nor too dense. In this work we introduce a representation schema with dense sampling that has a low computational cost, while also succeeding in increasing its recognition accuracy by examining specific regions of each video frame, shown to be informative about the activity taking place. Considering that motion information relevant to human activities is mainly concentrated in areas where the motion undergoes change, we extract these regions, referred to as Motion Boundary Activity Areas (MBAA), and densely sample interest points in them. In this way we simultaneously avoid errors introduced by false alarms from less relevant regions, we achieve computationally efficient dense interest point sampling, and we increase the method’s accuracy. This technique is presented in Section 4.2.1 and is mainly adopted to represent Activities of Daily Living (ADLs).

A similar technique is also presented in Section 3.2.2, where dense sampling is proposed to take place inside areas that compensated motion undergoes a change, referred to as
compensated activity areas (cAA). This reduces significantly the computational cost, as it is usual for dense sampling to extract interest points from videos with moving camera.

1.3.2 A Hybrid Global-Local Descriptor

In this work we introduce spatial localisation information for hybrid local-global action representation by including the trajectory coordinates in the action descriptor. Our experimental results show that, indeed, this technique successfully retains both global and local information and achieves high recognition rates when combined with appropriate encoding schemes.

We contribute a novel, hybrid descriptor that includes both global and local information in the action representation: local patches are extracted around densely sampled interest points in Motion Boundary Activity Areas (MBAAs), and a variation of the concatenated Histogram of Oriented Gradient (HOG) and Histogram of Oriented Optical Flow (HOF) activity descriptor \cite{12} is produced, while global information is introduced to these local patch-based HOGHOF descriptors by adding the trajectory points’ Cartesian coordinates to the action representation.

1.3.3 A Time Invariant Trajectory

To address the problem of varying temporal extent for each activity, we introduce temporally optimal trajectories, referred to from now on as optimal trajectories, which are constructed by statistically processing the histograms of optical flow around trajectory points. Their optimality lies in the fact that their temporal duration is determined by actual changes in motion. Temporal change detection takes place by applying Sequential Statistical Change Detection (SSCD), and specifically the Cumulative Sum (CUSUM) method, where the time of change of an unknown data distribution to a new, also unknown data distribution, is detected in nearly real-time.

1.3.4 Fisher encoding for Improved Recognition

The SoA in activity recognition usually uses a BoVW scheme in order to describe video sequences and recognize actions. Most methods do not delve into this part of the activity recognition framework in depth: usually, patch-based algorithms \cite{21}, \cite{12} use standard K-means clustering combined with a Chi-Square kernel for training a Support Vector Machine (SVM) classifier. The clustering is usually followed by hard binning for encoding and creating the BoVW histograms. However, K-means does not provide the best
possible clustering, as it requires a large number of cluster centers and requires a long computational time for reaching an optimal solution. At the same time, hard binning is likely to miss details and meaningful information that Fisher encoding can take into account, as soon as Fisher encoding has a probabilistic nature, allowing to bin a feature vector with a weight based on the whole set of cluster centers instead of matching a feature vector to the closest cluster center, as hard binning does.

Inspired by recent results in image classification [37], [46] we differentiate our approach from the SoA by deploying Fisher encoding for activity recognition. From our results we see that GMM clustering not only converges to an optimal vocabulary size faster than K-Means, but it also provides a more discriminative solution than K-Means do. Furthermore, Fisher provides a better encoding than BoVW, as it can combine diverse activity descriptors without losing its discriminative power. For that reason, we highly recommend Fisher encoding instead of BoVW.

1.3.5 Fast activity localisation

In Chapter 5, we present a spatio-temporal localisation algorithm that automatically detects regions where an activity occurs. Not only do these regions determine the temporal extent of the activity in the video sample, but they also simultaneously provide its spatial localisation. We enhance the simple sliding window used in similar works [47], [48] with statistical sequential boundary detection (SSBD) in order to decrease the computational cost, and applied it in real case scenarios.

1.3.6 ADL Recognition Dataset

Activity recognition methods use several benchmark datasets, presented and discussed in our previous work [17]. Early methods [14], [6] had practical limitations and were performed in constrained environments. More recent works focus on more demanding videos from Hollywood movies [12], [49] or sports events [15], [50].

ADL recognition involves real-life scenarios, which will arise in data from applications like smart homes or systems that support independent living by monitoring ADLs. Publicly available action datasets of ADLs are the University of Rochester Activities of Daily Living dataset (URADL) [11] and KIT,[16]. However, as detailed in Sec. 4.4.3.2, the URADL videos take place in a constrained environment with limited anthropometric variance, while in KIT ADLs of particular interest take place for extended periods of time.
Additionally, we recorded our own, realistic ADL dataset for the Dem@Care project, referred to from now on as the Dem@Care dataset. Recordings took place in two different rooms of the Greek Association for Alzheimer’s Disease and Related Disorders (GAADRD) in order to increase scene variability. Elderly people with and without dementia were asked to perform a series of ADLs in a home-like environment and were free to move among various parts of the room, instead of remaining at approximately the same location as in the URADL videos. 32 healthy individuals and 25 people with dementia, all aged over 65 and of both genders, were recorded, introducing large anthropometric variability and resulting in realistic recordings of ADLs. More details on the Dem@Care datasets can be found in [17].

1.4 Structure of thesis

In the second Chapter 2 we will elaborate on related work that is focused on activity recognition. In Chapter 3 we present an activity recognition technique which uses compensated Activity Areas and evaluate several encoding techniques in unconstrained videos. Chapter 4 focuses on ADL recognition and proposes a novel algorithm based on Sequential Statistical analysis for creating optimal trajectories. Chapter 5 presents a fast activity localisation algorithm based on Sequential Statistical Boundary Detection, while the thesis concludes with some discussions and future work in Chapter 6.
Chapter 2

Literature Review

Contents

2.1 Activity representation ..................................... 10
  2.1.1 Global activity representation .......................... 11
  2.1.2 Local activity representation .......................... 14

2.2 Activity detection ........................................... 17

2.3 Activity evaluation benchmark datasets ......................... 19
  2.3.1 KTH action dataset .................................. 20
  2.3.2 UCF action dataset .................................. 22
  2.3.3 Hollywood1 action dataset ........................... 22
  2.3.4 URADL action dataset ................................. 23
  2.3.5 KIT action dataset .................................. 24
  2.3.6 Dem@Care1 action dataset ............................ 25
  2.3.7 CHUN action dataset ................................ 26

2.4 Conclusion .................................................. 29

In this Chapter, we elaborate on the main aspects of activity recognition and present benchmark data-sets. In Section 2.1, we analyze the ways that an activity can be represented, differentiating among global (Section 2.1.1) and local-based (Section 2.1.2) approaches. We continue with Section 2.2 that elaborates on work that focuses on activity detection and finalize with benchmark and our own recorded action data-set on Section 2.3.

2.1 Activity representation

In this section, a thorough analysis of the ways that an activity can be described is provided. We separate the methods into global and local-based representations, as it is a
common practice to discriminate activities based on the level at which the activity analysis takes place. Furthermore, advantages and disadvantages are presented in conclusion of each subsection, while gaps in the literature are highlighted and solutions proposed.

### 2.1.1 Global activity representation

Global approaches are mainly techniques that gather information from the whole actor, aggregated in a single feature. These techniques can usually be categorized into **model-based** approaches, that need to fit an actor to a predefined human model, and **holistic** ones, that handle the whole body as a single moving entity.

**Model-based approaches:**

Body models are commonly based on a parametric representation of the human body recovered from video frames (or images), and entail body-part detection and tracking for accurate activity representation. An early work on model-based activity representation [3] (see Fig. 2.1), inspired from psychophysical work on visual interpretation of biological motion [51], showed that humans are able to recognize actions solely from the motion of a few moving light displays (MLD) attached to the human body. They then developed a model that focuses on localizing specific body points in video frames and tracking them.

Pose estimation [4], [52] is another way to describe actions. However strong prior assumptions have to be made. Mere occlusion and pose variations can render almost impossible for these algorithms to work in unconstraint environments or without using strong prior information. Characteristic samples of this class are given on Fig. 2.1

Recently, this domain has regained attention, as economical depth cameras (e.g. Kinect, Asus Xtion Pro) have been released to the market providing fast and accurate pose extraction. Some very promising works have already been developed such as [53].

**Holistic approaches:**

Holistic approaches, on the other hand, do not require the detection of body parts. Instead, whole human body is used in order to obtain appropriate structure and dynamics to describe the activity that is performed.

One of the earliest uses of silhouettes, representing the entire body as an entity, is introduced in [5], where motion silhouettes are calculated from a single view camera at each instant. These structures are then accumulated in a motion energy image (MEI), which indicates where motion occurs, and a motion history image (MHI) which retains the temporal information of the action. In order to build a recognition system these methods compare the templates using Hu shape descriptors. Characteristic samples of this approach are depicted in Fig. 2.2.
Figure 2.1: Examples of model based approaches centered on human body parts detection and tracking. Top row basketball pose is taken from [3], while medium and bottom row poses are taken from [4].

Figure 2.2: Motion energy (MEI) and Motion History Images (MHI) for a dance piroette taken from [5].
Other global methods extract motion history volumes (MHV) [7] or space-time shapes [6], which accumulate body silhouettes in order to describe activities. In both of these works, the Poisson solution is used in order to obtain additional features, such as saliency, orientation, action dynamics and shape structure, while silhouettes are obtained by background subtraction. Characteristic samples of these techniques are depicted in Fig. 2.3.

Generally, silhouettes indeed provide meaningful motion information, however they cannot deal with self-occlusions, camera motion and background clutter.

Another type of holistic techniques are those that rely on optical flow information. The most renowned amongst them was proposed in [8], where optical flow is used to track soccer players and a descriptor is built in order to represent some simple actions from a distance. As we can see from Fig. 2.4, the player is tracked throughout time and optical flow is calculated in a rectangular area around him. Optical flow is then divided in 4 distinct channels, one for each possible orientation, in order to ensure that vectors in opposite directions do not cancel each other out. The result is then blurred in order to eliminate possible faulty vectors that are attributed to noisy displacement vectors.

Optical flow is also used with holistic templates in [15] in order to introduce motion information. Regularity flow is taken into account in a spatio-temporal manner incorporating motion information to recognize actions.

Recently, another holistic feature, based on spatio-temporal 3D Gaussian steerable filters, was introduced in [54]. The recognition results were highly promising and further works, such as in [55], adopted and applied this technology in an action bank framework. This scheme can be easily extended for action detection, however it has been shown that it entails a high computational cost.
2.1.2 Local activity representation

Local activity representation is the most widespread way to describe an activity today, as it leads to State-of-the-Art (SoA) accuracy rates. Local patches are usually sampled either densely or by using a spatio-temporal detector. Specialized descriptors are built around the sampled interest points and a Bag-of-Visual-Words framework is used in order to aggregate them into a fixed size feature. The spatio-temporal descriptors are usually extensions of image-based 2D appearance histograms. They describe the region around the interest points, expanded into the temporal dimension (i.e. 3D-cuboids) in order to describe activities. The main advantages of local representations are: they are relatively independent to scale and shift invariant, they can deal with partial occlusions (i.e. human/object, object/object) and they do not need a preprocessing step (e.g. background subtraction, motion segmentation) to avoid possible failures. However, they suffer from their orderless representation, as BoVW methods do not retain spatio-temporal correlations among the features.

Spatio-temporal interest point detectors:
Interest point sampling is the initial step that a local-based technique requires in order to describe an activity. Spatio-temporal detectors minimize specific saliency functions in order to detect interest points which are induced by sudden changes in appearance and/or motion.

One of the earliest spatio-temporal detectors was proposed in [9] and then in [56], where a Harris corner detector [57] is extended to temporal domain. Space-Time interest points
are chosen as the points whose local neighbour, which is automatically selected, has a significant variation in both the spatial and temporal domain. A characteristic example of how the descriptor is constructed is depicted in Fig. 2.5.

Spatio-temporal action cuboids [39] are another concept behind interest point detection that can be found in the literature. They detect local maxima from a combined detector framework that uses Gaussian operators and Gabor filters in order to increase the sparse number of features that [9] provides. An improved work of [39] is presented in [58] where Gabor filters are combined with a differencing mask and different temporal scales are taken into account in the feature selection in order to tackle [39] limitations.

Another work that detects spatio-temporal interest points in videos was proposed in [40]. The authors extended a salient region detector by applying an entropy metric within a cylindrical region around each candidate point. The salient points that are selected are thresholded points that maximise the energy locally.

The Hessian detector was also extended to the temporal domain in [59] for interest point detection. Integral video structure and the determinant of a 3D Hessian matrix are used in order to flag the salient feature locations.

Despite the significant research effort devoted to the development of an accurate spatio-temporal interest point detector, it has not resulted in significantly increased recognition accuracy rates. The main disadvantage of these techniques is that they are sparse and interest points extracted shown to be insufficient for describing actions discriminatively.
Thus, related work [60], inspired by the recent methods of image classification [41], [42] has turned its attention to dense sampling approaches.

**Spatio-temporal descriptors:**

Spatio-temporal descriptors are computed around the resulting spatio-temporal interest points. Similarly to spatio-temporal interest point detectors, these structures extend image-based descriptors to the temporal domain in order to represent activities. As a consequence, we can encounter the extensions of SIFT, SURF, HOG, in the 3D domain [61], [59] and [20]. The temporal concatenation of image patch descriptors retains how a specific point and the region around it (i.e. containing its gradients) changes throughout time in order to adequately represent activities. However, more motion information can also be included in the descriptor, if specific motion attributes are taken into consideration (i.e. Optical Flow). This has been introduced in [12] and [21] where optical flow and its gradients are imported to a patch-based descriptor, leading to HOF [12] and MBH [21] structures respectively. An evaluation of these descriptors and their detectors is provided in [60].

Apart from describing regions around spatio-temporal interest points, the tracking of these points can also result in robust features for activity representation. For instance, in [11] a KLT tracker is used to form trajectories, while [10] propose matching SIFT descriptors using a Markov chain model in order to create correspondences among points. In both cases, trajectories are stored in a log-polar histogram of tracked velocities. In [62] the authors introduce trajectons. Densely sampled interest points are tracked using a simplistic tracking technique in [21], while an improvement of the same with motion compensation is given in [43]. Trajectories are also used in more recent works such as [44, 45]. Characteristic examples of their results are depicted in Fig. 2.6.

Currently, methods that use 3D local volumes with spatio-temporal information achieve the highest accuracy in activity recognition when used in a BoVW framework (K-Means clustering combined with a Chi-Square distance). However the main drawback that BoVW have is the lack of geometric relations between the features. Earlier work [12], inspired by spatial pyramid matching [63] which met with wide success in image classification, introduced weak geometric relations among descriptors in the BoVW framework, as depicted in Fig. 2.7. Further progress in the topic was made in [22] where an advanced hierarchical combination of features was proposed along with a data mining technique for improving recognition. Context information is also introduced in [10] where cuboid trajectory neighbourhoods are represented with a SIFT descriptor and relations among them are captured by a stationary Markov distribution vector at different levels. Recent work with several spatial pyramids was also introduced in [64], but did not achieve satisfactory improvements when compared with previous methods.
2.2 Activity detection

Activity recognition assumes that the video segments being analyzed include only one activity, while their activity boundaries (i.e. start and end frame of and action) are accurately defined by ground-truth annotation. Activity detection is an open research topic in computer vision that localizes activities in videos in space and time. Most of the previous works solve this problem by deploying a spatio-temporal window combined with machine learning algorithms.

Early techniques, such as those proposed in [65] and extended in [66], were mainly focused on detecting abnormalities in activity patterns extracted from CCTV videos (i.e. surveillance videos), after deploying an unsupervised segmentation algorithm. Behaviour patterns were discovered through unsupervised model selection and feature selection on
the eigenvectors of a normalised affinity matrix, leading to temporal activity localisation. However, these methods do not perform activity classification after their localisation.

Early works \cite{48, 67} focused on detecting specific action-poses in movies (e.g. drinking, smoking) by using a spatio-temporal video block classifier or by training specialized space-time cubes cast by localisation hypotheses. In \cite{68}, temporal sliding windows are used to detect human activities in videos, where the extraction of spatio-temporal interest points is followed by a Bag-of-Visual-Words (BoVW) framework. However, this approach did not improve detection rates compared to earlier works. In \cite{69}, activity detection takes place by accumulating space-time volumes from each video frame. The selection of each cuboid is then performed by a weighted spatio-temporal graph, leading to a higher computational cost than when using a simple spatio-temporal sliding cuboid, as the computation of such a matrix requires the square of the time needed to detect spatio-temporal cuboids.

Spatial localisation is taken under consideration in more recent works \cite{47}, where a human detector and tracker using a HOG descriptor is built to follow human subjects throughout the video and improve detection accuracy. However, a classifier needs to be computed beforehand to recognize upper bodies, where many false positives may arise. Furthermore, such a procedure may increase the computational cost, which is undesirable in real world scenarios. More focus on spatial localisation is found in \cite{70}, where the classifier for person detection was treated as a latent variable and in \cite{71} where structured output learning is applied. More recently in \cite{72}, hierarchical space-time segments are introduced and used to represent human bodies and detect/recognize activities afterwards.

For temporal localisation of activities, a fast technique was proposed in \cite{73}, based on Naive-Bayes Mutual Information Maximization (NBMIM) for multi-class action categorization. Inspired by this technique, we also base the detection of motion boundaries (the temporal segmentation of actions) on the instantaneous log-likelihood ratio. However, in our case the computation is data driven and does not require prior knowledge/description of the activities to be detected.

Recently, an alternative sliding window technique \cite{13}, where weights denote the start, middle and end keyframes of the windowed frames under examination, has led to State-of-the-Art (SoA) spatio-temporal activity detection and recognition accuracy. In that work, the authors introduced the idea of decomposing actions into atomic action units (actoms) and developed the Actom Sequence Model (ASM) to detect activities. A more recent method \cite{74} shows the improvements that sophisticated high level representation schemes, such as Fisher encoding, can add to this. Similarly to this approach, we adapt high level representation schemes to improve detection accuracies.
Other techniques that have performed action detection were introduced in [54], [55]. Both of them rely on a global feature template to detect and localize actions concurrently, while a different recognition approach is proposed.

It is worthwhile to note that, although the techniques presented here achieve very accurate detection rates, their computational cost is too high for real-life scenarios. In this work, spatiotemporal activity detection and classification takes place at a much lower computational cost, allowing for deployment in real world applications.

### 2.3 Activity evaluation benchmark datasets

Activity recognition from visual data has been evolved into one of the most active topics in computer vision within the last decade. Many action datasets collected from different sources (e.g. television, smart homes, lab, surveillance or individual recordings) have been presented in the literature, focusing on a large range of human activity aspects.

Early approaches, such as KTH and Weizmann [14], [6], focused on simplistic periodic activities that present several human participants performing them. These datasets contain uniform background and a static camera and were mainly built for evaluating activity recognition systems. IXMAS and HumanEva [75], [76] expanded the previous works by introducing a small set of simplistic activities with multiple cameras in a simplified room environment. Despite their limited applicability in real life scenarios, the aforementioned datasets are considered to be some of the most important indicators regarding the robustness of an activity recognition system and are still used in state-of-the-art methods evaluation studies.

More challenging datasets, usually referred to as ”actions in the wild”, appeared recently in the literature. These collections are considerably more challenging than the former ones and have been acquired by leveraging social media(e.g. facebook, twitter), video sharing websites(e.g. YouTube, vimeo) and television broadcasts(e.g. Olympic games, movies). Some characteristic examples belonging to this category are movie collections from INRIA: HOHA1 [12] and HOHA2 [49], sports video samples provided by Stanford university and University of Central Florida (UCF): Olympic Sports [50] and YouTube [77].
action datasets. In these scenarios introducing moving camera, great intra-class variation, background clutter and complete to partial subject occlusion render activity recognition extremely challenging leading to computational and memory demanding solutions.

Recent works turned their attention to large scale activity classification, introducing UCF101 action dataset [78]. This dataset comprises of 101 action classes in an unconstrained environment with several activities grouped into 25 classes and divided into five types: 1) Human-Object Interaction 2) Body-Motion Only 3) Human-Human Interaction 4) Playing Musical Instruments 5) Sports. The great challenge here is to learn a system that would be able to generalize in such a diverse and variant scenarios without losing its discriminative power.

Another category of activity data-sets that recently emerged are those that mainly focus on activities of daily living (ADL). These activities are performed by single individuals (e.g. patients with dementia or arthritis) throughout the day and monitored to create behavioural patterns. They are performed in a controlled indoor environment (i.e. dining room, kitchen), recorded with a fixed static camera and are usually much longer than unconstrained datasets as those mentioned in the previous paragraph. Amongst the most popular ones is the URADL in [11], however its usage is limited for evaluation purposes, as the samples are too short and there are not enough participants involved. Karlsruhe Institute of Technology followed with the KIT ADL dataset [16] and Munich university with the TUM action dataset [79]. The former one contained a large number of viewpoints and human participants while the latter mainly focuses on activities that take place in a kitchen environment. CMU introduced another ADL dataset in [80], which records a spate of human subjects preparing 5 different recipes recorded by several cameras. The most serious disadvantage of these recordings is that they only depict one action (i.e. cooking), limiting by this way its applicability to a very specific occasion. In the following subsections, we provide a more analytic description of the data-sets that have been used in our experiments.

2.3.1 KTH action dataset

One of the most popular datasets for human activity recognition is the KTH dataset [14]. The KTH videos consist of 2391 sequences, recorded in four different environments: outdoors $s_1$, outdoors with scale variation $s_2$, outdoors with different clothing $s_3$, and indoors $s_4$. Six different actions are performed by actors: box, handclap, handwave, jog, run, walk, while 25 subjects carried out these actions, for increased anthropometric variance. The dataset was split into 16 training and 9 testing videos, as suggested on [14],
# Chapter 2. Literature Review

## Dataset

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Action Classes</th>
<th>Videos</th>
<th>Evaluation metric</th>
<th>Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>KTH</td>
<td>Classes=6, Actions= Box/ Clap/ Wave/ Jog/ Run/ Walk</td>
<td>600 videos (160x120)</td>
<td>192 train/192 validation/216 test, Multi-class average accuracy</td>
<td>Indoor/ Outdoor videos, 25 participants/ 4 scenes</td>
</tr>
<tr>
<td>UCFsport</td>
<td>Classes=10, Actions= Dive/ Golf/ Kick/ Lift/ Ride/ Run/ Skate/ Bench/ High bar/ Walk</td>
<td>150 videos (720x480)</td>
<td>Leave-one-subject-out, (300 train/150 test), Multi-class average accuracy</td>
<td>Broadcast television videos</td>
</tr>
<tr>
<td>Hollywood (HOHA)</td>
<td>Classes=8, Actions= Answer phone/ Get-out-car/ Handshake/ Hug/ Kiss/ Sit down/ Sit up/ Stand up</td>
<td>430 videos (~ 400x~300)</td>
<td>Mean average-precision, 219 train/211 test</td>
<td>Short sequences from 32 movies</td>
</tr>
<tr>
<td>URADL</td>
<td>Classes=10, Actions= Answer Phone/ Chop Banana/ Eat Snack/ Dial Phone/ Drink Water/ Eat Banana/ Look-up in Phonebook/ Peel Banana/ Use Silverware/ Write on Whiteboard</td>
<td>150 videos (1280x720)</td>
<td>Leave-one-subject-out, Multi-class average accuracy</td>
<td>10 participants/ Indoor Kitchen activities</td>
</tr>
<tr>
<td>KIT</td>
<td>Classes=15, Actions= {Cut vegetables/ Dry dishes/ Fry vegetables/ Peel vegetables/ Stir soup/ Wash dishes/ Wipe countertop} + {Cut vegetables/ Empty dishwasher/ Peel vegetables/ Eat pizza/ Set table/ Eat soup/ Sweep floor/ Wipe table.}</td>
<td>(640x480)</td>
<td>10 subjects test/7 subjects train , Multi-class average accuracy</td>
<td>17 participants/ 2 different viewpoints</td>
</tr>
<tr>
<td>DemCare</td>
<td>Classes=11, Actions= Clean up table/ Drink beverage/ End phonecall/ Enter room/ Eat snack/ Handshake/ Prepare snack/ Read paper on the couch/ Serve beverage/ Start phonecall/ Talk to visitor</td>
<td>32 Videos (640x480)</td>
<td>Leave-one-subject-out, Multi-class average accuracy</td>
<td>32 participants/ 1 scene</td>
</tr>
<tr>
<td>CHUN</td>
<td>Classes=9, Actions= Answer phone/ Dial phone/ Look on map/ Pay bill/ Prepare drugs/ Prepare tea/ Read paper/ Water plant/ Watch TV</td>
<td>64 Videos (640x480)</td>
<td>Leave-one-subject-out, Multi-class average accuracy</td>
<td>64 participants/ 1 scene</td>
</tr>
</tbody>
</table>

| Table 2.1: List of human activity video datasets and their properties. |
based on the subjects that perform the actions, in order to evaluate our algorithm. KTH activity samples are depicted on Fig. 2.9.

2.3.2 UCF action dataset

The UCF - sports videos [15], consist of 150 videos depicting 11 kinds of actions. The main challenges here are the moving camera and the variability observed among the actions. In order to increase the number of training videos for a more complete evaluation, we vertically flip the original videos and use them for training, but we only use the original videos for testing as in [15]. The UCF sport actions dataset contains ten different types of human actions: swinging on the pommel horse, on the floor, at the high bar and golf swinging, diving, kicking, weight-lifting, horse-riding, running, skateboarding and walking. Fig. 2.10 depicts frame samples from this dataset.

2.3.3 Hollywood1 action dataset

The Hollywood1 data, proposed by [12], contains Hollywood movies with activities that are quite challenging to characterize: there is high variability in the human actors performing the actions, shot changes throughout the videos, viewpoint variations and
Chapter 2. Literature Review

2.3.4 URADL action dataset

The University of Rochester Activities of Daily Living (URADL) dataset was examined in detail, due to its inclusion of ADLs which are relevant in many practical applications. In the URADL videos, 5 different actors performed 10 different activities, 3 times each, in a kitchen environment, resulting in 150 videos. High resolution video frames were captured from a static RGB camera placed across the actors. The short duration of these videos rendered the dataset more appropriate for evaluating recognition of ADLs approaches, rather than for the determining algorithmic usefulness in real life scenarios.

A disadvantage of the URADL dataset is that it lacks significant anthropometric variations, as most actors have a similar appearance and perform the activities in the same manner. Additionally, the URADL videos are characterized by considerable environmental constraints: all actions take place in the same location, behind a kitchen counter, while the subjects do not move significantly. The actors are clearly visible and there are very few occlusions, as seen in sample frames in Fig. 2.12. Nevertheless, URADL is widely used in the literature [11], its small size making it useful for the fast evaluation of algorithms.
In order to evaluate our algorithm, we use a leave-one-subject-out testing approach. For these videos the names of the activities are represented by: AP : Answer Phone, CB : Chop Banana, ES : Eat Snack, DP : Dial Phone, DW : Drink Water, EB : Eat Banana, LiP : Look up in Phonebook, PB : Peel Banana, US : Use Silverware, WoW : Write on Whiteboard.

2.3.5 KIT action dataset

A second, more challenging ADL dataset is the KIT Robo-Kitchen activity dataset, publicly provided by the Karlsruhe Institute of Technology [16]. In these videos, cameras were installed in two different places inside a smart home: one that records the participant in a kitchen environment and another placed to view the room, where 17 human actors were called to perform 14 different activities. The first experimental setup for KIT, named “Counter Top”, concerns kitchen activities around a kitchen counter and sink that are recorded by three cameras, while the second set of recordings, named “Room Setup”, features a different set of kitchen related activities, recorded by two cameras. We examine the activities of each setup separately (since they contain different activities), but use the data from all camera views for each set of experiments. This allows us to use a larger number of training and testing videos, and also demonstrate the viewpoint invariance of our method. The number of actors, their different appearance and large number of activities performed in various manners led to considerable anthropometric variance. The KIT kitchen environments are more realistic than those in URADL, with furniture and kitchen equipment placed in different parts of the room, necessitating more movement around the room to perform each activity, which results in more realistic ADLs than in URADL.
Kit-Counter Top Setup:
In our experiments, we test our algorithms on the Counter Top setup, where kitchen activities are performed on a kitchen counter around the sink, as shown on the top row of Fig. 2.13. Three different cameras record the activities, giving a multi-view observation for each person, which can also be used for stereo vision analysis, but is not examined here. In our experiments, we accumulate all videos from the three cameras and use them in a leave-one-subject-out train/test split. Activities for this setup, presented in the tables below, are represented as: Cut: cut vegetables, Dry: dry the washed dishes, Fry: fry vegetables, Peel: peel vegetables, Stir: stir a cooking soup, Wash: wash dishes, Wipe: wipe the countertop.

Kit-Room Setup:
In Room Setup, two different cameras record the following activities: Cl: clear the table, Cf: drink coffee and read a newspaper, Ct: cut vegetables, ED: empty the dishwasher, Pl: peel vegetables, Pz: eat pizza, ST: set the table, Sp: eat soup, Sep: sweep the floor, Wp: wipe table. The bottom row of Fig. 2.13 shows some sample frames from this setup. As mentioned above, videos from both cameras are accumulated and used in a leave-one-subject-out train/test split.

2.3.6 Dem@Care1 action dataset
Taking into account the aforementioned limitations of public ADL datasets, we proceeded with the recording of ADLs in the premises of the Greek Association of Alzheimer’s Disease and Related Disorders (GAADRD) in Thessaloniki [17]. The subjects were age over 65, with conditions ranging from mild cognitive impairment (MCI) to dementia and
full-blown Alzheimer’s, while an almost equal number of healthy individuals in the same age group was also recorded performing ADLs. The participants were of both genders and the activities they performed required moving around the room, similarly to real life. These factors, as well as the size of the population being recorded (32 participants) introduced great anthropometric variations in our ADL videos and made them more realistic than current benchmark data.

In this first dataset 32 patients with Mild Cognitive Impairment (MCI) were called to perform a set of predefined activities. Activities were designed so that information concerning the patients’ capabilities could be extracted. Eating and drinking scenarios were recorded in a kitchen environment. The eating scenario included the preparation of a meal and its consumption, while in the drinking scenario, a beverage was served in a glass and later on consumed. Both scenarios were followed by a cleaning up activity, so that it can be observed if the patient is capable to leave the scene in a proper condition. The socializing capability of the patient was checked by initializing two different scenarios. In the first one the human subject was called to use a phone to contact another person. The scenario included the start phone-call action, which detected when the patient picks up the telephone handset, and the end phone-call action, which detected when the patient hang ups the phone and terminates the conversation. A visiting activity was the second scenario that examined their socializing capability. In this case, a visitor enters the room where the patient stays, has a handshake or hug with him and starts a conversation. Finally, the patients’ capability of allocating recreational time within their day was checked by a ”reading a paper” activity. In this scenario a patient is called to sit in a sofa or chair, take a book and read it. The description of the activities and further details are aggregated in Table 2.2.

The ADLs performed are encoded in the tables below as CU: clean up table, DB: drink beverage, EP: end phonecall, ER: enter room, ES: eat snack, HS: handshake, PS: prepare snack, RP: read paper on the couch, SB: serve beverage, SP: start phonecall, TV: talk to visitor. DemCare frame samples are depicted on Fig. 2.14

2.3.7 CHUN action dataset

Another realistic ADL dataset was recorded at the Nice University Hospital (CHUN) for evaluating the applicability of our algorithm. The CHUN dataset consists of 15 hr and 10 min recordings of 64 PwD that perform ADLs in a Lab environment. The camera viewpoint monitors the whole room where the patient is and performs semi-directed ADL’s (i.e. specified on a paper).
<table>
<thead>
<tr>
<th>Abilities</th>
<th>Initials</th>
<th>ADLS</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eating capability (Kitchen)</td>
<td>PS</td>
<td>Prepare snack</td>
<td>The patient is called to take a plate and a snack from the table and prepare a meal.</td>
</tr>
<tr>
<td></td>
<td>ES</td>
<td>Eat snack</td>
<td>The patient picks the snack that is on the table and eats it.</td>
</tr>
<tr>
<td>Drink capability (Kitchen)</td>
<td>SB</td>
<td>Serve Beverage</td>
<td>The patient takes a bottle of water or orange juice and pours it inside a glass. He brings the glass in front of him.</td>
</tr>
<tr>
<td></td>
<td>DB</td>
<td>Drink Beverage</td>
<td>The patient drinks the liquid that his has served in his glass by bringing the glass to his mouth.</td>
</tr>
<tr>
<td>Cleanup capability (Kitchen)</td>
<td>CU</td>
<td>Clean Up table</td>
<td>The patient cleanup the table in front of him, by discarding the glass and the plate to a bin.</td>
</tr>
<tr>
<td>Phone capability (Social)</td>
<td>SP</td>
<td>Start phone-call</td>
<td>The patient picks up the phone and dials a number, indicating that he initializes a phone-call.</td>
</tr>
<tr>
<td></td>
<td>EP</td>
<td>End phone-call</td>
<td>The patient puts down the phone, indicating the termination of the phone-call.</td>
</tr>
<tr>
<td>Having visitor capability (Social)</td>
<td>ER</td>
<td>Enter room</td>
<td>An activity which indicates that a person opened the door and entered the room.</td>
</tr>
<tr>
<td></td>
<td>HS</td>
<td>Handshake</td>
<td>The patient greets the visitor by having a handshake with him.</td>
</tr>
<tr>
<td></td>
<td>TV</td>
<td>Talk to visitor</td>
<td>The patient talks to his visitor, by standing in front of him and making some gestures.</td>
</tr>
<tr>
<td>Recreation capability (Reading)</td>
<td>RP</td>
<td>Read paper</td>
<td>The patient sits in a sofa or chair and reads a book that is placed in a table next to him.</td>
</tr>
<tr>
<td></td>
<td>UC</td>
<td>Use closet</td>
<td>The patient opens up a closet, picks a book and closes its door. (exists only in DemCare2)</td>
</tr>
</tbody>
</table>

Table 2.2: The set of activities that are observed in DemCare1 and DemCare2 action datasets and their description
The ADLs observed include: AP: answering phone and DP: dialing phone, LoM: look on map, PB: pay bill, PD: prepare drugs, PT: prepare tea, RP: read paper, WP: water plant and WtV: watch TV. They included large anthropometric variations and activity performance styles, while severe occlusions introduced great difficulty in discriminating actions. Video samples from CHUN dataset are shown in Fig. 2.15.
2.4 Conclusion

In this chapter we have seen the most pioneering works in the activity detection and activity recognition literature. Early works were mainly focused on extracting holistic features from human body parts, while local based approaches were recently proposed producing State-of-the Art accuracy rates.

Numerous methods have been developed to tackle moving camera, variant viewpoint videos and background clutter videos, depending on environmental and camera constraints. However, the advocated algorithms are unacceptably slow.

Despite the large number of methods on spatio-temporal interest point detection, there is a conceptual gap between dense and sparse detection approaches. Motivated by this, we have developed two hybrid interest point detectors for constrained and unconstrained real life scenarios (i.e. videos of ADL and broadcast videos which are more unconstrained respectively). Furthermore, focus have been placed on developing a sequential statistical algorithm based on motion pattern analysis of human trajectories, to address the time varying nature of human activities in different contexts.

Activity detection and localisation, on the other hand, has little to show and a large number of issues remain unexplored. Most works focus on temporal sliding window analysis and demand a high computational cost to process an unsegmented video. We propose a near real time activity detection algorithm based on sequential statistical analysis of activity patterns and discuss its strengths and limitations.
Chapter 3

Moving camera action recognition and encoding schemes in Compensated Activity Areas (cAA) using a Hybrid Descriptor

Contents

| 3.1  | Introduction .................................................. | 31 |
| 3.2  | Methodology .................................................. | 32 |
| 3.2.1 | Motion compensation ......................................... | 33 |
| 3.2.2 | Compensated Activity Areas (cAA) .......................... | 34 |
| 3.2.3 | The hybrid activity descriptor ............................ | 36 |
| 3.3  | Encoding framework .......................................... | 39 |
| 3.3.1 | Bag-of-Visual-Words (BoVW) .................................. | 39 |
| 3.3.2 | VLAD vector .................................................. | 40 |
| 3.3.3 | Fisher vector ................................................ | 41 |
| 3.4  | Experiments .................................................... | 41 |
| 3.4.1 | Spatial localisation ......................................... | 41 |
| 3.4.2 | Activity recognition ......................................... | 43 |
| 3.4.3 | Computational time and memory consumption ............... | 53 |
| 3.5  | Discussion ..................................................... | 53 |

In this chapter, we present a novel approach for human activity recognition in videos with camera motion and/or large viewpoint variations, localising activity areas after motion compensation, and building a dense hybrid activity descriptor in them. The
issues limiting the SoA (State-of-The-Art) activity recognition methods, such as insufficient spatial information, heavy computational cost, data sparsity are addressed. The proposed methodology is compared with the SoA encoding methods. The experimental results demonstrate the superiority of the proposed method, showing that it can lead to improvements in accuracy over the SoA, without introducing a heavy computational burden.

3.1 Introduction

Numerous methods have been proposed for human activity recognition, with many State-of-the-Art (SoA) works highlighting the importance of deploying a meaningful but compact action descriptor. As highlighted in Sec. 2.1.2, early interest point detectors [9], [56], [39], [40] were powerful, but provided too sparse interest points. Recent works [60], [21], [45], [44], inspired from the success of dense sampling in image classification [41], [42] extract dense interest points, however their large number can lead to ambiguities and to high computational costs, especially for videos with a moving camera. This led us to build a novel motion segmentation binary mask, the compensated Activity Area (cAA), that separates motion-compensated pixels undergoing changes over time from static ones. A dense grid is superimposed over each cAA and interest points are densely sampled in it for a more efficient and informative representation than that extracted from interest points and dense sampling over the whole frame.

Furthermore, an encoding framework (i.e. Bag-of-Visual-Words(BoVW), VLAD, Fisher) follows usually local-based representations, in order to construct fixed size feature vectors and render video segment classification feasible. However, their implementation eliminates any spatial relations between interest points, which could provide meaningful information. Several techniques have tried to deal with this issue [12], [64] inspired from the pyramid paradigm [63] commonly used in image classification, where spatio-temporal pyramids were introduced for action cuboids. However, the spatial information in them proved to be insufficient for accurate action recognition and did not succeed in improving recognition rates. We propose to tackle this issue at the representation level by building a hybrid activity descriptor that incorporates local motion and appearance features with global trajectory spatial information. Our experiments show that this descriptor, when combined with Fisher encoding, can exploit spatial information and lead to better recognition rates. Theoretically, GMM clustering, which is used to compute a visual vocabulary for Fisher encoding scheme, computes mean and standard deviation for each cluster center. When an activity is located in a specific region in the video frame, then the standard deviation for location variables will be small and would be possible to
discriminate it from other activities that happen elsewhere in the image. On the other hand, when an activity does not occur in a specific region, location information will give a large standard deviation on the cluster center and consequently it will not be specific enough to discriminate the action. In this way, location information can be leveraged to provide a robust spatial information to the final high level activity descriptor.

Experiments were also carried out on benchmark action datasets by deploying three SoA encoding schemes to define our activity descriptor and reporting recognition rates. This was motivated by the great number of feature encoding techniques that have been recently proposed in the literature [44], [45], [81], and the absence of a fair comparison under a common set of experiments.

Summarizing, this Chapter presents a feature-based method that improves upon the State of the Art in human action recognition through the following contributions:

- Dense sampling only in motion compensated areas where pixels are active (i.e. they undergo a change over time). We succeed to reduce the computational cost and eliminate potential false alarms induced by camera motion and erroneous optical flow.

- Hybrid activity representation that combines appearance, motion and trajectory descriptors (Histograms of Oriented Gradients (HOG), Histograms of Optical Flow (HOF) and trajectory descriptor) around interest points at a local and global level and in multiple scales to retain granularity, scale invariance and maintain geometry and spatial localisation information.

- An evaluation of recent encoding techniques on benchmark activity data-sets (i.e. KTH, UCF sports, HOHA and YouTube), in order to highlight the most appropriate one for activity recognition. Advantages and disadvantages are discussed in the experimental section.

The chapter is organised as follows: Sec. 3.2 presents the approach adopted in this work, with the motion compensation of the optical flow in Sec. 3.2.1, the analysis of cAA masks in Sec. 3.2.2 and the hybrid descriptor in Sec. 3.2.3. Sec. 3.3 follows with the encoding schemes and experimental results are given in Sec. 3.4.

### 3.2 Methodology

In order to classify human activities, the proposed method estimates the dominant motion (Sec. 3.2.1), eliminates it from the optical flow (OF), and computes dense trajectories
over the so-called “compensated Activity Areas” (Sec. 3.2.2) for each frame. Appearance, motion and spatial information, namely HOG/HOF and trajectory descriptors, are extracted around each sampled interest point to describe human activities (Sec. 3.2.3). The resulting activity descriptor is encoded in a fixed size vector by using BoVW, VLAD or Fisher, and fed to a multi-class SVM classifier for activity recognition (Sec. 3.3). The whole process is depicted in Fig. 3.1.

3.2.1 Motion compensation

Dominant motion estimation and compensation is an essential step for analysing videos that are recorded by a moving camera. The central idea is that dominant motion caused by a moving camera can be modelled and removed from the OF estimates, resulting in the compensated OF, i.e. the foreground motion.

In this work, we use the 8-parameter quadratic model (equivalent to a homography) of [82] instead of the simpler 4-parameter model [83], from the combination of 2D translation, 2D rotation and scaling. We choose this technique because we believe that Block Matching Algorithm (BMA) has a good balance between accurate motion estimation and computational efficiency. A more complicated compensation algorithm or OF would require more computations but would not lead to significantly more accurate results. The bilinear model of the dense OF [18] global motion vector \((u, v)\) for a point
(x, y) can be expressed as:

\[ u = a_0 + a_1 x + a_2 y + a_3 xy \]
\[ v = a_4 + a_5 x + a_6 y + a_7 xy \]  

We divide the video frame into blocks and use a block matching algorithm (BMA) with least-squares to obtain the dominant motion, approximating the OF \((u, v)\). An iterative algorithm uses least squares for each block to minimize the sum-of-squared estimation error:

\[ E_{all} = \sum_{i=1}^{N} (u_i - \hat{u}_i)^2 + (v_i - \hat{v}_i)^2 \]  

where \((\hat{u}_i, \hat{v}_i)\) are the dominant motion vectors calculated from Eq. (3.1) using the estimated parameters and \((u_i, v_i)\) is the block match result of the \(i^{th}\) block. The iterative rejection algorithm then eliminates erroneous block matches larger than the average estimation error:

\[ E_{avg} = \frac{1}{N} \sum_{i=1}^{N} \sqrt{(u_i - \hat{u}_i)^2 + (v_i - \hat{v}_i)^2} \]  

This continues until the iterations leave the set of rejected blocks unaltered. The resulting approximation of the dominant motion vector \((\hat{u}, \hat{v})\) is removed from the initial estimate \((u, v)\), giving the compensated motion vector \(CU = (cu, cv)\), hereafter called as compensated optical flow. An example is depicted in Fig. 3.4.

### 3.2.2 Compensated Activity Areas (cAA)

The compensated OF (or the original unaltered OF when static camera is concerned) is used to build “compensated Activity Areas” (cAA), which indicate regions of true motion, so as to reduce the computational burden and false alarms, since data from static pixels will not be used. In order to extract the cAA, for every time instant \(k\) a set of frames is examined from \(k - W_0\) to \(k\). In our case, the size of the temporal window \(W_0\) that provided the most meaningful results was determined by the frame per second (fps) recording rate of the videos. In particular, we observed from cross validation on the KTH and UCFsports action datasets that the best size for \(W_0\) is usually equal to half of the fps (frame per second) of the video recording, so we use \(W_0 = (fps/4) \simeq 6\).

The data at frame \(k\) and pixel \(\tau\) is \(CU_k(\tau)\), i.e. the compensated flow caused either by compensated motion or by measurement noise. Each pixel’s compensated OF at frame \(n\), \(k - W_0 < n \leq k\), can be modeled by the following hypotheses:

\[ H_0 : CU_k^0(\tau) = z_k(\tau) \]
\[ H_1 : CU_k^1(\tau) = CU_k(\tau) + z_k(\tau); \]
where \( \text{CU}_k^H(\tau) \) is the compensated OF for \( i = 0, 1 \), \( \text{CU}_k(\tau) \) is the OF caused by actual motion and \( z_k(\tau) \) caused by measurement noise. Thus, \( H_0 \) corresponds to the case where the pixels’ compensated OF is caused by measurement noise, and \( H_1 \) where it is caused by actual motion. The algorithm can then determine if a pixel is active by finding if its compensated OF follows the distribution of the measurement noise.

We make the assumption that noise induced compensated OF follows a Gaussian distribution. This assumption can be considered valid, as OF estimates originate from sums of many i.i.d. random variables, which ultimately converge towards the same Gaussian distribution based on the Central Limit Theorem [84]. This does not apply to motion-induced compensated flow values, as they change depending on the video and the activities in it, unlike noise-induced compensated flow (which contains far fewer outliers).

The concept behind this idea is that human-induced optical flow vectors represent the true motion in the scene, but also contain noisy estimates over time. When motion changes quickly, true motion gradients will feature higher peaks and may change significantly over time. On the other hand, noise-induced optical flow vectors are caused by estimation errors and are not correlated to the motion and the changes in it. Therefore, they usually deviate around a small mean value under a standard deviation which is usually produced from small deviations in the video frame’s luminance, so they can be modelled by a Gaussian around this value.

**Kolmogorov-Smirnov test:**

To verify that the compensated OF follows a noise-induced Gaussian distribution, we employ the Kolmogorov-Smirnov (K-S) test [85], which checks if the empirical data distribution matches a reference distribution, in this case the Gaussian fit to our data. We apply the K-S test on compensated OF from 18 videos randomly sampled from the UCF Sports dataset [15]. Fig. 3.2 shows the magnitude difference between the compensated OF inside (blue) and outside (green) AA through 18 UCFsport video samples while Fig. 3.3 shows the compensated flow and their empirical cumulative distribution function (cdf) respectively. The separation between the two compensated OFs in Fig. 3.2 is obvious, while the Root Mean Square Deviations (RMSDs) of the empirical cdf fits show in Fig. 3.3 that only data outside the cAA (RMSD=0.1240) can be modelled by a Gaussian distribution, as opposed to that located inside the cAA (RMSD=0.4593). Thus, the K-S test confirms that static pixels follow an approximately Gaussian distribution. A classical measure of Gaussianity is the data kurtosis, which is equal to zero for Gaussian data and can be estimated by its excess form [86] for the compensated OF \( \text{CU}_k(\tau) \) by:

\[
\text{Kurt}(\text{CU}_k(\tau)) = \sum_{k=1}^{W_0} \frac{\text{CU}_k(\tau)^4}{W_0} - 3 \sum_{k=1}^{W_0} \left( \frac{\text{CU}_k(\tau)^2}{W_0} \right)^2 \tag{3.5}
\]
Then, the kurtosis of $CU_k(\tau)$ forms a “kurtosis mask”, which obtains high values in moving pixels, and low values in the static ones.

To determine a threshold that will separate these classes, we examined all video samples from UCFsports dataset [15] where the ground truth of the foreground (and consequently the background) is provided manually. A mean value and a standard deviation of compensated OF is extracted for background regions and are used to define the binary mask as:

$$cAA(\tau) = \begin{cases} 
0 & \text{if } Kurt(CU_k(\tau)) < mean[BG(\tau)] + 2 \times std[BG(\tau)] \\
1 & \text{else}
\end{cases}$$

where $BG(\tau)$ denotes the compensated OF values that belong on the background of the video samples.

The robustness of the kurtosis for extracting cAAs has been analysed in [87] as well, where it is shown to provide accurate activity areas, even for videos with slightly varying backgrounds (e.g. in backgrounds where wind may move the branches of the trees). In that work, the authors used luminance values for each pixel instead of motion (i.e. optical flow). However, that solution would require a larger window size $W_0$ and consequently more video frames to converge into a robust solution contrary to motion analysis which requires just a few frames. The cAAs superimposed on frames from some videos used in our experiments, depicted in the bottom row of Fig. 3.4, show that the regions of human motion are accurately localized. The robustness of cAA localisation power and comparisons to related work on two public available datasets is provided in Sec. 3.4.1.

### 3.2.3 The hybrid activity descriptor

Once the cAA is extracted, multi-scale dense sampling is performed on a grid with step size $W_{\text{step}} = \{8, 16, 24, 32\}$ pixels, and the candidate points $P_{c,t} = (x_t, y_t)$ are accumulated in a temporary set. Rectangular regions at double the size of the sample step $Rect_{\text{size}} = 2 \times W_{\text{step}}$ are examined around these points, to detect other candidates in the surrounding area. If not, they are referred to as sample points $P_{s,t} = (x_t, y_t)$ and tracked over time using KLT [88]. The sample points are then accumulated in a set of points $\text{tra}j_t = \{P_{(s,t)}, P_{(s,t+1)}, ..., P_{(s,t+W-1)}\}$ for a fixed temporal length $W = 15$, and form a cAA trajectory structure.

Motion and appearance features are extracted on a multi-scale grid around each matched point that varies based on the sampling step: $\{(16 \times 16), (24 \times 24), (32 \times 32), (48 \times 48)\}$. This increases the granularity of the description and incorporate structural object and
Chapter 3. Hybrid HOGHOF descriptor and cAA for action recognition

Figure 3.2: Concatenated compensated OF (magnitude) inside (blue line) and outside (green line) cAAs from 18 UCF sports videos. Compensated OF in cAAs is higher and more irregular than outside cAAs.

Figure 3.3: The plot is used for KS-test visualisation and shows the similarity between the empirical-cdf of the centered and scaled compensated OF inside (blue line) and outside (green line) cAA and the cdf of the standard normal distribution (red). The RMSD between the Gaussian model of the compensated OF and the compensated OF distribution inside cAA is higher than outside it, showing that the compensated OF follows a Gaussian distribution for static pixels (All cdfs have been shifted to x-axis so that the standard normal cdf could be centered to zero when cumulative probability equals to 0.5).
scene information. HOFs are used for motion and HOGs for appearance information. They are combined in a concatenated HOGHOF feature vector, in order to describe the scene as fully as possible. For the HOF and HOG descriptors, the orientations of the flow and spatial gradients (extracted with the Sobel operator) are distributed among 8 bins, with an extra zero-bin for the HOF. HOFs and HOGs are estimated in 4 cells around each interest point and each cell’s histogram values are normalized to ensure illumination and motion scale invariance for HOG and HOF respectively. The resulting HOFs and HOGs are also normalized within the blocks containing the 4 cells to ensure that low contrast is enhanced. The HOGs and HOFs of each trajectory are then aggregated separately (i.e. so that they can retain their attributes) throughout time for a fixed temporal length $W$, segmented into three equally sized sets over time ($n_t = 3$), averaged, normalized by the $L_2$ norm and finally concatenated in order to create the final HOGHOF descriptor. This activity descriptor (i.e. HOGHOF) is then concatenated with the aforementioned trajectory descriptor (i.e. $\text{tra}j_t$), in order to construct the hybrid global-local descriptor, which we refer to as HOGHOF+Traj.

We need to highlight that both descriptors are manipulated separately throughout the whole procedure, so that we could create a robust low-level representation that would not confuse two different representations. Furthermore, we differentiate from related work, as we adopt a mirroring binning for HOF descriptor so that we can classify similar activities with different direction in the same class. In Fig. 3.5 we can see the overall procedure adopted in this work in order to construct HOGHOF+Traj activity descriptor.
3.3 Encoding framework

For vocabulary construction we use $K$-Means and Gaussian Mixture Model (GMM) clustering, which are further analyzed in Appendix B. Let $CC = \{\bar{c}_1, \bar{c}_2, ..., \bar{c}_K\}$ be the visual vocabulary acquired from the aforementioned clustering techniques and $X = \{\bar{x}_1, \bar{x}_2, ..., \bar{x}_N\}$ the action descriptor extracted from a video sample. In this section, we elaborate on the appropriate encoding techniques that encode action descriptors $x_i$ in a fixed size feature vector $F(X)$, either by computing frequency occurrences (Sec. 3.3.1) or by accumulating the first and second differences among them (Sec. 3.3.2, Sec: 3.3.3). Appropriate Kernelization follows in each case, so that we can train a linear multi-class SVM model and classify unknown activities. A block diagram of the whole process is depicted in Fig. 3.6.

3.3.1 Bag-of-Visual-Words (BoVW)

The most common technique for activity recognition is BoVW, where a $K$ bin histogram encodes the frequency of each cluster center. Thus, when each $\bar{x}_i \in R^D, \ i = \{1, ..., N\}$ is assigned to its closest cluster center $\bar{c}_j \in R^D, \ j = \{1, ..., K\}$, the corresponding index in the BoVW vector $f_j$ is increased by 1. Dividing the frequency vector $f$ by $L_2$ transforms all vectors to the same scale, while Chi-Square Kernelization follows to train a Linear SVM classifier. For two BoVW histograms $f \in R^K$ and $g \in R^K$, the Chi-Square kernel
Figure 3.6: Training visual descriptors are clustered using K-Means or GMM algorithm and based on the extracted means (or cluster centers accordingly) appropriate encoding is performed (VLAD, Fisher, Chi-Square). Training encoded data are used to train a SVM classifier, while test encoded data evaluate its discriminative power.

is given by:

\[ K(f, g) = \sum_{j=1}^{K} \frac{2f_j \cdot g_j}{f_j + g_j} \]

### 3.3.2 VLAD vector

The Vector of Locally Aggregated Descriptors (VLAD) is used to encode activity descriptors into a fixed feature vector, proposed in [37].

In this encoding approach, we use K-Means clustering in order to obtain the visual vocabulary \( \tilde{c}_j \in \mathbb{R}^D, j = \{1, ..., K\} \), and we associate each local descriptor \( \tilde{x}_i \in \mathbb{R}^D, i = \{1, ..., N\} \) to its nearest visual word \( NN(\tilde{x}_i) \). For each codeword \( \tilde{c}_j \) the differences \( \tilde{x}_i - \tilde{c}_j \) of the vectors \( \tilde{x}_i \) assigned to \( \tilde{c}_j \) are accumulated:

\[
\tilde{f}_j = \sum_{\tilde{x}_i:NN(\tilde{x}_i)=j} (\tilde{x}_i - \tilde{c}_j) \tag{3.6}
\]

All these \( D \)-dimensional residuals are then concatenated into a fixed size feature vector \( F(X) = [\tilde{f}_1, \tilde{f}_2, ..., \tilde{f}_K] \) of dimension \( KD \), in order to form the final VLAD descriptor for each video sample. Component-wise \( L_2 \) normalization follows, where the vectors \( \tilde{f}_j \) are divided by their norm \( ||\tilde{f}_j||_2 \) and their square root, so that the transform \( sign(z)\sqrt{|z|} \) is applied to all scalar components of \( F(X) \).
3.3.3 Fisher vector

Fisher encoding [38] uses visual vocabulary extracted from GMM clustering in order to calculate the first and the second order differences among the visual feature vectors and their corresponding Gaussians and build a fixed size feature vector. Each action descriptor \( \bar{x}_i \in \mathbb{R}^D, i = \{1, ..., N\} \) is compared against all the mixtures of Gaussians \( \bar{c}_j \in \mathbb{R}^D, j = \{1, ..., K\} \) in the visual vocabulary, and their first and second difference is computed as:

\[
\begin{align*}
    f_{1j} &= \frac{1}{N} \sqrt{\frac{\pi_j}{\pi_j}} \sum_{i=1}^{N} q_{ij} \Sigma_j^{-1/2} (\bar{x}_i - \bar{\mu}_j), \\
    f_{2j} &= \frac{1}{N} \sqrt{\frac{2\pi_j}{\pi_j}} \sum_{i=1}^{N} q_{ij} [(\bar{x}_i - \bar{\mu}_j) \Sigma_j^{-1}(\bar{x}_i - \bar{\mu}_j) - 1],
\end{align*}
\] (3.7)

where \( q_{ij} \) is the Gaussian soft assignment of the descriptor \( x_i \) to the \( j \)-th Gaussian, while \( \mu, \Sigma \) and \( \pi_k \) are the mean, covariance and mixture weights of each Gaussian. The two distances are then concatenated to form the final Fisher vector: \( F_X = [f_{11}, f_{21}, ..., f_{1K}, f_{2K}] \), which characterizes each video sequence. Each vector is normalized using a Hellinger kernel, which gave very good results in [46], and for two Fisher vectors \( f \in \mathbb{R}^{2KD} \) and \( g \in \mathbb{R}^{2KD} \), is computed by:

\[
K(f, g) = \sum_{j=1}^{K} \text{sign}(f_j) \text{sign}(g_j) \sqrt{\|f_j\| \cdot \|g_j\|}.
\]

where \( K \) denotes the number of Gaussians, \( D \) the dimensionality of the activity descriptor and \( 2KD \) the final Fisher vector size.

3.4 Experiments

In this section, we provide two kinds of evaluation in order to evaluate our representation and recognition algorithm. In sec. 3.4.1 we evaluate the spatial activity localisation power of cAA, while in sec. 3.4.2 we evaluate the recognition power of the proposed activity descriptors and their alterations.

3.4.1 Spatial localisation

To evaluate spatial activity localisation, two challenging datasets with various human activities and camera motions have been used: the UCFsports Action Dataset [15] and the TV Human Interaction dataset [89]. The UCF dataset consists of 150 videos from...
real sports broadcasts, depicting 10 kinds of actions “in the wild”, and is one of the most
renown datasets for activity localisation, as spatial localisation ground truth is provided
with the dataset in the form of bounding boxes around the activities. The TV Human
Interaction dataset [89] consists of 300 videos from TV programs: 200 of them contain 4
different classes of daily interactions (handshake, highfive, kiss and hug), while the other
100 are labeled as negative. Here, spatial action localisation and recognition is performed
using only the first 200 videos.

Spatial action localisation results are summarized in Table 3.1 and Table 3.2 for the
UCF sports and TV human interaction datasets respectively. The localisation scores
are computed as the mean average of IoU (Intersection over Union) for the classes over
the tested videos, as in the SoA literature [71, 72, 90]. For the UCF sports dataset
evaluation is performed for all nine classes and compared with the current SoA [72], who
also provide results for all these classes. Table 3.1 shows that we outperform [72] almost
for all nine classes in terms of average IoU, improving activity localisation accuracy on
average by 10.4%. The two other related works on the UCF sports videos examine only
three of the nine activities, namely dive, ride and run. As table 3.1 shows, our method
improves upon [71, 90] for the diving activity, without requiring the expensive manual
annotation and learning needed by these methods. Activity localisation results are lower
than the SoA for riding and running, as these activities contain significant background
clutter that leads to errors in the resulting superpixels. A potential solution to this
would involve the use of prior knowledge and robust training for improved accuracy, as
in [71, 90]. However, this step will lead to higher computational burden and we will lose
the advantage of increased speed compared to the SoA.

For the TV Human Interaction dataset, our method is compared to [71, 90] who only
report results for the kiss action, and to [72] who report results for all classes. Our
approach outperforms all three methods for all categories, while it improves upon the

<table>
<thead>
<tr>
<th>Actions</th>
<th>Ours</th>
<th>72</th>
<th>71</th>
<th>90</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dive</td>
<td>50.9%</td>
<td>44.3%</td>
<td>22.6%</td>
<td>37.0%</td>
</tr>
<tr>
<td>Golf</td>
<td>49.7%</td>
<td>50.5%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Kick</td>
<td>59.6%</td>
<td>48.3%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Ride</td>
<td>36.7%</td>
<td>30.6%</td>
<td>63.1%</td>
<td>64.0%</td>
</tr>
<tr>
<td>Run</td>
<td>47.6%</td>
<td>33.1%</td>
<td>48.1%</td>
<td>61.9%</td>
</tr>
<tr>
<td>Skate</td>
<td>46.3%</td>
<td>38.5%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Bench</td>
<td>81.2%</td>
<td>54.3%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Swing</td>
<td>50.6%</td>
<td>20.6%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Walk</td>
<td>49.1%</td>
<td>39.0%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Average</td>
<td>54.2%</td>
<td>39.9%</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 3.1: UCF Sports average IoU for spatial localisation.
Table 3.2: TV Human Interaction IoU for spatial localisation.

<table>
<thead>
<tr>
<th>Actions</th>
<th>Ours</th>
<th>[72]</th>
<th>[71]</th>
<th>[90]</th>
</tr>
</thead>
<tbody>
<tr>
<td>HandShake</td>
<td>37.7%</td>
<td>26.9%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>High5</td>
<td>51.1%</td>
<td>32.9%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Hug</td>
<td>50.1%</td>
<td>34.2%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Kiss</td>
<td>52.5%</td>
<td>29.2%</td>
<td>18.5%</td>
<td>39.5%</td>
</tr>
<tr>
<td>Average</td>
<td>47.9%</td>
<td>30.8%</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

average results of IoU in [72] by 17.1%. The IoU rates are a little bit lower for this dataset, compared to UCF sports, most likely because the ground truth data for the TV human interaction dataset is only provided for the heads of people instead of their whole body.

3.4.2 Activity recognition

Activity recognition evaluation is carried out with the KTH [14], UCF sports [15], HOHA [12] and YouTube [77] action datasets, further analyzed in Sec. 2.3. We highlight the superiority of cAA against Dense sampling, the usefulness of a hybrid descriptor such as HOGHOF+Traj, especially with a sophisticated encoding scheme such as Fisher, leading to recognition rates reaching or surpassing the SoA. We demonstrate the descriptors’ performance for a wide range of vocabulary sizes, varying from 1000 to 4000 cluster centers for BoVW and 32 to 256 for VLAD and Fisher. The maximum vocabulary size was set based on the observations of previous works on activity recognition [46], [37], [21], which showed that these sizes led to optimal recognition rates. Furthermore, we observed experimentally that for vocabulary sizes outside this range, statistical deviations towards improvement or deterioration remain below 1%.

3.4.2.1 KTH results

One of the most popular datasets for human activity recognition is the KTH dataset [14]. The KTH videos consist of 2391 sequences, recorded in four different environments: outdoors $s_1$, outdoors with scale variation $s_2$, outdoors with different clothing $s_3$, and indoors $s_4$. Six different actions are performed by actors: box, handclap, handwave, jog, run, walk, while 25 subjects carried out these actions, for increased anthropometric variance. The dataset was split into 16 training and 9 testing videos, as suggested on [14], based on the subjects that perform the actions in order to evaluate our algorithm.

The results on the KTH action dataset in Table 3.4 and Fig. 3.7 show that cAA outperforms dense sampling for all vocabulary sizes, and for both HOGHOF and HOGHOF+Traj
descriptors with BoVW. We can also observe on the right picture of Fig. 3.7 that dense sampling needs a larger number of trajectory points than cAA in order to converge into a comparable visual vocabulary, since it analyzes interest points densely sampled over the entire frame. We believe that this happens because cAAs cover a much smaller part of each video frame, so they lead to far fewer trajectories of points in them. Dense sampling also entails a larger computational cost and more memory resources. However, we can see that trajectory inclusion in the representation scheme reduces the classification accuracy when cAA is concerned, whereas dense sampling leads to better accuracy. This is attributed to the fact that the trajectory descriptor is noisier inside the cAAs, possibly because of reduced detail of the trajectory points, leading to less accurate hard binning.

On the other hand, Fig. 3.8 shows that HOGHOF+Traj only improves when BoVW/dense sampling or Fisher encoding is used. This happens because Fisher encoding is the only one that includes mean and standard deviation information in its encoding, so it benefits from trajectory information when it is possible or does not use it at all when it is useless (i.e. when an activity does not performed on a specific location in the frame, or when the location changes throughout video samples). Other encoding schemes (BoVW, VLAD) cannot encode this information as they use Euclidean distance and hard binning in order to encode activity descriptors. Thus, inclusion of the trajectory descriptor decreases the recognition rates when location information is ambiguous. We can safely say that trajectory descriptor do not remarkably help in KTH case, but mostly spoil our prediction rates. This is mainly attributed to the fact that most activities take place in the whole video frame or present in different location. So, even Fisher encoding cannot significantly improve accuracy rates when trajectory is included. The Fisher vector achieves the highest recognition rates, close to the SoA (first row in Table 3.9), while the VLAD accuracy ranks as second best. Both Fisher and VLAD encoding schemes use smaller visual vocabularies, leading to better and faster recognition rates than the baseline BoVW.
Chapter 3. Hybrid HOGHOF descriptor and cAA for action recognition 45

Figure 3.8: Average Accuracy over all classes for each encoding scheme. Comparison among HOGHOF and HOGHOF+Traj is presented in each case.

<table>
<thead>
<tr>
<th></th>
<th>Box</th>
<th>Clap</th>
<th>Wave</th>
<th>Jog</th>
<th>Run</th>
<th>Walk</th>
</tr>
</thead>
<tbody>
<tr>
<td>Box</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Clap</td>
<td>0.014</td>
<td>0.986</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wave</td>
<td>0.056</td>
<td>0.076</td>
<td>0.868</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jog</td>
<td></td>
<td>0.965</td>
<td>0.035</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Run</td>
<td></td>
<td>0.174</td>
<td>0.826</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Walk</td>
<td></td>
<td></td>
<td>0.007</td>
<td>0.979</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AA</td>
<td>0.9398</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3.3: Confusion matrix of our best result for the KTH videos. Accuracy rates are recorded as the ratio of the correctly recognized actions to the groundtruth ones.

A confusion matrix of our best result, is presented in Table 3.3. The HOGHOF+Traj descriptor was encoded with a 32 cluster center Fisher vector and classification rates for each class are recorded. It is worthwhile to note that almost all classes are detected quite accurately, achieving recognition rates above 90%, except for the Run and Wave actions, which are mistakenly detected sometimes as Jog and Clap respectively.

3.4.2.2 UCF sports results

The UCF - sports videos [15] consist of 150 videos depicting 10 kinds of actions namely: Swinging on the pommel horse and the floor (Bench), at the high bar (HighB) and golf swinging (Golf). Dive, kick, weight-lift (Lift), horse-riding (Ride), run, skateboarding and
walking. The main challenges here are the moving camera and the variability observed among the actions. In order to increase the number of training videos for a more complete evaluation, we horizontally flip the original videos and use them for training, but we only use the original videos for testing as in [15].

In the first experiments, recorded in the first 4 rows of Table 3.5 and depicted in Fig. 3.9, trajectories extracted from cAA are tested against those sampled in a dense grid under a BoVW encoding scheme. It is obvious that sampling interest points from cAA outperforms dense sampling for all vocabulary sizes, both for the HOGHOF and the HOGHOF+Traj descriptors. HOGHOF+Traj performed better for most vocabulary sizes, achieving best recognition rates with 4000 cluster centers.

The next 2 rows in Table 3.5 aggregate the results with Fisher encoding. HOGHOF+Traj outperforms HOGHOF for all vocabulary sizes, while achieving the best results amongst all encoding schemes, with recognition rates comparable to the SoA, as seen in the second row of Table 3.9. The bottom 2 rows in Table 3.5 aggregate the results when VLAD encoding is used: again the hybrid HOGHOF+Traj performs better than HOGHOF, and its results are slightly better than BoVW.

Overall, the Fisher encoding schema not only performed better than BoVW and VLAD, but also required a far lower number of clusters to extract a discriminative vocabulary, and therefore entails a lower computational cost than BoVW. Comparisons of HOGHOF and HOGHOF+Traj comparisons are depicted in Fig. 3.10.

A closer look at our descriptor is provided by the confusion matrix in Table 3.6. Actions with significant camera motion, such as Dive, Kick and Walk are distinguished with very accurate recognition rates. While, others with cluttered background, such as Run and Ride, are easily misclassified, as the compensation algorithm cannot tackle the moving camera effect in such difficult occasions.
Chapter 3. Hybrid HOGHOF descriptor and cAA for action recognition

Figure 3.9: Average Accuracy over all classes when cAA applied against dense sampling on UCF sports action dataset. HOGHOF+Traj is depicted on the left figure, while HOGHOF is used on the right one.

<table>
<thead>
<tr>
<th>Encoding</th>
<th>Desc./Vocab.Size</th>
<th>1000/32</th>
<th>2000/64</th>
<th>3000/128</th>
<th>4000/256</th>
</tr>
</thead>
<tbody>
<tr>
<td>BoVW</td>
<td>HOGHOF (cAA)</td>
<td>73.73%</td>
<td>75.57%</td>
<td>74.56%</td>
<td>75.52%</td>
</tr>
<tr>
<td></td>
<td>HOGHOF+Traj (cAA)</td>
<td>74.37%</td>
<td>73.11%</td>
<td>72.66%</td>
<td>77.21%</td>
</tr>
<tr>
<td></td>
<td>HOGHOF (Dense)</td>
<td>70.67%</td>
<td>73.02%</td>
<td>74.14%</td>
<td>71.63%</td>
</tr>
<tr>
<td></td>
<td>HOGHOF+Traj (Dense)</td>
<td>69.46%</td>
<td>70.98%</td>
<td>69.70%</td>
<td>75.97%</td>
</tr>
<tr>
<td>Fisher</td>
<td>HOGHOF (cAA)</td>
<td>84.35%</td>
<td>85.12%</td>
<td>83.46%</td>
<td>79.97%</td>
</tr>
<tr>
<td></td>
<td>HOGHOF+Traj (cAA)</td>
<td>87.24%</td>
<td>86.33%</td>
<td>85.19%</td>
<td>82.75%</td>
</tr>
<tr>
<td>VLAD</td>
<td>HOGHOF (cAA)</td>
<td>66.49%</td>
<td>71.17%</td>
<td>73.49%</td>
<td>75.33%</td>
</tr>
<tr>
<td></td>
<td>HOGHOF+Traj (cAA)</td>
<td>68.87%</td>
<td>75.51%</td>
<td>75.63%</td>
<td>75.82%</td>
</tr>
</tbody>
</table>

Table 3.5: UCF aggregated recognition rates (i.e. percentage of correct classification) by measuring average accuracy over all classes.

Figure 3.10: HOGHOF and HOGHOF+Traj comparisons for each encoding scheme. Average Accuracy over all classes is reported.
Activity recognition in the presence of camera motions is investigated in [91], where camera motion is compensated by using motion-planes. Motion-planes are defined as regions that lie in the same depth, undergoing similar camera motion.

### 3.4.2.3 Hollywood1 (HOHA1) results

The Hollywood1 data [12] contains 8 different activities from Hollywood movies that are quite challenging to characterize: answer phone (AP), get out of car (GoC), handshake (HS), hug, kiss, sit down (SitD), sit up (SitU) and stand up (StandU). High variability in the human actors performing the actions, shot changes throughout the videos, viewpoint variations and a moving camera need to be tackled simultaneously for good recognition results. We followed the split proposed in [12] (219 videos samples for training and 211 for testing) and apply our method to it.

The 4 top rows of Table 3.7 give the average accuracy over all Hollywood classes when BoVW is applied for encoding. A comparison between cAA and Dense sampling for the HOHA dataset is also depicted in Fig. 3.11. Here, cAA improves recognition rates compared to dense sampling with HOGHOF, but not with the hybrid HOGHOF+Traj. HOGHOF+Traj does not achieve high accuracy due to the high intra-class variation and varying video frame size, which distort trajectory information, so the BoVW scheme cannot separate the feature space optimally, leading to poor recognition rates. This is attributed to the nature of trajectory descriptor, which in this dataset (HOHA) increases the ambiguity and consequently the distance between the activity descriptors and the visual vocabulary when hard binning is concerned.
Chapter 3. Hybrid HOGHOF descriptor and cAA for action recognition

Contrary to BoVW, Fisher encoding (2 middle rows in Table 3.7) exploits meaningful trajectory information and leads to comparable to SoA recognition rates with HOGHOF+Traj, as presented in the third row of Table 3.9. VLAD (bottom 2 rows in Table 3.7) performs similarly to BoVW, resulting in poorer performance with the hybrid descriptor, although it gives improved results for smaller vocabulary sizes (a.k.a. lower computational cost). Visualization of the above conclusions are depicted in Fig. 3.12.

The mean Average Precision(mAP) for all action classes is aggregated in Fig. 3.13. Our hybrid descriptor (i.e. HOGHOF+Traj), when encoded with a 64 cluster centers
Chapter 3. Hybrid HOGHOF descriptor and cAA for action recognition

<table>
<thead>
<tr>
<th>Encoding</th>
<th>Desc./Vocab.Size</th>
<th>1000/32</th>
<th>2000/64</th>
<th>3000/128</th>
<th>4000/256</th>
</tr>
</thead>
<tbody>
<tr>
<td>BoVW</td>
<td>HOGHOF(cAA)</td>
<td>29.93%</td>
<td>31.1%</td>
<td>32.67%</td>
<td>31.85%</td>
</tr>
<tr>
<td></td>
<td>HOGHOF+Traj(cAA)</td>
<td>19.44%</td>
<td>20.58%</td>
<td>21.17%</td>
<td>22.00%</td>
</tr>
<tr>
<td></td>
<td>HOGHOF(Dense)</td>
<td>25.82%</td>
<td>27.49%</td>
<td>28.00%</td>
<td>28.83%</td>
</tr>
<tr>
<td></td>
<td>HOGHOF+Traj(Dense)</td>
<td>20.05%</td>
<td>23.10%</td>
<td>23.48%</td>
<td>23.33%</td>
</tr>
<tr>
<td>Fisher</td>
<td>HOGHOF(cAA)</td>
<td>40.60%</td>
<td>39.73%</td>
<td>39.29%</td>
<td>39.30%</td>
</tr>
<tr>
<td></td>
<td>HOGHOF+Traj(cAA)</td>
<td>40.06%</td>
<td>41.28%</td>
<td>38.92%</td>
<td>39.39%</td>
</tr>
<tr>
<td>Vlad</td>
<td>HOGHOF(cAA)</td>
<td>34.18%</td>
<td>34.26%</td>
<td>35.88%</td>
<td>34.22%</td>
</tr>
<tr>
<td></td>
<td>HOGHOF+Traj(cAA)</td>
<td>24.15%</td>
<td>24.86%</td>
<td>25.68%</td>
<td>26.59%</td>
</tr>
</tbody>
</table>

Table 3.7: Mean average precision over varying vocabulary sizes for BoVW, Fisher and VLAD encoding, tested on Hollywood action dataset (HOHA).

Fisher scheme, leads to very accurate recognition rates that outperform past related work [12], [67] and are comparable to [22]. The main disadvantage is noted in SitUp action, which doesn’t reach quite good recognition rates, as it is usually misclassified as StandUp action and limits the overall mAP rate. Get out of Car (GoC), Hug, Kiss and Stand up, on the other hand, achieve quite high recognition rates, close the SoA, supporting the usefulness of our technique. Furthermore, the use of cAA provides a computationally efficient representation algorithm that is much faster than other related works and Fisher proved to be a very robust tool which can be leveraged in order to benefit from a set of diverse action descriptors and achieves a high classification rate.
3.4.2.4 YouTube results

The YouTube data [77] contains 11 action categories from YouTube leading to a total of 1597 sequences: basketball shooting, biking/cycling, diving, golf swinging, horse back riding, soccer juggling, swinging, tennis swinging, trampoline jumping, volleyball spiking, and walking with a dog.

This dataset is abundant in camera motion, severe human anthropometric differences, viewpoint variance, cluttered background, different types of occlusion and changing illumination conditions. We follow the original setup [77] which leaves one group out cross validation for a pre-defined set of 25 folds. Average accuracy over all classes is reported as performance measure.

The 4 top rows of Table 3.8 give the average accuracy over all YouTube action classes when BoVW encoding is taken into account. A comparison between cAA and Dense sampling for the YouTube dataset is also depicted in Fig. 3.14. We can see that cAA improves against dense sampling in both representation schemes and especially when HOGHOF+Traj is considered.

More information about the BoVW performance follows on the two bottom figures in Fig 3.15 where we can see that HOGHOF dominates against HOGHOF+Traj and particularly when dense sampling is concerned. This is attributed to the unconstrained nature of these videos, which do not contain consisted long trajectories, so the inclusion of trajectory coordinates does not provide reliable location information because severe viewpoint and scale variance often introduce severe noise in the representation vectors.

Observing the two top figures of Fig. 3.15 we can see that Fisher encoding exploits meaningful trajectory information and leads to comparable to SoA recognition rates when combined with HOGHOF+Traj action descriptor. VLAD performs similarly to
Figure 3.15: Average Accuracy over all classes when Activity Area applied against dense sampling. HOGHOF+Traj descriptors were used on the left figure, while HOGHOF was used in the right one.

<table>
<thead>
<tr>
<th>Encoding</th>
<th>Desc./Vocab.Size</th>
<th>1000/32</th>
<th>2000/64</th>
<th>3000/128</th>
<th>4000/256</th>
</tr>
</thead>
<tbody>
<tr>
<td>BoVW</td>
<td>HOGHOF(cAA)</td>
<td>48.91%</td>
<td>50.15%</td>
<td>50.61%</td>
<td><strong>52.64%</strong></td>
</tr>
<tr>
<td></td>
<td>HOGHOF+Traj(cAA)</td>
<td>46.87%</td>
<td>50.05%</td>
<td>50.05%</td>
<td>50.68%</td>
</tr>
<tr>
<td></td>
<td>HOGHOF(Dense)</td>
<td>50.94%</td>
<td>47.71%</td>
<td>48.92%</td>
<td>49.10%</td>
</tr>
<tr>
<td></td>
<td>HOGHOF+Traj(Dense)</td>
<td>35.97%</td>
<td>39.61%</td>
<td>42.09%</td>
<td>44.30%</td>
</tr>
<tr>
<td>Fisher</td>
<td>HOGHOF(cAA)</td>
<td>75.50%</td>
<td>75.95%</td>
<td>76.10%</td>
<td><strong>76.48%</strong></td>
</tr>
<tr>
<td></td>
<td>HOGHOF+Traj(cAA)</td>
<td>73.70%</td>
<td>74.73%</td>
<td>75.75%</td>
<td>76.18%</td>
</tr>
<tr>
<td>Vlad</td>
<td>HOGHOF(cAA)</td>
<td>52.08%</td>
<td>53.50%</td>
<td>55.66%</td>
<td>59.07%</td>
</tr>
<tr>
<td></td>
<td>HOGHOF+Traj(cAA)</td>
<td>50.13%</td>
<td>53.71%</td>
<td>54.62%</td>
<td>57.10%</td>
</tr>
</tbody>
</table>

Table 3.8: Mean average precision over varying vocabulary sizes for BoVW, Fisher and VLAD encoding, tested on YouTube action dataset.

BoVW, resulting in poorer performance than Fisher, although it gives improved results for smaller vocabulary sizes (a.k.a. lower computational cost). Comparison with SoA [21] is given in the 4th row of Table 3.9, however no significant conclusions can be made as soon as the authors of [21] used the first version of YouTube dataset which contained less data than the one used in this thesis.
Chapter 3. Hybrid HOGHOF descriptor and cAA for action recognition

<table>
<thead>
<tr>
<th></th>
<th>[60]</th>
<th>[20]</th>
<th>[12]</th>
<th>[21]</th>
<th>[22]</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>KTH</td>
<td>92.1%</td>
<td>91.4%</td>
<td>91.8%</td>
<td>94.2%</td>
<td>94.5%</td>
<td>93.98%</td>
</tr>
<tr>
<td>UCF</td>
<td>85.6%</td>
<td>86.7%</td>
<td>N.A.</td>
<td>88.2%</td>
<td>N.A.</td>
<td>87.24%</td>
</tr>
<tr>
<td>HOHA</td>
<td>N.A.</td>
<td>24.7%</td>
<td>38.4%</td>
<td>N.A.</td>
<td>56.8%</td>
<td>41.28%</td>
</tr>
<tr>
<td>youTube</td>
<td>N.A.</td>
<td>N.A.</td>
<td>N.A.</td>
<td>84.2%</td>
<td>N.A.</td>
<td>76.48%</td>
</tr>
</tbody>
</table>

Table 3.9: Comparison to SoA average accuracy over all classes for the three action datasets.

<table>
<thead>
<tr>
<th></th>
<th>KTH</th>
<th>UCF</th>
<th>HOHA</th>
<th>youTube</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dense</td>
<td>2.33</td>
<td>0.44</td>
<td>1.4</td>
<td>0.6</td>
</tr>
<tr>
<td>cAA</td>
<td>2.62</td>
<td>0.46</td>
<td>1.47</td>
<td>0.59</td>
</tr>
</tbody>
</table>

Table 3.10: Time consumption on the three action datasets

3.4.3 Computational time and memory consumption

In table 3.10 we aggregate the computational cost in fps (frame per seconds) and the mean area that taken into consideration to extract the activity descriptors in all action datasets. Observing these results and keeping in mind the recognition rates that we acquired in the previous section (Sec. 3.4.2), we note that the cAA not only increased the recognition rates over all datasets but also improved the computational time.

3.5 Discussion

An improved approach for the recognition of human activities, from challenging videos including large viewpoint variations and camera motion, is presented in this chapter. After motion compensation, so-called compensated Activity Areas (cAAs) are created, and dense sampling applied in them, so as to leverage the advantages of dense information, but only in regions of interest that will give meaningful results. Thus, the use of cAAs increases the reliability of our method, since only relevant data will be processed, while at the same time helping to maintain the computational cost remain at reasonable levels.

A hybrid descriptor (i.e. HOGHOF+Traj) is created for this data, incorporating both local and global information, and compared with three different commonly used encoding frameworks, namely BoVW, Fisher and VLAD encoding. The experiments are conducted on standard benchmark datasets, some of which (e.g. HOHA1 and youTube) are quite challenging. The proposed method improves recognition rates without incurring a heavy computational cost, thanks to the focus on interest points inside cAAs only.
Having deployed a robust activity descriptor on unconstrained scenarios, we proceed with constrained videos focused on Activities of Daily Living (ADL) scenarios discussed in Chapter 4. Further extension is introduced to the interest point detector, while time invariant trajectory structure is proposed.
Chapter 4

Recognition of ADLs from optimal trajectories and Motion Boundary Activity Areas (MBAA)

Contents

4.1 Introduction ................................................. 56
4.2 ADL Representation ......................................... 58
  4.2.1 Motion Boundary Activity Areas (MBAA) ............ 59
  4.2.2 ADL descriptor ........................................ 65
  4.2.3 Temporal Trajectory Segmentation based on Statistical Sequential Change Detection (SSCD) ............ 68
4.3 ADL recognition schemes ................................. 72
  4.3.1 Bag-of-Visual-Words ............................... 72
  4.3.2 Fisher encoding ..................................... 73
4.4 Experiments on ADL benchmark datasets ............... 74
  4.4.1 Action datasets ...................................... 74
  4.4.2 Experimental parameters ............................ 77
  4.4.3 Experimental results ............................... 78
  4.4.4 Dem@Care results .................................. 82
  4.4.5 CHUN results ....................................... 85
4.5 Discussion ................................................. 85
Activities of Daily Living (ADL) differ from activities usually captured from a moving camera in an unconstrained scenario as we saw in Chapter 3. In ADL videos, fixed cameras are usually placed in a smart home and monitor the behavioural and activity pattern of a human user. Motion and appearance features are not adequate in these scenarios as severe occlusions are often present, hiding the activity being performed. Visible activities usually follow a motion pattern performed with a specific orientation changes in time, while a large part of the video frame is not taken into account, as activities usually take place around human users. Motivated by these observations we suggest to include location information by a trajectory descriptor, introduce Sequential Statistical Change Detection (SSCD) on the orientation of motion patterns in order to build time-invariant trajectories and propose Motion Boundary Activity Areas that localize where an activity occurs inside a video frame to accelerate computational time and recognition rates.

4.1 Introduction

The increasing use of technology is making its presence felt in many aspects of daily life, as surveillance and monitoring systems are being used in various applications, ranging from security to home-based monitoring, for example in smart homes or assisted living scenarios. The recognition of human activities is essential for such monitoring, and is the focus of this chapter, which presents a solution for the recognition of Activities of Daily Living (ADLs).

Despite the very accurate recognition results the State-of-the-Art (SoA) for activity recognition achieves, these methods still entail a high computational cost. A serious issue is related to the dense sampling that these techniques adopt in order to build a representation scheme. The SoA is using dense sampling techniques rather than the earlier interest point detectors, which result in interest points that are too sparse. However, dense sampling introduces a higher computational cost to the algorithm, and a less costly approach is required. In Chapter 3.2.2 we proposed dense sampling inside compensated Activity Areas (cAA) and succeeded to reduce the computational cost without reducing accuracy. In this Chapter, we reduce the computational cost even more, by sampling interest points around human body parts, thus achieving even higher recognition rates.

Furthermore, the SoA [21], [62] has shown that trajectory descriptors can be added to feature descriptors and improve activity recognition accuracy. However, the tracking algorithm of [21], [62] is quite simplistic and includes noisy features. Motivated by this, we propose an improved KLT tracker, whose correspondences over time are rectified by a...
RANSAC homography estimator. In general this leads to outliers that are fewer than 10% of the point matches, because optical flow already provides quite similar correspondences, resulting in a fast RANSAC calculation. Recent work of the same author [43] showed that this disadvantage can also be overcome by rectifying SURF descriptor matched over time using RANSAC. However, this technique entails a larger computational cost than ours due to the cumbersome matching procedure.

On a higher representation level (i.e. trajectory descriptor), temporal variations, which may be produced from different camera frequency recording or different activity styles, are systematically neglected. The SoA prefers to run exhaustive cross-validation experiments for selecting the appropriate trajectory length rather than computing it automatically. In order to solve this issue, we propose in Section 4.2.3.2, a novel technique that monitors HOF descriptors over an action trajectory’s lifetime and terminates them whenever a change in HOF occurs.

A high computational cost is also introduced in the BoVW framework, where $K$-Means clustering is traditionally used for creating visual vocabularies. This technique however demands a large number of $K$ samples in order to converge to a solution resulting in a high computational cost. Furthermore, the Chi-Square($\chi^2$) distance for hard binning, which is broadly used on the encoding process, loses a large amount of information between the vectors. Inspired by the highly accurate recognition rates obtained in section 3.4, we demonstrate the superiority of Fisher encoding against BoVW for fast and accurate ADL classification. In this chapter we further support these findings by the experimental results presented in Section 4.4.

Summarizing, this Chapter presents some novel enhancements to the representation and encoding parts of our activity recognition system whose essential blocks, depicted in Fig. 4.1, comprise of:

1. Dense interest point sampling in regions of changing motion, Motion Boundary Activity Areas (MBAAAs), which are informative about the activity taking place and can reduce the computational cost of action representation while maintaining high recognition rates.

2. Improved trajectories calculated based on dense optical flow using the KLT tracker [88] after rectifying bad correspondences using RANSAC. Higher accuracy and coherency in the dense flow trajectories are achieved by removing outlier displacements, something that has been overlooked in the existing literature.

3. Sequential statistical change detection (SSCD) for the online extraction of the activity trajectories’ temporal length, related to actual changes in motion.
4. We have determined that probabilistic encoding with soft binning, such as Fisher, which is commonly used in image processing, is most appropriate for achieving accurate activity recognition results.

For comparison with the SoA techniques, we applied our method to benchmark video datasets of ADLs. Albeit very useful for comparisons, these datasets feature a few actors in a controlled environment and limited context. They also do not feature significant anthropometric variations, which may limit their applicability in certain real situations.

In this work we also present more realistic and challenging ADL videos for benchmarking purposes, recorded on the premises of the Greek Association for Alzheimer’s Disease and Related Disorders (GAADRD) [17] and at the Nice University Hospital (CHUN).

This chapter is organised as follows: Sec. 4.2 presents the adopted ADL representation scheme, with the Motion Boundary Activity Areas (MBAA) in Sec. 4.2.1, and the hybrid ADL descriptor in Sec. 4.2.2 and its optimal trajectories (Sec 4.2.3.2). Sec. 4.3 follows with the recognition and encoding schemes. Experimental results are given in Sec. 4.4.

4.2 ADL Representation

Action representation is the first, essential part of an activity recognition algorithm, where characteristic descriptors of the data are extracted, before being introduced to the action recognition stage. The block diagram of Fig. 4.2 depicts the entire Action Representation procedure.

Motion is analyzed by estimating the optical flow (OF), from which areas of changing motion (MBAA, Sec. 4.2.1) are extracted. Interest points (central points in the blue rectangles of Fig. 4.2(d)) are densely sampled and matched with interest points from previous frames using an enhanced KLT tracker [88]. The tracked points are represented by red lines for a sample video frame in Fig. 4.2(e). HOG and HOF descriptors are computed around each interest point and used to construct action cuboids at several scales. Finally, in order to determine the temporal extent of the trajectories, sequential statistical change detection based on CUSUM is applied to the HOF descriptors of the tracked points, leading to optimal trajectories (in Sec. 4.2.3). The resulting subsequences with the enhanced multi-scale HOGHOF descriptor are then used in the recognition framework for fast ADL classification, described in section 4.3.
4.2.1 Motion Boundary Activity Areas (MBAA)

As explained in Sec. 3.1, action representation with dense sampling [21] achieves SoA recognition rates, but at a high computational cost, which can become cumbersome in practice. Compensated Activity Areas (cAA) were presented in Section 3.2.2, in order to densely sample interest points around moving pixels. In this section, in order to reduce the cost of dense sampling, but maintain a high discriminative ability, we follow a different sampling technique: We detect areas of motion that undergo change throughout time (i.e. motion boundaries), which contain meaningful information, and sample dense spatio-temporal interest points only in these areas. We name these regions **Motion Boundary Activity Areas** (MBAA), which are binary masks used for localizing regions
Figure 4.2: Block diagram for ADL visual features representation. OF estimates are used to extract MBAAs, dense sampling within MBAAs on four spatial scales results in interest points that are tracked by KLT and construct trajectories. Outliers are removed from trajectories by the application of RANSAC and HOGHOF features are derived on multiple scales around the tracked points. CUSUM SSCD is applied to the resulting trajectories’ HOFs to detect changes in motion and form final ADL representation.

of a video frame where optical flow values change during a predefined temporal window $W$.

We also depart from related work [21] where Motion Boundary Histograms (MBH) were introduced. The use of MBH is meaningful only when camera motion is taken into account. However, it loses a large amount of information in a fixed camera scenario, as it only encodes motion edges. On the other hand MBAA use changes in motion patterns throughout time in order to localize motion boundaries and represent motion patterns using HOF histogram instead. This provides a fast action descriptor, more discriminative than AA and dense sampling, as we can see in experimental section 4.4, and more compact than the previous ones.

Methodology:

MBAAs are derived based on higher order statistical processing of OF gradients, to find the pixels where OF changes significantly. For MBAA extraction, once the OF is estimated, its gradients are computed over successive frames by applying the horizontal and vertical Sobel operator and calculating the square root of their sum of squares (root sum square, RSS). Gradients of optical flow are indicative of motion edges, producing higher values on motion boundaries, and lower values in smoother motion regions. In
them, they are a strong indicator of the regions correspond to changes in the motion of human activities. In practice, high motion gradients (i.e. changes in motion) can also be induced by background clutter or illumination variations, introducing erroneous noisy values to our interest point sampling. To extract MBAAs, for every time instant $k$, motion gradients are accumulated over a predefined temporal window $W$, which after cross validation in the URADL dataset was defined to be equal to $fps/2$. Then, to separate true motion boundaries from noise-induced ones, we make the assumption that noisy OF gradients can be modeled as a Gaussian distribution (hypothesis $H_0$), while true changes in OF induce a deviation from Gaussianity (hypothesis $H_1$), depicted in Eq. 4.1 for pixel $r$ and frame $k$:

$$H_0 : u^0_k = z_k(r)$$
$$H_1 : u^1_k = u_k(r) + z_k(r)$$

where $u^i_k(r), i = \{0, 1\}$ denotes the estimated OF gradient for each hypothesis $H_i$, $u_k(r)$ is the true OF gradient and $z_k(r)$ the additive noise induced OF gradient.

The assumption of Gaussianity for noise-induced OF gradients can be considered valid, as OF gradients estimates originate from sums of many i.i.d. random variables, which ultimately converge towards the same Gaussian distribution based on the Central Limit Theorem [84]. This does not apply to motion-induced OF gradients, as they change depending on the video and the activities in it, unlike noise-induced OF (which contains far fewer outliers).

**Kolmogorov-Smirnov test:**

To verify that the gradients of constant OF follow Gaussian distribution as they are noise-induced, we employ the Kolmogorov-Smirnov (K-S) test [85], which checks if the empirical data distribution matches a reference distribution, in this case the Gaussian fit to our data. We apply the K-S test on OF gradients from 10 videos randomly sampled from the URADL dataset [11] that are gathered in two groups: flow gradients inside manually extracted ground truth MBAAs and flow gradients outside of these MBAAs. Fig. 4.3 shows the empirical cumulative distribution function (cdf) of these flow gradients and the cdf of their Gaussian approximation for both cases: it is evident that flow gradients outside MBAAs are indeed adequately approximated by a Gaussian distribution, while flow gradients in MBAAs that are time-varying do not match the Gaussian model. The Root Mean Squared Deviation (RMSD) for data inside an MBAA, averaged over the 10 URADL videos, is 0.3847, while outside MBAAs it is 0.1975.

Fig. 4.4 depicts the approximation graphs between our real data and those produced when data is modeled by a Gaussian pdf. The data has been sampled from 10 different
video samples from the URADL dataset. It is obvious that data undergoing a change in their motion gradients (inside MBAAs) produce much higher OF gradient values and deviate significantly from those that do not (outside MBAAs). Furthermore we can see that noise induced data follows a Gaussian distribution with a mean value much lower than 1 and close to 0, while pixels undergoing changing motion do not, giving unpredictable OF gradients.

**Kurtosis metric:**

The Kurtosis has been shown to be a reliable test of deviations from Gaussianity [92], whose value becomes equal to zero for Gaussian data. Based on this property, we build a binary mask that separates time-varying OF values from those that remain constant in the time interval being examined. In order to accurately estimate the empirical value of the kurtosis, we apply the unbiased Kurtosis estimator of [93] to the OF gradients $u_k(r)$.
at pixel $r$ and in frames $k$ within the temporal window $W$ under examination:

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

We examined 10 videos from the URADL dataset [11] where we manually extracted MBAAs as ground truth. The kurtosis values of the flow gradients inside and outside the MBAAs are shown in Fig. 4.5, where their difference is very clear: in Fig. 4.5 (a) the vertical axis, showing the kurtosis values, has values up to 0.1, with most kurtosis values below $h \approx 2 \cdot 10^{-2}$, while in Fig. 4.5 (b), for data in the MBAAs, kurtosis values are much higher, surpassing 300 in some cases. More detailed quantitative results in Table 4.1 show that the kurtosis reliably separates pixels with changing OF from the rest. Similar empirical results with many other ADL datasets led us to the following rule for binarizing the kurtosis of flow gradients:

$$MBAA(x,y) = \begin{cases} 
0 & \text{if } G_2[x,y] < h \\
1 & \text{else}
\end{cases}$$

The threshold $h$ was manually defined by the mean value of the kurtosis of the 10 aforementioned URADL videos, added to its standard deviation, which was weighted in this case by a factor of 2.

**MBAAs vs AAs:**

These theoretical and empirical results lead us to use the kurtosis as a reliable metric for extracting MBAAs. In section 3.2.2 we introduced Activity Areas (AA), which
contain pixels with true motion, that are separated from noisy ones via higher order statistical analysis over the frames being examined. In this work, we consider that the most descriptive information about a video exists in regions where motion changes, as data from smoother motion regions is not as informative and discriminative. For this reason, we introduce MBAAs, which are binary masks that localize regions of a video frame where \textit{optical flow values change} during a predefined temporal window. The size of the temporal window that provides the most meaningful results is determined by the frame per second (fps) recording rate of the videos. In particular, we observed from thorough cross validation on ADL datasets (i.e. KTH, URADL, DemCare) that the best size for the temporal window is usually equal to half of the fps of the video recording, so
Figure 4.6: MBAAs are much smaller than AAs but are located in regions of changing motion that are informative about the activity taking place.

we use \( W = (fps/2) \). As an example, Fig. 4.6 (a), (b) shows motion vectors between two frames of a URADL video and their dominant directions, while Fig. 4.6 (c), (d) shows the corresponding AAs and MBAAs. MBAAs produce significantly less data, which is however highly informative as it is located in regions of changing motion, relevant to the activity taking place.

To further support the use of MBAAs, we run exhaustive experiments on URADL and compare recognition accuracy when using AAs [87] and MBAAs. MBAAs significantly reduce the representation computational cost, requiring about 5 fewer hours, and also improve recognition accuracy by increasing it on average by 8 – 10%. These comparisons show that we acquire better recognition results by working only inside MBAAs, indicating that MBAAs indeed contain more informative data and lead to fewer false alarms than AAs, which also contain constantly moving regions. Extensive comparisons between the two masks were conducted on URADL dataset and are provided in Sec. 4.4.3-Table 4.3.

4.2.2 ADL descriptor

We propose a rich, informative and computationally efficient approach for ADL representation by densely sampling interest points in MBAAs and tracking them via KLT, combined with RANSAC for outlier rejection. Appearance and motion information is extracted locally on \( n \) scales to fully describe trajectory points at several levels of detail and account for scale and anthropometric variations. Global spatial localisation
supplements local information by adding trajectory point coordinates, resulting in a rich hybrid (localeglobal) descriptor.

4.2.2.1 Global Trajectory Descriptors

Once MBAAs are formed, we extract multi-scale interest points as follows: in each frame, we extract “candidate points” $P^s_{c}$ on $s = [1, 2, ..., n]$ scale grids and examine if they belong to an MBAA. When more than half of the block’s candidate points $P^s_{c}$ are inside an MBAA, the central point of that grid block is defined as an interest point $P^s_{i}$. Otherwise, there is no interest point for that block at that scale, so the resulting interest points form a subset of the candidate points $P^s_{i} \subseteq P^s_{c}$. The accumulated interest points $P^s_{i}$ are tracked by a pyramidal implementation of the KLT (Lukas-Kanade) tracker [88]. Outliers in the resulting trajectories are expected to be erroneous point correspondences and are removed via RANSAC [94]. RANSAC, often used in image matching, is based on the computation of a homography matrix between sets of points for a pair of images. The procedure converges to an optimal homography matrix $H$ by examining random subsets of the data, which is then used to eliminate possible outliers in the point correspondences [94].

The resulting (inlier) raw trajectory coordinates are concatenated to form the trajectory descriptor, called the “action trajectory”, which incorporates global information in our representation.

4.2.2.2 Local HOG HOF Descriptors

The global information of the extracted action trajectories is combined with local appearance and motion descriptors, namely histogram of oriented gradients (HOG) and oriented OF (HOF), which are computed around each trajectory interest point $P^s_{i}$. Both HOGs and HOFs are computed at $n$ spatial scales around each $P^s_{i}$ interest point, to guarantee scale invariance.

HOGs are computed by applying a Sobel operator to gray-scale illumination values on the horizontal and vertical axis, extracting image edges in all directions to describe appearance, while HOFs are derived from the horizontal and vertical OF values. At each scale $s$ and grid with step size $k_s$, blocks are formed around each trajectory interest point with width and height related to the grid step size by $k_s + 8$. Each of these blocks is divided into $n_x \times n_y$ cells, where HOGs and HOFs are estimated. We depart from the SoA [12], [20] by following a mirroring technique to create direction invariant HOGs and HOFs, where mirrored shape and motion directions are mapped into the same bin. This
results in the same histogram for a human that runs from left to right, or the opposite, ensuring invariance to the direction of motion. The resulting histograms are normalized to the same range of values by dividing them with their $L_2$ norm.

### 4.2.2.3 Hybrid ADL descriptor

HOGs and HOFs are estimated for each interest point $P_i$ and, when an action trajectory is completed, they are concatenated, forming a spatio-temporal HOGHOF descriptor. As mentioned in Sec. 4.1 and detailed in Sec. 4.2.3, the temporal length of action trajectories is determined by applying statistical sequential change detection (SSCD) to the HOFs, to find when the OF distribution changes, corresponding to the termination of a sub-activity. Our experimental results showed that SSCD leads to trajectories of at least 15 frames.

In order to create a descriptor of fixed length from temporally varying trajectories while retaining their information, the HOGs and HOFs of each action trajectory are segmented into $n_t$ equally sized sets over time, averaged, normalized by the $L_2$ norm and concatenated. This segmentation and normalization leads to trajectory descriptors of the same length, which still capture the characteristics of sub-activities of varying temporal extent extracted as in Sec. 4.2.3.

So, in our case, let $HOG(x, y, t)$ and $HOF(x, y, t)$ be the histograms extracted inside a trajectory block $B_{sc} = (x, y, w_{sc}, h_{sc})$, around an interest point with coordinates $(x, y)$, with sampling of size 8 for each of the 4 scales: $(w_{sc}, h_{sc}) = (8, 8)(16, 16)(24, 24)(32, 32)$, where $w_{sc}$ and $h_{sc}$ are the width and height respectively, and the index $sc$ denotes the different sizes of the block, corresponding to scale size. The resulting spatio-temporal
Chapter 4. *Optimal trajectories and MBAA for ADL recognition* 68

descriptor around each interest point \((x, y)\) is the \(L_2\) normalized concatenation of the averaged histograms within each temporal sub-volume that is formed by dividing the initial descriptor by \(n_t\).

### 4.2.3 Temporal Trajectory Segmentation based on Statistical Sequential Change Detection (SSCD)

Current SoA action recognition methods that incorporate trajectory information in a BoVW representation scheme \([21]\) consider that trajectories have a fixed (manually derived) temporal length. However, in practice there may be large variations in the way each activity is carried out, as similar actions are often performed at different time scales, dependent on camera properties like frames per second being recorded, or anthropometric variations between the human subjects that perform the activities.

To address these issues, we propose a novel solution for the extraction of temporal trajectories based on statistical sequential change detection (SSCD) \([95], [96]\), which finds moments of change in the HOFs corresponding to meaningful changes in the activities taking place. The Cumulative Sum(CUSUM) \([97]\) test is applied to each trajectory’s HOFs for meaningful temporal segmentation and consequently improved recognition results. A strength of CUSUM, and SSCD in general, is that it is able to detect an unknown change point between two unknown data distributions \([98]\). Additionally, sequential algorithms only process the data made available up to the current point in time, allowing them to operate in an online manner.

In order for CUSUM to detect changes in the video’s motion, the original and current distributions of the data need to be extracted empirically \([96]\). Data distributions can be approximated and changes between them can be detected by using as few as \(W_0 = 8 – 10\) video frames. In this work, the data comprises of a set of HOFs, i.e. a set of \(N \times 1\) histograms, so we are essentially finding the distribution of these histograms. The initial distribution \(f_0\), is approximated from the first \(W_0\) motion histograms (HOFs) \(f_0 = f\{\bar{h}_1, \bar{h}_2, ..., \bar{h}_{W_0}\}\) and the current data distribution \(f_1\), uses the latest \(W_1\) observations, including the current frame’s HOF \(\bar{h}_k : f_1 = f\{\bar{h}_{k-(W_1-1)}, \bar{h}_{k-W_1}, ..., \bar{h}_k\}\). For simplicity, in the sequel we consider that \(W_0 = W_1\), as this does not affect the accuracy of the distribution approximation. Thus, for modeling the distribution of the HOFs, we have as input data a \(W_0 \times N\) matrix, where \(W_0\) is the number of samples and \(N\) the number of bins, i.e. the initial length of each HOF. The minimum time that SSCD takes to converge \((W_0 + W_1)\) is on average 1 second (20-30 frames), while we do not consider that trajectories that are larger than 4 seconds (80-120 frames). This algorithm performs well when camera frequency is larger than 8 frames per seconds.
4.2.3.1 Multivariate Gaussian HOF Modeling

We make the assumption that the initial and current distributions of the HOFs can be approximated by a multivariate Gaussian distribution. We validate this assumption by applying the Mardia tests, which are detailed below. The Gaussian distributions of the HOF pdfs are given by:

\[ f_l(\bar{h}_i) = \frac{1}{(2\pi)^{N/2}\|C_l\|^{1/2}} \exp\left(-\frac{1}{2}(\bar{h}_i - \bar{\mu}_l)^T C_l^{-1}(\bar{h}_i - \bar{\mu}_l)\right), \ l \in \{0, 1\}. \]  

These pdfs can be approximated by estimating their mean \( \bar{\mu}_l, l = \{0, 1\} \):

\[ \bar{\mu}_0 = \frac{1}{W_0} \sum_{i=1}^{W_0} \bar{h}_i, \ \bar{\mu}_1 = \frac{1}{W_1} \sum_{i=k-(W_1-1)}^{k} \bar{h}_i \]

and covariance matrices \( C_l, l = \{0, 1\} \):

\[ C_0(i, j) = E[(\bar{h}_i - \bar{\mu}_0)^T (\bar{h}_j - \bar{\mu}_0)] = \frac{1}{W_0} \sum_{i=1}^{W_0} (\bar{h}_i - \bar{\mu}_0)^T (\bar{h}_j - \bar{\mu}_0) \]

\[ C_1(i, j) = E[(\bar{h}_i - \bar{\mu}_1)^T (\bar{h}_j - \bar{\mu}_1)] = \frac{1}{W_1} \sum_{i=k-(W_1-1)}^{k} (\bar{h}_i - \bar{\mu}_1)^T (\bar{h}_j - \bar{\mu}_1) \]

We make the assumption that the HOFs, representing the distribution of the directions of optical flow in each temporal window, are uncorrelated, as in most cases there are no great variations in motion over time and between nearby spatial neighborhoods. Under the Gaussian assumption, this leads to the representation of HOFs as independent and identically distributed (i.i.d.) in each temporal window. In practice, the joint pdf of \( W_0 \) HOFs can then be represented by the product of the individual HOFs in each frame \( k = \{1, 2, ..., W_0\} \). The HOFs’ covariance matrix is diagonal, as the data is independent between different time instances \( i \neq j \), with zero cross-correlation for different trajectories and auto-correlation equal to the square of the variance:

\[ C_{0|1}(i, j) = \begin{cases} 
0 & i \neq j, \\
\sigma^2 & i = j.
\end{cases} \]

In order to validate the Gaussian approximation of our data, we applied appropriate tests of multivariate normality namely the Mardia tests [99], [100]. The Mardia tests involve the calculation of (a) the multivariate skewness, (b) the multivariate skewness corrected for small samples and (c) the multivariate kurtosis. When the null hypothesis for all these tests is validated, it implies that the residuals are normally distributed. In order to verify the Gaussianity of the data, we normalize samples \( \bar{h}_i \) via \( \bar{h}_i^{\text{norm}} = (\bar{h}_i - \bar{\mu})/\sigma \), and apply the Mardia tests to \( \bar{h}_i^{\text{norm}} \) to examine its normality.
For a set of $L d$-dimensional random variables $X = [\bar{x}_1, \bar{x}_2, ..., \bar{x}_L], \forall \bar{x}_i \in R^d$, where $d$ is the dimension of $\bar{x}_i, i = [1, 2, ..., L]$, the measure of multivariate skewness is:

$$b_{1,d} = \frac{1}{L^2} \sum_{i=1}^{L} \sum_{j=1}^{L} \left[ (\bar{x}_i - \bar{\mu})^T \Sigma^{-1} (\bar{x}_j - \bar{\mu}) \right]^3$$

where $\Sigma$ is the sample covariance matrix of $X$. Mardia showed that under the null hypothesis of multivariate normality, $\frac{d}{6} b_{1,d}$ is asymptotically distributed as a Chi-square distribution with $\frac{d(d+1)(d+2)}{6}$ degrees of freedom. A measure of the multivariate kurtosis is given by:

$$b_{2,d} = \frac{1}{L} \sum_{i=1}^{L} \left[ (\bar{x}_i - \bar{\mu})^T \Sigma^{-1} (\bar{x}_i - \bar{\mu}) \right]^2$$

Under the null Mardia hypothesis, $b_{2,d}$ is asymptotically normally distributed with mean $d(d+2)$ and variance $8d(d+2)/L$.

Taking into consideration these measures, we applied the Mardia skewness and kurtosis tests to 24 different samples from the KTH activity recognition dataset, in order to validate the Gaussian assumption of our HOF data. The KTH videos contain a variety of human actions, like walking, jogging, boxing etc, performed by various subjects both indoors and outdoors. Our experimental results showed that the skewness null hypothesis is accepted in more than 99.6% of the cases, while the kurtosis null hypothesis is always accepted, indicating that our data (the HOF orientations) can indeed be modeled by a multivariate normal distribution.

### 4.2.3.2 Statistical Sequential Change Detection (SSCD)

Based on the empirical Gaussian approximations of the HOFs’ distribution before and after a change, we approximate the log-likelihood ratio, i.e. the test statistic $T_k$ for frame $k$ in the implementation of CUSUM [101], by:

$$T_k = \ln \left( \frac{f_1(\bar{h}_k)}{f_0(\bar{h}_k)} \right) = \ln(f_1(\bar{h}_k)) - \ln(f_0(\bar{h}_k))$$

CUSUM is applied via the computationally efficient iterative form of Page [101]:

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

From Eq. (4.3) we can see that when the current histogram is close to the initial one (i.e. there has been no change in motion), the log-likelihood ratio will be nearly zero, while when the current histogram deviates from the initial one, there is an increase in $T_k$ and consequently in $S_k$. Thus, changes are detected at the points in time where there
is a sharp increase in the $S_k$ curve. Replacing Eq. (4.2) into Eq. (4.3), we obtain the log-likelihood ratio:

$$T_k = \frac{1}{2} \ln \left( \frac{|C_0|}{|C_1|} \right) + \frac{1}{2} (\bar{h}_k - \bar{\mu}_0)^T C_0^{-1} (\bar{h}_k - \bar{\mu}_0)$$

$$- \frac{1}{2} (\bar{h}_k - \bar{\mu}_1)^T C_1^{-1} (\bar{h}_k - \bar{\mu}_1)$$

(4.4)

Considering that our covariance matrix is diagonal, its inverse form is given by:

$$C_l^{-1} = \begin{pmatrix}
\frac{1}{\sigma_{l,1}^2} & 0 & \cdots & 0 \\
0 & \frac{1}{\sigma_{l,2}^2} & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & \frac{1}{\sigma_{l,W_0}^2}
\end{pmatrix}, \quad l = \{0, 1\}$$

and its determinant $|C_l| = \prod_{n=1}^{W_0} \sigma_{l,n}^2$, $l = \{0, 1\}$. It should be noted that all diagonal elements of the covariance matrix are non-zero, since they correspond to the autocorrelation of the HOFs, so the covariance is indeed invertible. Under the i.i.d. data assumption, all values of the variance are $\sigma_{l,n}^2 = \sigma^2$, $l = \{0, 1\}$, $n = [1, 2, ..., W_0]$, as explained in the previous section.

By plugging in $T_k$ calculated at each frame, we get a value for the test statistic $S_k$ which significantly increases when there is a change in our data, i.e. the motion features. In order to detect this increase, we compare the test statistic $S_k$ at each frame with a threshold derived from its previous values. In the statistical literature, this threshold cannot be determined in a theoretical manner. The widely accepted solution is to experimentally find a threshold that leads to few false alarms [102]. We automatically estimate the threshold at each time instant $k$ based on the recent test statistics values:

$$\eta_k = \text{mean}[S_{k-1}] + c \cdot \text{std}[S_{k-1}]$$

where $\text{mean}[...]$ is the mean of the test statistic’s values until frame $k - 1$, $\text{std}[...]$ is their standard deviation and $c$ is the threshold parameter, empirically selected to equal 2.5.

We decided to use this value after cross validation performed on numerous characteristic ADL videos for a set of eight different numbers in the interval $[0.001, 10]$. When the test statistic $S_k$ surpasses this threshold $\eta_k$, a change in motion is detected, signifying the end of that trajectory.

This procedure leads to the temporal segmentation of the extracted trajectories based on actual changes in the activities taking place, rather than by using a manually selected constant threshold as in the literature. Introducing this non ad-hoc approach to segmenting trajectories indeed improves the recognition results of our method, as can be
seen in the experiments of Sec. 4.4. In Fig. 4.8 we can see several examples of optimal trajectories in several activities taken from URADL and KTH dataset. Green lines depict the extension size that optimal trajectories introduce to a fixed size trajectory of 15 frames. We can see that trajectories are terminated in a more meaningful way when a change in the motion pattern occurs or when the activity stops.

4.3 ADL recognition schemes

In Sec. 4.2.2 we described how ADLs are represented by appearance and motion characteristics both in space and time, resulting in a set of spatio-temporal descriptors. For ADL recognition, we follow a BoVW and a Fisher representation in conjunction with a multiclass Support Vector Machine (SVM), due to its successful use in the SoA, for predicting ADLs within video segments.

4.3.1 Bag-of-Visual-Words

In a typical BoVW pipeline, depicted with blue arrows in Fig. 4.9, a representative visual vocabulary is first built by applying a $K$-Means clustering algorithm to the spatio-temporal descriptors extracted from the training videos. The resulting $K$ cluster centers form a visual vocabulary based on which sparse frequency histograms, using a hard binning approach, are built for both training and test video segments (i.e. the nearest cluster center to the studied feature vector increased by 1).
Training frequency histograms are then used to compute a symmetric Chi-Square Kernel among training data, as described in Sec. 3.3.1 and create a multi-class SVM. Test frequency histograms are also encoded using a Chi-Square kernel among them and the training data and fed to the precomputed SVM in order to recognize ADLs in test video segments.

**4.3.2 Fisher encoding**

The fisher encoding pipeline, depicted with red arrows in Fig. 4.9, is the second adopted encoding scheme, which uses a GMM clustering procedure in order to extract the desirable visual vocabulary. Usually an EM (i.e. Expectation Maximization) algorithm is used to acquire the $m$ most distinguishable and representative Gaussians of the training feature space which are then used to calculate Fisher differences for the training and test data and accumulate them into a fixed size Fisher vector, as analyzed in Sec. 3.3.3.

Fisher vectors extracted from training data are then transformed using a Hellinger Kernel and used to learn a multi-class Linear SVM. Test Fisher vectors are also transformed using a Hellinger kernel and then used in conjunction with the precalculated SVM in order to predict the ADLs that occur within test video segments.

We implemented and compared two clustering and quantization techniques to determine the most appropriate one for action recognition. In this section, we describe how clustering and quantization are performed, while Fig. 4.9 shows an overview of the recognition process.
4.4 Experiments on ADL benchmark datasets

In this section we present the ADL datasets in Sec. 4.4.1, experimental settings and parameters in Sec. 4.4.2, and experimental results in Sec. 4.4.3.

4.4.1 Action datasets

Experiments with publicly available ADL datasets took place to objectively determine the applicability and robustness of our algorithm and compare it with the SoA. Popular benchmark ADL datasets used include URADL [11] and the KIT robo-kitchen data-set [16], which contain activities performed by human actors in a home and kitchen scenario, while experiments also took place with the KTH videos, which are not strictly limited to ADLs in a home environment, as people perform various activities like running, boxing, clapping, both indoors and outdoors.

Our detailed examination of benchmark ADL videos revealed some of their limitations: current ADL datasets have relatively limited anthropometric variance, as activities are mostly carried out by a few actors in controlled environments. In order to also test our method on videos that are realistic and contain similar activities to what happens in people’s daily life, we used the ADL datasets recorded for the EU project Dem@Care (www.demcare.eu) at the Greek Association for Alzheimer’s Disease and Related Disorders (GAADRD), detailed in Sec. 2.3.6 and Sec. 2.3.7.

4.4.1.1 KTH action dataset

One of the most popular datasets for human activity recognition is the KTH dataset [14]. The KTH videos consist of 2391 sequences, recorded in four different environments: outdoors $s_1$, outdoors with scale variation $s_2$, outdoors with different clothing $s_3$, and indoors $s_4$. Six different actions are performed by actors: box, handclap, handwave, jog, run, walk, while 25 subjects carried out these actions, for increased anthropometric variance. Sample frames from KTH with their characteristic rectangles and interest point trajectories are depicted in Fig. 4.10. The dataset was split into 16 training and 9 testing videos, as suggested in [14], based on the subjects that perform the actions, in order to evaluate our algorithm. Further information about this dataset are provided in Sec. 2.3.1 and experimental results can be found in sec. 4.4.3.1.
Chapter 4. Optimal trajectories and MBAA for ADL recognition

4.4.1.2 URADL action dataset

The University of Rochester Activities of Daily Living (URADL) dataset was examined in detail, due to its inclusion of ADLs which are relevant in many practical applications. In the URADL videos, 5 different actors performed 10 different activities, 3 times each, in a kitchen environment, resulting in 150 videos. High resolution video frames were captured from a static RGB camera placed across the actors. The short duration of these videos rendered the dataset more appropriate for evaluating recognition of ADLs approaches, rather than for determining algorithmic usefulness in real life scenarios.

A disadvantage of the URADL dataset is that it lacks significant anthropometric variations, as most actors have a similar appearance and perform the activities in the same manner. Additionally, the URADL videos are characterized by considerable environmental constraints: all actions take place in the same location, behind a kitchen counter, while the subjects do not move significantly. The actors are clearly visible and there are very few occlusions, as seen in sample frames with their trajectories and ADL descriptors in Fig. 4.11. Nevertheless, URADL is broadly used in the literature [11], its small size making it useful for the fast evaluation of algorithms.

In order to evaluate our algorithm, we use leave-one-subject-out testing and represent the activities by: AP = Answer Phone, CB = Chop Banana, ES = Eat Snack, DP = Dial Phone, DW = Drink Water, EB = Eat Banana, LiP = Look up in Phonebook, PB = Peel Banana, US = Use Silverware, WoW = Write on Whiteboard. Further details about the dataset can be found in Sec. 2.3.4, while experimental results are given in sec. 4.4.3.2.

4.4.1.3 KiT Robo-kitchen action dataset

A second, more challenging ADL dataset is the KIT Robo-Kitchen activity dataset, publicly provided by the Karlsruhe Institute of Technology. In these videos, cameras were installed in a stereo setup in two different kitchen environments, where 17 human actors were called to perform 14 different activities. The first experimental setup for KIT, named “Counter Top”, concerns kitchen activities around a kitchen counter and sink that are recorded by three cameras, while the second set of recordings, named “Room Setup”, features a different set of kitchen related activities, recorded by two cameras.

We examine the activities of each setup separately (since they contain different activities), but use the data from all camera views for each set of experiments. This allows us to use a larger number of training and testing videos, and also demonstrate the viewpoint invariance of our method. The number of actors, their different appearance and large
number of activities performed in various manners led to considerable anthropometric variance. The KIT kitchen environments are more realistic than those in URADL, with furniture and kitchen equipment placed in different parts of the room, necessitating more movement around the room to perform each activity, which results in more realistic ADLs than in URADL. The reader can see Sec. 2.3.5 for more dataset details, while experimental results can be found in sec. 4.4.3.3.

KIT-Counter Top Setup: In the Counter Top setup, kitchen activities are performed on a kitchen counter around the sink, as in the top row of Fig. 4.12, recorded by three cameras. We accumulate all videos from the three cameras and use them in a leave-one-subject-out train/test split. The resulting high recognition rates demonstrate that this proposed method has strong viewpoint invariance. Activities are represented as: Cut = cut vegetables, Dry = dry the washed dishes, Fry = fry vegetables, Peel = peel vegetables, Stir = stir a cooking soup, Wash = wash dishes, Wipe = wipe the countertop.

KIT-Room Setup: In the Room Setup, two cameras record the activities: Cl: clear the table, Cf: drink coffee and read a newspaper, Ct: cut vegetables, ED: empty the dishwasher, Pl: peel vegetables, Pz: eat pizza, ST: set the table, Sp: eat soup, Sep: sweep the floor, Wp: wipe table (bottom row of Fig. 4.12). As for the Counter Top scenario, videos from both cameras are accumulated in a leave-one-subject-out train/test split.

4.4.1.4 Dem@Care Action Dataset

In addition to using public ADL benchmark datasets, we proceeded with the recording of ADLs in a realistic home-like environment, at the Greek Association of Alzheimer’s Disease and Related Disorders (GAADRD) in Thessaloniki [17]. The subjects were aged over 65, with conditions ranging from mild cognitive impairment (MCI) to dementia and Alzheimer’s, while an almost equal number of healthy individuals in the same age group was also recorded performing the same ADLs. The participants were of both genders and the activities they performed required moving around the room, similarly to real life. These factors, as well as the size of the population being recorded (32 participants) introduced great anthropometric variations in our ADL videos and made them more realistic than current benchmark data. The ADLs performed are represented as CU: clean up table, DB: drink beverage, EP: end phonecall, ER: enter room, ES: eat snack, HS: handshake, PS: prepare snack, RP: read paper on the couch, SB: serve beverage, SP: start phonecall, TV: talk to visitor. Fig. 4.13 shows sample frames from the Dem@Care dataset with their trajectories and HOGHOF rectangles. Further details about the dataset are provided in sec. 2.3.6, while results are aggregated in sec. 4.4.4.
4.4.1.5 CHUN Action Dataset

Another realistic ADL dataset was recorded at the Nice University Hospital (CHUN) for evaluating the applicability of our algorithm. The CHUN dataset consists of 15 hr and 10 min recordings of 64 PwD that perform ADLs in a Lab environment. The camera viewpoint monitors the whole room where the patient is and performs semi-directed ADL’s (i.e. written on a paper). The ADLs observed include: (AP) answering phone and (DP) dialing phone, (LoM) look on map, (PB) pay bill, (PD) prepare drugs, (PT) prepare tea, (RP) read paper, (WP) water plant and (WtV) watch TV. They included large anthropometric variations and activity performance styles, while severe occlusions introduced great difficulty in discriminating actions. Figure 4.14 depicts some activities with their trajectory and HOGHOF rectangles. Further details for the dataset are provided in sec. 2.3.7, while experimental results can be found in sec. 4.4.5.

4.4.2 Experimental parameters

The experiments presented here were carried out using an Intel core i5 – 3570 CPU 3.4GHz, 8GB RAM personal computer with a x64-bit Windows operating system. Two different OF algorithms were used to evaluate their potential to reduce computational cost and the accuracy they achieve. The first\(^\text{[19]}\) is a computationally efficient technique for extracting dense OF maps (0.2027 sec per 640 × 480 pixel frame) and was used for evaluating the parameters of our algorithm and its robustness. The second method\(^\text{[18]}\) is more precise, but has a higher computational cost (0.8021 sec per 640 × 480 pixel frame).

In this work, the computational cost is measured in terms of frames per second (fps) processing time for extracting the descriptors: a higher fps corresponds to fast processing speeds and improved computational cost. We measure the cost in terms of fps to allow for comparisons between videos of different durations. Our measurements of computational burden focus on the cost of descriptor extraction, as this is the most “costly” step of the methods examined, so the same fps are observed for the $\chi^2$ BoVW and Fisher-based encoding schemes in the tables that follow. In our experiments using K-means, we used $K = 4000$ cluster centers to partition our feature vector space, as in \cite{21, 12}, where cross validation took place to define the best vocabulary size for activity recognition. We also use this size to enable comparisons with the SoA, which also uses the same vocabulary size.

To limit the method’s complexity, cluster centers were derived from a randomly selected subset of 100,000 feature vectors from the training set. To further ensure the most discriminative cluster centers are used, K-means was initialized 10 times. We used 256
cluster centers for GMM based clustering to compute Fisher vector distances and applied PCA \cite{46} to reduce vocabulary dimensionality to 80, improve accuracy and decrease the memory footprint of our representation.

We tested 3 different kinds of action descriptors to demonstrate the advantages and disadvantages of each algorithmic step. As baseline, we implemented the local HOGHOF descriptor. We then applied SSCD (statistical sequential change detection), analyzed in Sec. 4.2.3, to produce temporally optimal trajectories, resulting in the HOGHOF\_SSCD descriptor. We also added trajectory coordinates to include global attributes in our descriptor, as analyzed in Sec. 4.2.2.1, depicted as HOGHOF\_TR. We constructed local descriptors with 9 bin histograms for the HOGs and HOFs at four spatial scales as in the SoA \cite{21}. Sampling took place with a grid step size of $k = 8, 16, 24, 32$ at each scale. Finally, we subdivided each HOGHOF action descriptor into a $n_x = n_y = 2, n_t = 3$ grid of cuboids for higher spatial resolution. All action descriptors were used in both K-Means and GMM recognition schemas, to compare the effect of our BoVW representation on the recognition accuracy.

\subsection*{4.4.3 Experimental results}

In this section we present the experimental results using the activity descriptors HOGHOF, HOGHOF\_TR and HOGHOF\_SSCD on five different activity datasets, namely KTH, URADL, KIT and DemCare.

\subsubsection*{4.4.3.1 KTH results}

Table 4.2 aggregates the accuracy rates with confidence intervals, computational times and comparisons with the SoA for all KTH classes. On average, the HOGHOF\_SSCD activity descriptor leads to the best results when dense OF \cite{18} is used. It should be noted HOGHOF\_SSCD outperforms the other descriptors for long duration activities, such as handclap, handwave, jog, walk, as they contain long trajectories, which are optimally segmented in time by SSCD, unlike short and fast duration activities like
Table 4.2: KTH ADL recognition accuracy with confidence intervals (CI) and processing speed in frames per second (fps) for all activities and different combinations of appearance (HOG), motion (HOF) and trajectory (traj, SSCD) descriptors, using the Optical Flow (OF) methods of [18] and [19]. Comparison with the SoA methods [20], [12], [21], [22], where applicable.

<table>
<thead>
<tr>
<th></th>
<th>χ² BoVW</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>HOGHOF</td>
<td>HOGHOF_TR</td>
<td>HOGHOF_SSCD</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[19]</td>
<td>[18]</td>
<td>[19]</td>
<td>[18]</td>
</tr>
<tr>
<td>Box</td>
<td>95.8%</td>
<td>97.2%</td>
<td><strong>98.6%</strong></td>
<td>95.1%</td>
</tr>
<tr>
<td>Clap</td>
<td>93.1%</td>
<td>91.6%</td>
<td>91.0%</td>
<td>81.9%</td>
</tr>
<tr>
<td>Wave</td>
<td>82.6%</td>
<td>84.0%</td>
<td>82.6%</td>
<td>77.7%</td>
</tr>
<tr>
<td>Jog</td>
<td>80.6%</td>
<td>83.3%</td>
<td>79.2%</td>
<td>81.2%</td>
</tr>
<tr>
<td>Run</td>
<td>73.6%</td>
<td>74.3%</td>
<td><strong>79.9%</strong></td>
<td>70.8%</td>
</tr>
<tr>
<td>Walk</td>
<td><strong>98.6%</strong></td>
<td>95.1%</td>
<td>91.6%</td>
<td>93.1%</td>
</tr>
<tr>
<td>AvAcc</td>
<td>87.4%</td>
<td>87.6%</td>
<td>87.2%</td>
<td>83.3%</td>
</tr>
<tr>
<td>CI (±)</td>
<td>7.91%</td>
<td>6.92%</td>
<td>6.23%</td>
<td>7.40%</td>
</tr>
<tr>
<td>fps</td>
<td>11.98</td>
<td>3.90</td>
<td>11.98</td>
<td>3.90</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Fisher</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>HOGHOF</td>
<td>HOGHOF_TR</td>
<td>HOGHOF_SSCD</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[19]</td>
<td>[18]</td>
<td>[19]</td>
<td>[18]</td>
</tr>
<tr>
<td>Box</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>Clap</td>
<td>95.1%</td>
<td>96.5%</td>
<td>97.2%</td>
<td>97.9%</td>
</tr>
<tr>
<td>Wave</td>
<td>84.7%</td>
<td>87.5%</td>
<td><strong>89.6%</strong></td>
<td>86.1%</td>
</tr>
<tr>
<td>Jog</td>
<td>91.7%</td>
<td>92.3%</td>
<td>89.6%</td>
<td><strong>95.8%</strong></td>
</tr>
<tr>
<td>Run</td>
<td>76.4%</td>
<td>83.3%</td>
<td>81.9%</td>
<td><strong>84.0%</strong></td>
</tr>
<tr>
<td>Walk</td>
<td>100%</td>
<td>98.6%</td>
<td>100%</td>
<td>98.6%</td>
</tr>
<tr>
<td>AvAcc</td>
<td>91.3%</td>
<td>93.1%</td>
<td>93.1%</td>
<td>93.8%</td>
</tr>
<tr>
<td>CI (±)</td>
<td>7.44%</td>
<td>5.27%</td>
<td>5.79%</td>
<td>5.51%</td>
</tr>
<tr>
<td>fps</td>
<td>11.98</td>
<td>3.90</td>
<td>11.98</td>
<td>3.90</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Related Work</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ours</td>
<td>[20]</td>
<td>[12]</td>
<td>[21]</td>
</tr>
</tbody>
</table>
| Box    | 100% | - | 97% | - | 100%
| Clap   | **97.9%** | - | 95% | - | 94%
| Wave   | 86.1% | - | 91% | - | **99%**
| Jog    | **95.8%** | - | 89% | - | 91%
| Run    | 84.0% | - | 80% | - | **89%**
| Walk   | 98.6% | - | 99% | - | 94%
| AvAcc  | 93.8% | 91.4% | 91.8% | 94.2% | 94.5%
| CI (±) | 5.51% | 5.50% | - | 3.46% | - |
box and run. HOGHOF_TR outperforms the other descriptors when combined with GMM-Fisher clustering and encoding, as their probabilistic nature successfully integrates the trajectory information, leading to a more complete descriptor. Overall, our method leads to results that are comparable with the SoA, or better than it.

4.4.3.2 URADL results

Table 4.3 aggregates all recognition rates, confidence intervals and processing speeds (in fps) for the activities that recorded on the URADL dataset. For $\chi^2$ BoVW, the simple HOGHOF descriptor with the OF of $[19]$ achieves the best accuracy, which can be attributed to the uncomplicated nature of the URADL videos.

For both encoding schemes and the HOGHOF descriptor, we provide recognition rates when interest points are extracted from Activity Areas(AA) which were analyzed in previous chapter (Sec. 3.2.2) instead of MBAAs. In column $[18]_{AA}$ it can be seen that AAs indeed lead to lower accuracy and a higher computational cost, as expected giving a worse result than MBAAs.

For GMM-based vocabulary construction with Fisher encoding, both HOGHOF_TR and HOGHOF_SSCD with dense OF $[18]$ achieve an average accuracy of 94%, surpassing all methods, including the SoA, due to the inclusion of accurate trajectory information and its meaningful temporal segmentation respectively, combined with probabilistic encoding. In all cases, the highest processing speed (fps) is achieved with HOGHOF_SSCD and the OF of $[19]$, which intuitively makes sense since SSCD leads to temporally optimal trajectories and the computationally efficient OF of $[19]$ is used.
Table 4.3: URADL recognition accuracy with confidence intervals (CI) and processing speed in frames per second (fps) for all activities and combinations of appearance (HOG), motion (HOF) and trajectory (traj, SSCD) descriptors, using the OF of [18] and [19]. All experiments are on MBAA data, except the columns [18]AA. Comparison with the SoA methods Messing:URADL, [21] with HOGHOF, [21] with MBH.

<table>
<thead>
<tr>
<th></th>
<th>HOGHOF</th>
<th>HOGHOF_TR</th>
<th>HOGHOF_SSCD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>[19]</td>
<td>[18]</td>
<td>[18]AA</td>
</tr>
<tr>
<td>AP</td>
<td>46.7%</td>
<td><strong>60.0%</strong></td>
<td>46.7%</td>
</tr>
<tr>
<td></td>
<td>46.7%</td>
<td>40.0%</td>
<td>33.3%</td>
</tr>
<tr>
<td>CB</td>
<td>93.3%</td>
<td>93.3%</td>
<td>93.3%</td>
</tr>
<tr>
<td></td>
<td>100%</td>
<td>93.3%</td>
<td>100%</td>
</tr>
<tr>
<td>DP</td>
<td>93.3%</td>
<td>73.3%</td>
<td>80.0%</td>
</tr>
<tr>
<td>DW</td>
<td><strong>100%</strong></td>
<td>66.7%</td>
<td>80.0%</td>
</tr>
<tr>
<td></td>
<td>80.0%</td>
<td>80.0%</td>
<td>93.3%</td>
</tr>
<tr>
<td></td>
<td>93.3%</td>
<td>80.0%</td>
<td>80.0%</td>
</tr>
<tr>
<td>EB</td>
<td>80.0%</td>
<td>86.7%</td>
<td>80.0%</td>
</tr>
<tr>
<td></td>
<td>100%</td>
<td>93.3%</td>
<td>80.0%</td>
</tr>
<tr>
<td>ES</td>
<td>80.0%</td>
<td>100%</td>
<td>73.3%</td>
</tr>
<tr>
<td>LiP</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>PB</td>
<td>100%</td>
<td>50.0%</td>
<td>66.7%</td>
</tr>
<tr>
<td>US</td>
<td>100%</td>
<td>80.0%</td>
<td>80.0%</td>
</tr>
<tr>
<td>WoW</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>AvAcc</td>
<td>89.3%</td>
<td>84.7%</td>
<td>76.0%</td>
</tr>
<tr>
<td>Ci (±)</td>
<td>10.53%</td>
<td>8.94%</td>
<td>13.0%</td>
</tr>
<tr>
<td>fps</td>
<td>2.51</td>
<td>0.93</td>
<td>0.75</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Fisher</th>
<th>HOGHOF</th>
<th>HOGHOF_TR</th>
<th>HOGHOF_SSCD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>[19]</td>
<td>[18]</td>
<td>[18]AA</td>
<td></td>
</tr>
<tr>
<td>AP</td>
<td>46.7%</td>
<td>53.3%</td>
<td><strong>60.0%</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>66.7%</td>
<td>46.7%</td>
<td>66.7%</td>
<td></td>
</tr>
<tr>
<td>CB</td>
<td>93.3%</td>
<td>93.3%</td>
<td>80.0%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>100%</td>
<td>100%</td>
<td>93.3%</td>
<td></td>
</tr>
<tr>
<td>DP</td>
<td>93.3%</td>
<td>73.3%</td>
<td>66.7%</td>
<td></td>
</tr>
<tr>
<td>DW</td>
<td>100%</td>
<td>93.3%</td>
<td>86.7%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>80.0%</td>
<td>80.0%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>80.0%</td>
<td>80.0%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>EB</td>
<td>80.0%</td>
<td>93.3%</td>
<td>80.0%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>ES</td>
<td>80.0%</td>
<td>100%</td>
<td>86.7%</td>
<td></td>
</tr>
<tr>
<td>LiP</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>PB</td>
<td>100%</td>
<td>50.0%</td>
<td>66.7%</td>
<td></td>
</tr>
<tr>
<td>US</td>
<td>100%</td>
<td>100%</td>
<td>80.0%</td>
<td></td>
</tr>
<tr>
<td>WoW</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>AvAcc</td>
<td>90.7%</td>
<td>92.0%</td>
<td><strong>94.0%</strong></td>
<td></td>
</tr>
<tr>
<td>Ci (±)</td>
<td>10.34%</td>
<td>8.67%</td>
<td>13.0%</td>
<td></td>
</tr>
<tr>
<td>fps</td>
<td>2.51</td>
<td>0.93</td>
<td>0.75</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Related Work</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP</td>
<td>66.7%</td>
</tr>
<tr>
<td>CB</td>
<td>93.3%</td>
</tr>
<tr>
<td>DP</td>
<td>86.7%</td>
</tr>
<tr>
<td>DW</td>
<td>93.3%</td>
</tr>
<tr>
<td>EB</td>
<td>100%</td>
</tr>
<tr>
<td>ES</td>
<td>100%</td>
</tr>
<tr>
<td>LiP</td>
<td>100%</td>
</tr>
<tr>
<td>PB</td>
<td>100%</td>
</tr>
<tr>
<td>US</td>
<td>100%</td>
</tr>
<tr>
<td>WoW</td>
<td>100%</td>
</tr>
<tr>
<td>AvAcc</td>
<td>94.0%</td>
</tr>
<tr>
<td>Ci (±)</td>
<td>6.59%</td>
</tr>
<tr>
<td>fps</td>
<td>0.83</td>
</tr>
</tbody>
</table>
Some of the activities examined are generally very difficult to recognize: for example, Answer Phone (AP) is usually confused with Dial Phone (DP), while other activities like Write On Whiteboard (WoW) or Look in Phonebook (LiP) are recognized with almost 100% accuracy.

4.4.3.3 KiT results

For the KiT dataset we compared HOGHOF,SSCD to HOGHOF in terms of accuracy with confidence intervals for different encoding solutions. Table 4.4 shows that for the “Counter Top” KiT scenario both encoding schemes $\chi^2$ and Fisher, the HOGHOF,SSCD descriptor led to the best recognition rates, surpassing the SoA [16] by 0.3% and 7.1% respectively. For the “Room Setup” case, HOGHOF led to the best results for $\chi^2$ encoding, and HOGHOF,SSCD gave better accuracy for Fisher encoding, while both our methods significantly surpassed the SoA [16] by 9.7% and 12.7% respectively. Individual activity accuracies are also presented in Tables 4.4 and 4.5, where it can be seen that most activities are recognized with very high accuracy, often reaching 100%. As for the previous datasets, HOGHOF,SSCD led to better results for activities of longer duration, as the application of SSCD is most meaningful for long trajectories.

4.4.4 Dem@Care results

In the experimental setup for the DemCare action dataset, we follow a Leave-One-Subject-Out technique and carried out experiments with both OF methods of [19], [18], the HOGHOF, HOGHOF,TR and HOGHOF,SSCD descriptors to estimate their accuracy and compare them with the SoA of [21] with HOGHOF and [21] with MBH.
Table 4.4: KIT Counter top setup activity recognition accuracy with confidence intervals and comparison with the SoA [16].

<table>
<thead>
<tr>
<th>Activity</th>
<th>χ² BoVW</th>
<th>Fisher</th>
<th>Rel. Work</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cut</td>
<td>HOGHOF</td>
<td>66.7%</td>
<td>71.4%</td>
</tr>
<tr>
<td>Dry</td>
<td>HOGHOF</td>
<td>92.9%</td>
<td>100%</td>
</tr>
<tr>
<td>Fry</td>
<td>HOGHOF</td>
<td>90.5%</td>
<td>90.5%</td>
</tr>
<tr>
<td>Peel</td>
<td>HOGHOF</td>
<td>80.9%</td>
<td>90.5%</td>
</tr>
<tr>
<td>Stir</td>
<td>HOGHOF</td>
<td>85.7%</td>
<td>85.7%</td>
</tr>
<tr>
<td>Wash</td>
<td>HOGHOF</td>
<td>85.7%</td>
<td>85.7%</td>
</tr>
<tr>
<td>Wipe</td>
<td>HOGHOF</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>AvAcc</td>
<td>HOGHOF</td>
<td>86.1%</td>
<td>89.1%</td>
</tr>
<tr>
<td>CI ±</td>
<td>HOGHOF</td>
<td>7.78%</td>
<td>7.26%</td>
</tr>
</tbody>
</table>

Table 4.5: KIT Room Setup activity recognition accuracy with confidence intervals and comparison with the SoA [16].

<table>
<thead>
<tr>
<th>Activity</th>
<th>χ² BoVW</th>
<th>Fisher</th>
<th>Rel. Work</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClrTb</td>
<td>HOGHOF</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>Coffe</td>
<td>HOGHOF</td>
<td>100%</td>
<td>72.2%</td>
</tr>
<tr>
<td>Cut</td>
<td>HOGHOF</td>
<td>83.3%</td>
<td>77.8%</td>
</tr>
<tr>
<td>EmpDi</td>
<td>HOGHOF</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>Peel</td>
<td>HOGHOF</td>
<td>88.2%</td>
<td>88.2%</td>
</tr>
<tr>
<td>Pizza</td>
<td>HOGHOF</td>
<td>78.6%</td>
<td>78.6%</td>
</tr>
<tr>
<td>SetTb</td>
<td>HOGHOF</td>
<td>92.8%</td>
<td>92.8%</td>
</tr>
<tr>
<td>Soup</td>
<td>HOGHOF</td>
<td>92.8%</td>
<td>85.7%</td>
</tr>
<tr>
<td>Sweep</td>
<td>HOGHOF</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>Wipe</td>
<td>HOGHOF</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>AvAcc</td>
<td>HOGHOF</td>
<td>93.6%</td>
<td>91.6%</td>
</tr>
<tr>
<td>CI ±</td>
<td>HOGHOF</td>
<td>4.91%</td>
<td>5.44%</td>
</tr>
</tbody>
</table>

The use of HOGHOF_SSCD with the OF from [19] led to the best recognition rates for the BoVW recognition schema, performing better than other action descriptors. This can be attributed to the fact that SSCD led to meaningful sub-trajectories for these activities, compensating for the hard binning of K-means and giving more discriminative descriptors which led to high recognition accuracy. When Fisher was used for recognition, the role of SSCD was less important than the role of trajectory information, as seen on the top row of Table 4.6. This is because of the larger movements around the room, which were well described by HOGHOF_TR and adequately encoded by GMM & Fisher.

HOGHOF_TR outperforms HOGHOF and HOGHOF_SSCD when using Fisher encoding and 2D_CLG OF from [19], shown in the middle of Table 4.6. It is worthwhile to note that the inclusion of global trajectory coordinates in HOGHOF_TR increase the recognition rates by a range from 1.901% to 7.293% when combined with Fisher compared
Chapter 4. *Optimal trajectories and MBAA for ADL recognition*

Figure 4.13: Optimal trajectories (red lines) & hybrid descriptors (green rectangles) for DemCare action dataset.

Table 4.6 shows that for the Leave-One-Subject-Out split, the SoA of [21] and the MBH led to the highest recognition rates. This can be explained by the fact that the SoA methods process the entire video frame, consequently including more training data and more information about the activities taking place, making their discrimination easier. Nevertheless, in practice it is best to discriminate with as little data as possible, aiming at the optimal tradeoff between accuracy and computational cost (i.e., as in HOGHOF,SSCD). Additionally, it should be noted that in all cases our methods led to results comparable to the SoA, despite using a small part of each video frame.

For this dataset, the OF of [18] does not provide the best results, as the DemCare videos contain noisy luminance values that introduce erroneous dense OF estimates. Table 4.6 shows that some activities with large motion can be easily distinguished from others such as Enter Room (ER), while others such as Talk to Visitor (TV) and Read Paper (RP) can be easily recognized thanks to the inclusion of their trajectory and location information in the descriptor (HOGHOF,TR). On the other hand, activities with small motion, or activities that are similar to each other, such as Drink Beverage(DB)/Eat Snack(ES) and Serve Beverage(SB)/Prepare Snack(PS), are easily confused, reducing the total recognition rate.
4.4.5 CHUN results

For this experiment, we followed a Leave-One-Subject-Out technique. Table 4.7 aggregates BoVW and Fisher results. We used only 2D_CLG OF [19] due to its low computational cost and test BoVW against Fisher recognition rates. It’s obvious that HOGHOF_TR improves recognition results, while Fisher performs better than BoVW in most cases. HOGHOF_SSCD gets better recognition rates when combined with BoVW, contrary to Fisher encoding scheme that benefits from trajectories with fixed length. We surpass both HOGHOF [21] and MBH [21] which proves our superiority on real case ADL datasets.

4.5 Discussion

In this chapter, we present a new approach for ADL recognition which achieves high recognition rates, comparable and often surpassing the best results currently available. Our method is inspired by the SoA [21], where HOGHOF descriptors are extracted around densely sampled points and their trajectories to adequately characterize a scene and the actions taking place in it.
In order to deal with the high computational cost that is produced by dense interest point extraction, we introduce dense sampling in regions of varying motion, the MBAAs, which are extracted via higher order statistical processing of accurate flow data. This succeeds not only in reducing the computational cost of the activity representation, but also in producing more discriminative interest points and better recognition rates.

Departing from the current literature, we introduce an enhanced KLT tracker, which deploys a RANSAC homography estimator to eliminate outlier correspondences and form accurate interest point trajectories, so as to achieve better recognition rates. The tracked interest points are characterized by local HOGHOF descriptors on multiple scales to ensure scale invariance, while spatial coordinates of the trajectory points are added to the resulting descriptor in order to supplement it with global information. The resulting hybrid descriptor (i.e. HOGHOF_TR) combines both local and global information, leveraging upon their respective advantages. Indeed, experimental results with benchmark ADL datasets and our own Dem@Care ADL videos verify that when this representation is combined with the Fisher encoding scheme, we obtain recognition accuracy that is comparable or better than the SoA.

We also introduce a new method for optimal temporal segmentation of the extracted trajectories, in order to accurately localize subactivities in time. Statistical sequential change detection (SSCD) is applied on the HOFs of the tracked interest points, in order to determine when the motion changes and, therefore, a new subactivity is taking place. The application of SSCD produced different ADL descriptor (i.e. HOGHOF_SSCD) which achieved the highest recognition rates in almost all experiments with benchmark ADL data when combined with the BoVW encoding scheme.

Detailed experiments took place with various combinations of our proposed representation and recognition schemes, as well as comparisons with the SoA supporting the above-mentioned contributions.

Having introduced two activity recognition algorithms applied both in constrained and unconstrained scenarios, in the Chapter 5 we analyze an activity detection technique allowing us process unsegmented videos automatically. For that purpose, a novel statistical sequential technique is applied on the motion patterns and a sliding window is then used in order to localize activity boundaries in videos within unknown ADLs and recognize them.
Table 4.6: DemCare recognition accuracy rates over all activity classes for: different combinations of appearance (HOG), motion (HOF), trajectory (traj, SSCD), the Optical Flow (OF) methods of [19] and [18] and comparisons with the SoA Wang:DenseTraj with HOGHOF and [21] with MBH.

<table>
<thead>
<tr>
<th>χ² BoVW</th>
<th>HOGHOF</th>
<th>HOGHOF_TR</th>
<th>HOGHOF_SSCD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>[19]</td>
<td>[18]</td>
<td>[19]</td>
</tr>
<tr>
<td>CU</td>
<td>86.7%</td>
<td>73.0%</td>
<td>83.2%</td>
</tr>
<tr>
<td>DB</td>
<td>89.7%</td>
<td>80.0%</td>
<td>78.3%</td>
</tr>
<tr>
<td>EP</td>
<td>93.7%</td>
<td>84.4%</td>
<td>82.8%</td>
</tr>
<tr>
<td>ER</td>
<td>98.4%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>ES</td>
<td>75.2%</td>
<td>71.3%</td>
<td>68.3%</td>
</tr>
<tr>
<td>HS</td>
<td>90.6%</td>
<td>93.7%</td>
<td>75.0%</td>
</tr>
<tr>
<td>PS</td>
<td>69.1%</td>
<td>60.6%</td>
<td>77.7%</td>
</tr>
<tr>
<td>RP</td>
<td>87.5%</td>
<td>87.5%</td>
<td>93.7%</td>
</tr>
<tr>
<td>SB</td>
<td>85.3%</td>
<td>94.1%</td>
<td>88.2%</td>
</tr>
<tr>
<td>SP</td>
<td>81.8%</td>
<td>84.8%</td>
<td>87.8%</td>
</tr>
<tr>
<td>TV</td>
<td>87.1%</td>
<td>87.5%</td>
<td>87.1%</td>
</tr>
<tr>
<td>AvAcc</td>
<td>85.9%</td>
<td>83.6%</td>
<td>83.9%</td>
</tr>
<tr>
<td>Conf±</td>
<td>4.84%</td>
<td>6.79%</td>
<td>5.26%</td>
</tr>
<tr>
<td>Time(fps)</td>
<td>8.03</td>
<td>3.92</td>
<td>8.31</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fisher</th>
<th>HOGHOF</th>
<th>HOGHOF_TR</th>
<th>HOGHOF_SSCD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>[19]</td>
<td>[18]</td>
<td>[19]</td>
</tr>
<tr>
<td>CU</td>
<td>91.2%</td>
<td>82.3%</td>
<td>83.8%</td>
</tr>
<tr>
<td>DB</td>
<td>97.2%</td>
<td>93.6%</td>
<td>98.8%</td>
</tr>
<tr>
<td>EP</td>
<td>79.7%</td>
<td>81.2%</td>
<td>89.1%</td>
</tr>
<tr>
<td>ER</td>
<td>99.2%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>ES</td>
<td>80.7%</td>
<td>68.1%</td>
<td>86.5%</td>
</tr>
<tr>
<td>HS</td>
<td>93.7%</td>
<td>90.6%</td>
<td>93.7%</td>
</tr>
<tr>
<td>PS</td>
<td>77.7%</td>
<td>62.8%</td>
<td>83.4%</td>
</tr>
<tr>
<td>RP</td>
<td>90.6%</td>
<td>93.7%</td>
<td>96.8%</td>
</tr>
<tr>
<td>SB</td>
<td>83.8%</td>
<td>76.5%</td>
<td>86.7%</td>
</tr>
<tr>
<td>SP</td>
<td>74.2%</td>
<td>86.3%</td>
<td>87.8%</td>
</tr>
<tr>
<td>TV</td>
<td>80.6%</td>
<td>96.7%</td>
<td>96.7%</td>
</tr>
<tr>
<td>AvAcc</td>
<td>86.2%</td>
<td>84.7%</td>
<td>91.2%</td>
</tr>
<tr>
<td>Conf±</td>
<td>4.99%</td>
<td>7.03%</td>
<td>3.59%</td>
</tr>
<tr>
<td>Time(fps)</td>
<td>8.03</td>
<td>3.92</td>
<td>8.31</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Related Work</th>
<th>Ours</th>
<th>HOGHOF [21]</th>
<th>MBH [21]</th>
</tr>
</thead>
<tbody>
<tr>
<td>CU</td>
<td>83.8%</td>
<td>91.1%</td>
<td>89.7%</td>
</tr>
<tr>
<td>DB</td>
<td>98.3%</td>
<td>90.4%</td>
<td>97.1%</td>
</tr>
<tr>
<td>EP</td>
<td>89.1%</td>
<td>93.7%</td>
<td>90.6%</td>
</tr>
<tr>
<td>ER</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>ES</td>
<td>86.5%</td>
<td>96.4%</td>
<td>91.9%</td>
</tr>
<tr>
<td>HS</td>
<td>93.7%</td>
<td>96.8%</td>
<td>100%</td>
</tr>
<tr>
<td>PS</td>
<td>83.4%</td>
<td>85.4%</td>
<td>92.0%</td>
</tr>
<tr>
<td>RP</td>
<td>96.8%</td>
<td>96.8%</td>
<td>93.7%</td>
</tr>
<tr>
<td>SB</td>
<td>86.7%</td>
<td>96.8%</td>
<td>95.3%</td>
</tr>
<tr>
<td>SP</td>
<td>87.8%</td>
<td>87.8%</td>
<td>90.9%</td>
</tr>
<tr>
<td>TV</td>
<td>96.7%</td>
<td>93.5%</td>
<td>90.3%</td>
</tr>
<tr>
<td>AvAcc</td>
<td>91.2%</td>
<td>93.5%</td>
<td>93.7%</td>
</tr>
<tr>
<td>Conf±</td>
<td>3.59%</td>
<td>2.61%</td>
<td>2.24%</td>
</tr>
<tr>
<td>Time(fps)</td>
<td>8.31</td>
<td>5.06</td>
<td>5.03</td>
</tr>
</tbody>
</table>
Table 4.7: Recognition rates on CHUN action dataset.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>BoVW</td>
<td>90.52%</td>
<td>96.79%</td>
<td>91.93%</td>
<td>95.18%</td>
<td>91.31%</td>
</tr>
<tr>
<td>Fisher</td>
<td>93.24%</td>
<td>95.58%</td>
<td>92.64%</td>
<td>95.23%</td>
<td>94.26%</td>
</tr>
</tbody>
</table>
Chapter 5

Fast activity detection via Statistical Sequential Change Detection (SSCD) using CUSUM

In this chapter, we expand our previous research, presented in Section 3.2 and Section 4.2, which focused on ADL recognition, and introduce a novel approach for activity detection, which is a necessary precursor to recognition in real applications. For effective video-based monitoring (e.g., AAL), the detection and recognition of Activities of Daily Living (ADLs) is central, and also the focus of this chapter. Activity detection and recognition from video for assisted living is based on unobtrusive ambient sensors, namely static video cameras, so that they do not disturb people in their daily life and can provide a comprehensive picture of their activities and lifestyle.
Sequential Statistical Change Detection (SSCD), presented in Chapter 4 segments trajectories based on changes that it detects on their motion pattern (i.e. HOF descriptor) in order to guarantee their time-invariant nature. Thus, this change detection (SSCD) performs motion-based temporal trajectory segmentation. This chapter 5 presents the application of sequential statistical change detection for the segmentation of the video into subsequences containing different activities, based on changes that it detects on their activity pattern which is encoded with the HOGHOF-Fisher distance from a SVM hypersphere of each activity, which has been built using a-priori training information.

5.1 Introduction

The automated analysis of real world videos is a challenging problem, as it requires the reliable discovery of unknown activities in time, that may occur at any moment, followed by their accurate classification, which is particularly demanding due to the various realizations of activities that may occur. In this work, we address both problems, that of activity discovery and that of recognition, by introducing novel, theoretically well-founded methods that are shown to lead to accurate detection and recognition rates, at a reduced computational cost.

The spatiotemporal localisation of activities is a necessary precursor to activity recognition in video sequences of long duration. In the literature, it is common to localize activities in space and time by deploying spatio-temporal sliding cuboids to detect activity subsequences, and then recognize the activities in them using a high level classifier. However, temporal sliding windows process video frames sequentially with an overlap and with multiple spatial and temporal windows in order to deal with scale variance, which significantly increases the computational cost and renders these techniques inappropriate for real-life scenarios, such as the detection and recognition of activities of daily living (ADLs) in videos of long duration.

In this work only regions of interest are examined in each frame, namely pixels where varying motion occurs, to reduce computational cost and improve the subsequent temporal localisation of activities and recognition rates. These regions correspond to a spatial binary mask, the Motion Boundary Activity Area (MBAA), that contains the regions in each video frame where varying motion occurs. We further reduce the duration of the video analysis to almost half of real time by applying coarse temporal segmentation before analyzing the video frames. This is achieved by extracting subsequences when MBAAs are detected and analyzing only the data in them: interest points are detected in MBAAs and tracked, as detailed in Sec. 5.2.2, and the video subsequences are considered to end when the trajectories in the MBAAs end. The resulting subsequence may still
contain multiple activities in it, however it requires less processing than the entire video, and will result in fewer false alarms due to the lack of uninformative frames with constant motion (or no motion).

For precise temporal segmentation, we introduce a novel method, Sequential Statistical Boundary Detection (SSBD), which leads to quicker and more reliable recognition outcomes at a much lower computational cost. This algorithm processes video frames sequentially using a non-overlapping sliding window, and extracts activity boundaries quickly via Cumulative Sum (CUSUM) sequential statistical change detection applied to the outcomes of a discriminative SVDD classifier. We first create a one-class classifier for normal activity patterns using a Support Vector Data Description (SVDD), and then calculate the instantaneous log-likelihood ratio (llr) between reference and current activity patterns. The llr is used in conjunction with the CUSUM termination test statistic for fast but robust activity localisation, resulting in accurate activity classification.

Figure 5.1 shows the main differences between the sliding window and SSBD approaches. Sliding window methods search through the entire video over multiple spatial and temporal scales in order to discover desired activities, while SSBD uses only a small number of frames (the reference data) to train its model and is then applied to each frame in an online manner, as described below. Initially, MBAAs spatially localize the activity in the video frame, eliminating the need for a costly thorough spatial search with multiscale windows. MBAAs are also used for initial coarse temporal segmentation, by terminating video subsequences when the MBAA pixels are all equal to zero, as this indicates that motion in them does not change any more. SSBD then uses the current (most recent) frames to estimate the log-likelihood ratio and incorporate it in the CUSUM test to find if a change from the normal (reference) pattern occurs, and detect the precise temporal boundaries of the activity to be classified.

In conclusion, we contribute to the literature on activity detection with the following:

- Motion Boundary Activity Areas (MBAA) are extracted for the spatial localisation of ADLs inside long videos that include ambiguous intervals (“noise”).
- Dense trajectories are extracted in MBAAs for temporal activity boundary detection, followed by the automatic extraction of video segments containing specific ADLs.
- Statistical Sequential Boundary Detection (SSBD) in activity patterns for fast ADL detection.
- Detection of ambiguous time intervals between ADLs, based on a one-against-one multi-class SVM and a voting procedure.
This chapter is organized as follows: Sec. 5.2 presents the ADL representation adopted in this work, while Sec. 5.3 presents our approach for the detection and classification of ADLs in videos. Experiments on data-sets from real scenarios are provided in Sec. 5.4, in order to evaluate the overall system.

### 5.2 Activity representation

In the first step of our algorithm, we use our activity representation framework, presented in Section 4.2, which is specifically designed for deployment in real-life scenarios at a low computational cost. This framework involves the construction of a trajectory structure over Motion Boundary Activity Areas (MBAA) by tracking densely sampled interest points (Sec. 5.2.1) and computing spatio-temporal cuboids around them in order to describe their motion and appearance throughout time. Hybrid cuboids are then constructed by concatenating HOGHOF and trajectory descriptors (Sec. 4.2.2). The overall procedure for the construction of our ADL descriptor is depicted in Fig. 5.2.

**Figure 5.1:** A person enters a smart home, where the presence of MBAAAs indicate the start of the activity sequence, which ends when the trajectories are terminated. Spatio-temporal sliding windows would need to process the whole sequence using 2 multi-scale overlapping windows, while SSBD builds a reference pattern and collects test data from new frames to detect changes in their pattern online. SSBD thus reduces the computational cost by avoiding the sequential application of SVMs for activity detection, classifying instead the subsequences only after a change has been found.

This chapter is organized as follows: Sec. 5.2 presents the ADL representation adopted in this work, while Sec. 5.3 presents our approach for the detection and classification of ADLs in videos. Experiments on data-sets from real scenarios are provided in Sec. 5.4, in order to evaluate the overall system.

### 5.2 Activity representation

In the first step of our algorithm, we use our activity representation framework, presented in Section 4.2, which is specifically designed for deployment in real-life scenarios at a low computational cost. This framework involves the construction of a trajectory structure over Motion Boundary Activity Areas (MBAA) by tracking densely sampled interest points (Sec. 5.2.1) and computing spatio-temporal cuboids around them in order to describe their motion and appearance throughout time. Hybrid cuboids are then constructed by concatenating HOGHOF and trajectory descriptors (Sec. 4.2.2). The overall procedure for the construction of our ADL descriptor is depicted in Fig. 5.2.
Figure 5.2: Block diagram of the ADL representation schema. Kurtosis analysis on OF values results in Motion Boundary Activity Areas (MBAA), where dense sampling is performed. Trajectory and local histograms (HOGHOF) are constructed around interest points in the MBAA on multiple scales, leading to the final ADL descriptor.

5.2.1 Optimal Trajectories over Motion Boundary Activity Areas

Spatial localisation of an ADL in our work is achieved by sampling interest points in regions of interest (RoIs), determined as those locations where motion undergoes a change. The RoIs used here are Motion Boundary Activity Areas (MBAA), which are presented in detail in Section 4.2.1. MBAA extraction analyses the gradients of optical flow values [18] statistically over $W$ successive frames for the real-time localisation of discriminative regions containing motion that is undergoing a change.

Once MBAA are found, a spatial grid is formed in each video frame creating a number of blocks with scale size $s = \{1, 2, \ldots, N\}$. The central pixel of each block is represented as $P^s(i)$ at the $i^{th}$ grid point and its corresponding location is given by $P^s(i) = (x, y)$. The accumulation of all grid points for each scale $s$ at each frame form $P_{all}^s$. These points are considered as candidate interest points $P_{c}^s \subseteq P_{all}^s$ only when more than 50% of the block belongs to the moving pixels of the MBAA. Candidate sampled points $P_{c}^s$ are then accumulated in a 2D-point set $S_{c}^s$ for each spatial scale $s$: $S_{c}^s = \{P_{c}^s(1), P_{c}^s(2), \cdots\}$, so that the final set comprises several candidate sets on multiple spatial scales $S_{c} = \{S_{c}^1 \cup S_{c}^2 \cup \cdots \cup S_{c}^N\}$. These points are then tracked using a boosted KLT tracker which uses a RANSAC estimator for homography computation, to remove bad correspondences $P_{b}^s(i)$ for each scale, denoted by $S_{b} = \{S_{b}^1 \cup S_{b}^2 \cup \cdots \cup S_{b}^N\}$, where $S_{b} = \{P_{b}^s(1), P_{b}^s(2), \cdots\}$. This provides the final validated interest points $P_{c}$ with good correspondences $S = \{S_{c} - S_{b}\}$.
The validated points $P_s \subseteq P_v$ will be tracked over time for a fixed temporal window $W = 15$ and then the interest points are concatenated according to their temporal order and form the final trajectory descriptor $Traj^s = \{P_s(k - W + 1), P_s(k - W), \cdots, P_s(k)\}$, where $k$ is the current frame and $W$ the optimal trajectory length.

When no more trajectories exist in the scene, the video subsequence is considered to end, and a sliding window (Sec. 5.3.1), enhanced with a statistical sequential boundary activity detector (Sec. 5.3.2), classifies the resulting test video segment in order to recognize the activities taking place within those frames. Figure 5.4 depicts an example where activity detection locates the start and end frame, while an in-between frame of a video segment is also depicted.

### 5.2.2 Hybrid spatio-temporal descriptor

Motivated by the highly accurate recognition rates of our hybrid descriptor for ADL representation, presented in Section 4.2, we also choose to use it here, in order to describe our ADL data. The experimental Section 4.4 shows that the recognition of ADLs in constrained environments (labs, nursing homes), presented in Sections 4.4.3.2, 4.4.3.3, 4.4.4, 4.4.5, is usually improved by including their spatial context in their description.

Temporally optimal trajectories over MBAAs (see Section 5.2.1) are then used to represent ADLs within video segments. HOG and HOF descriptors are computed around trajectory points $Traj^s$ in a rectangle block area with size equal to the grid block: $(8s, 8s)$ in order to encapsulate appearance and motion information in the ADL descriptor. The computed HOG and HOF descriptors of the same trajectory are then concatenated sequentially and subdivided into a $n_x \times n_y \times n_t$ ($n_x = n_y = 2$, $n_t = 3$) grid of cuboids. For each cuboid, histograms are averaged over time and normalised by the $L_2$ norm, in order to form the final local HOGHOF descriptor that will represent the ADLs performed around trajectory points. The trajectory descriptor is then concatenated with the local HOGHOF descriptor, thus adding spatial localisation information and generating the hybrid, global/local HOGHOF+Traj (see Section 4.2.2) that has been successfully applied for ADL recognition (Section 4.4).

### 5.3 Activity detection

The hybrid descriptors described above are extracted for all training video samples, which are manually segmented a priori so as to contain a specific activity, and are imported into a clustering algorithm, either K-Means or GMM in this work, in order to build a visual vocabulary for the ADLs to be detected.
Figure 5.3: Block diagram of ADL detection and recognition. Training pre-segmented samples are used to build visual vocabularies (i.e. cluster centers or means) and all data is encoded into a single size feature vector based on the VLAD/Fisher framework. \( C(C - 1)/2 \) multi-class SVM classifiers are built and a sliding window filters the unsegmented test data, in order to detect and recognise the ADLs and ambiguous temporal segments that occur within them.

VLAD and Fisher encoding methods are compared and depending on their classification accuracy rate and computational cost, they will be applied to the video segments’ hybrid descriptors and visual vocabulary cluster centers, in order to transform them into fixed size feature vectors. The central idea of the detection algorithm is to use a sliding window classifier, so that we can recognise the ADLs that occur during a video segment (Sec. 5.3.1). A faster option is provided in Section 5.3.2, where log-likelihood ratio estimation and the corresponding CUSUM sequential change detection test statistic are applied in non-overlapping temporal windows in order to detect a video segment’s activity boundaries (Sec. 5.3.2). The methodology followed in this work is shown in Fig. 5.3.

5.3.1 Sliding window activity detection

Temporal sliding windowing is a widespread technique that is used to detect and then recognise activities that may exist within an unsegmented video sample, i.e. a video sample that may contain more than one ADL. Before the application of the sliding window, a global representation needs to be constructed for each training video segment, to model the appropriate classifiers. A clustering algorithm (i.e. K-Means, GMM) is deployed to partition the ADL feature space and acquire the visual vocabulary corresponding to different activities. Afterwards, appropriate encoding (e.g. VLAD [37], Fisher [38])
Figure 5.4: Activity Areas on the left and trajectories on the right of each frame depict the start and end of a video segment.

takes place to characterise each video segment by a fixed size feature vector. The feature vectors obtained from training videos are used to train $C(C-1)/2$ SVM linear classifiers, where $C$ is the number of ADL classes, for the global representation of the training video classes.

Test video samples, on the other hand, are segmented using the activity detection algorithm and classified using a temporal sliding window. For activity detection using the sliding window technique, an activity is considered to start when the first trajectory is sampled from an MBAA, while the termination of a video segment is denoted when no other trajectories are found in that MBAA. Fig. 5.4 depicts an example where activity detection locates the start and end frame, and also shows an in-between frame of a video segment. These video segments are then classified by a temporal sliding window of size $W_0$, with a sampling step $J_0$. For $J_0 < W_0$, there is temporal overlap among the predictions, and thus better localisation of activity boundaries (i.e. the start and end of an activity) is achieved.

One-against-one SVM classification is used to predict each windowed video segment’s class, and a voting schema is applied, in order to declare the prominent ADL recognised by the classifiers. The class with the highest number of votes always wins, while in the case of a draw, a second round of voting is performed among the 1st and 2nd classifiers, and the one that wins among these two is the detected activity. In case of triple or higher draws, we announce ambiguity among classes (a scenario that usually occurs during a
transition between clearly detected activities) and no recognition data is stored. Fig. 5.5 visualizes ambiguous activity states and an MBAA sample.

5.3.2 Statistical Sequential Boundary Detection

The detection of activities, i.e. activity boundaries in a video segment, can be extracted in a more principled manner than by the sliding window technique, by applying statistical sequential change detection, namely the Cumulative Sum (CUSUM) approach, to the test video samples, which is also faster and more accurate. After having created the visual vocabulary from the training data, we proceed with the analysis of the test video segments (unsegmented video samples), which are derived from video subsequences resulting when all trajectories are terminated (see Section 5.2.1). ADL descriptors are then encoded into VLAD or Fisher vectors at each frame instance and Statistical Sequential Boundary Detection (i.e. SSBD) takes place.

The reference window initially has size $W_0$, equal to the number of frames that are recorded per 1 second (FPS). Experiments with larger windows gave significantly lower recognition rates (See Figure 5.10), while the window needs to be at least as long as the $fps$ in order to capture the activity in the recording. This gives the initial reference data $X_0 = \{x_{i+1}, x_{i+2}, \ldots, x_{i+W_0}\}$, where $x_j, \forall j \in [i+1, \ldots, i+W_0]$ is the respective VLAD/Fisher descriptor at each frame instance $j$. The $X_0$ samples are then used to train a Support Vector Data Description (SVDD) model, which will create a hypersphere around them, enclosing similar ADLs and excluding outliers. Thus, given the reference data $X_0$, we need to solve the optimisation problem:

$$\min_{R, a, \xi} \quad R^2 + C \sum_{j=i+1}^{i+W_0} \xi_j$$

subject to $\|\phi(x_j) - a\|^2 \leq R^2 + \xi_j, \forall j \in [i+1, i+W_0]$  \hspace{1cm} (5.1)
in order to obtain the radius of the hypersphere \( R \), the Lagrange multipliers \( a \) and the slack variables \( \xi \) that will define the reference SVDD model, where \( \phi() \) is a function mapping data to a higher dimensional space, and \( C > 0 \) is a user specified parameter. A characteristic paradigm of a SVDD hypersphere is depicted in Fig. 5.6 which is a toy example from [103].

After equation (5.1) is solved and a SVDD reference model is determined, we compute all reference distances from the hypersphere:

\[
D_0 = \{d_j = \|\phi(x_j) - a\|^2 : \forall j \in [i + 1, ..., i + W_0]\}
\]  

(5.2)

When the number of the distances that are enclosed in the hypersphere (\( d_j \leq R^2 \)) is bigger than \( W_0/2 \) (i.e. more than the half of the reference data is in the hypersphere), we approximate their probability density function (pdf) \( f(D_0, \theta_0) \) by its deterministic parameters \( \theta_0 \) given by the mean \( \mu_{D_0} \) and variance \( \sigma^2_{D_0} \) of \( D_0 \) for a Gaussian pdf. Otherwise, the first \( J_0 \) reference data are considered to contain too many outliers and are therefore insufficient for approximating the pdf. In that case, the window that is used
for reference sampling slides by $J_0$, so that a new set of data is extracted (i.e. $i = i + J_0$) and processed in the same manner. This procedure can be repeated until the reference window exceeds the end frame of the video segment.

Once the SVDD model is built for the first $W_0$ reference frames (e.g. for frames $i$ to $i + W_0$ in Fig 5.7), we examine the subsequent frames in order to detect changes in them. The first $J_0$ frames after frame $i + W_0$ form the test window $W_1 = J_0$, with the minimum number of frames needed to approximate the “test” distribution. Thus, we extract at each frame instance $k$ the test data $X_1 = \{x_{k-(W_1-1)}, x_{k-(W_1-2)}, \cdots, x_k\}$ and use the first $W_1 - 1$ in order to compute the ”test” distribution and check the last one ($x_k$) to detect a new change. The test window $W_1$ is increased by 1 for each new frame instance, if no change is detected and the test calculation repeats with the new test data. This procedure continues until $W_1 > W_0$, i.e. the test data $W_1$ becomes larger than the reference data $W_0$. The reference data are then updated by including test data: $X_0' = X_0 \cup X_1$ and the procedure restarts. This procedure is repeated until all $N$ video segment frames are tested. It should be noted that the size of the window with the test data $W_1$ needs to be larger than $J_0$, as our experiments with ADL datasets demonstrated that at least $J_0$ samples are needed to give statistically reliable approximation of the test pdf $f(D_1, \theta_1)$.

More specifically, test distances are computed from the reference SVDD hypersphere as soon as we have test data:

$$D_1 = \{d_j = \|\phi(x_j) - a\|^2 : \forall j \in [k - (W_1 - 1), ..., k]\}. \quad (5.3)$$

The first $W_1 - 1$ distances $d_j$ until the current frame $k - 1$ are used in order to compute the parameters $\theta_1$ of $f(D_1, \theta_1)$ for frame $k$, while the $d_k$ distance is used in order to determine whether a change occurs within the ADL patterns.

Once the reference and test data distributions have been approximated by computing $\theta_0$ and $\theta_1$, we can detect an unknown moment of change in an online manner by applying Statistical Sequential Change Detection (SSCD), as in Section 4.2.3.2. The SSCD used here is based on the commonly used instantaneous log-likelihood ratio (LLRT) between the reference and test pdfs.

For quickest online change detection [97], we implement Cumulative Sum (CUSUM) SSCD, as described in Section 4.2.3.2. However, in practice there maybe be an abrupt increase or decrease in $\theta$, respectively, because a change can be reflected either by an increase from the hyperspheres centre $R$ or by an abrupt decrease from it. So we deploy the two-sided CUSUM testing, which uses two CUSUM tests, as proposed by
The \( s^i_k \) and \( s^d_k \) are used in the iterative formula for CUSUM sequential detection:

\[
S^i_k = s^i_k + S^i_{k-1}, \quad S^d_k = s^d_k + S^d_{k-1} \\
G^i_k = \max(0, s^i_k + G^i_{k-1}), \quad G^d_k = \max(0, s^d_k + G^d_{k-1})
\]

with initial values equal to zero \( S^i_0 = 0, \ S^d_0 = 0, \ G^i_0 = 0, \ G^d_0 = 0 \).

For two-sided online change detection, whenever \( G^i_k \) or \( G^d_k \) exceeds a threshold \( h > 0 \), which is set separately for each dataset by following a cross validation procedure (See Table 5.4), the detection algorithm searches for the minimum of \( S^i_k \) or \( S^d_k \) respectively, and determines a change point respectively as:

\[
\hat{n}_c = \arg \min_{k-(W_0-1) \leq n_c \leq k} (S^i_{n_c}) \\
\hat{n}_c = \arg \min_{k-(W_0-1) \leq n_c \leq k} (S^d_{n_c})
\]

\( X_0 \) is then updated by including \( X_1 \) so that \( X'_0 = X_0 \cup \{x_j : j \in [i+1, ..., i+\hat{n}_c]\} \) and an SVM classifier is used to predict the ADL inside this range. The reference window then slides by \( \hat{n}_c \) (i.e. \( i = i + \hat{n}_c \)) and the procedure restarts. If \( G^i_k \) or \( G^d_k \) do not exceed \( h > 0 \), then \( d_k \) is included in \( D_1 \), \( \theta_1 \) is updated accordingly and the change detection procedure restarts until \( W_1 > W_0 \).

When \( W_1 \) surpass \( W_0 \) and no change is detected within the test samples, then \( X_1 \) data are concatenated with the reference \( X'_0 = X_0 \cup X_1 \), updating the SVDD model and the algorithm reset from the start.

Fig. 5.7 shows the overall SSBD procedure. Test and reference are initially sampled, SVDD are computed from the reference and distances are computed in order to build \( \theta_0 \), \( \theta_1 \) and whether a change occurs at time instance \( k \). The algorithm is repeated until a change occurs, \( W_1 \) exceeds \( W_0 \) or \( k \) reaches the end of the video segment \( N \). Classification of the reference data follows on each occasion in order to recognize the ADL that occurs.
5.3.2.1 Practical implementation of CUSUM

In order to apply CUSUM SSCD, we need to determine the pdf of the data before and after a change. As the distributions are unknown, we determine them empirically by fitting appropriate model parameters to our data. However, this implies that the family of pdfs of the data is known a priori. In this work, we make the assumption that the model for the pdf is a Gaussian distribution, and we verify it empirically by applying the Kolmogorov-Smirnov (K-S) test. Indeed the K-S tests applied to the data showed that it can indeed be satisfactorily approximated by a Gaussian pdf given by:

\[
\text{pdf}(f(X_{0|1})) = \frac{1}{\sigma_{D_{0|1}} \sqrt{2\pi}} \exp\left( - \frac{(d_k - \mu_{D_{0|1}})^2}{2\sigma_{D_{0|1}}^2} \right)
\]  

(5.8)
By substituting Eq. (5.8) into Eq. (5.4), the instantaneous log-likelihood ratio becomes:

\[
s^i_k = \ln \frac{\sigma_{D_0}}{\sigma_{D_1}} + \frac{(d_k - \mu_{D_0})^2}{2\sigma^2_{D_0}} - \frac{(d_k - \mu_{D_1})^2}{2\sigma^2_{D_1}}
\]

\[
s^d_k = -\ln \frac{\sigma_{D_0}}{\sigma_{D_1}} - \frac{(d_k - \mu_{D_0})^2}{2\sigma^2_{D_0}} + \frac{(d_k - \mu_{D_1})^2}{2\sigma^2_{D_1}},
\]

(5.9)

where \(x_i\) is the new test sample that is taken under consideration and \(\mu_{D_0}, \mu_{D_1}, \sigma_{D_0}, \sigma_{D_1}\) are empirically determined parameters of the reference and test distributions. They are approximated using the VLAD/Fisher descriptors at each frame instant based on the initial \(W_0\) frames and the most recent \(W_1\) test samples, respectively.
Algorithm 1 elaborate on the steps required for solving activity boundary detection issue.

**Data:** Set Video segment with $N$ samples, $i, W_0, J_0, h$ :

```
while $(i + W_0) < N$ do
    Set *reference window* : $X_0 = \{x_{i+1}, x_{i+2}, \ldots, x_{i+W_0}\}$ ;
    Calculate SVDD model $(a,R,\xi)$ using $X_0$;
    Calculate $D_0 = \{d_j = \| \phi(x_j) - a \|^2 : \forall j \in [i+1, \ldots, i+W_0] \}$ ;
    
    if number of $(d_j < R)$ is less than $(W_0/2)$ then
        Slide reference window by $J_0$ so that $i = i + J_0$ and restart alg. ;
    else
        Calculate $\theta_0 = (\mu_{D_0}, \sigma_{D_0}^2)$ using $D_0$ ;
        Set $W_1 = J_0$ ;
        while $(k < N) \& (W_1 < W_0)$ do
            Set *test window* : $X_1 = \{x_{k-(W_1-1)}, x_{k-(W_1-2)}, \ldots, x_k\}$ ;
            Calculate $D_1 = \{d_j = \| \phi(x_i) - a \|^2 : \forall j \in [k-(W_1-1), \ldots, k] \}$ ;
            Calculate $\theta_1 = (\mu_{D_1}, \sigma_{D_1}^2)$ using $D_1 \bigcup d_k$ ;
            Calculate instant log-likelihood ratios:
            $s'[k] = \ln \frac{\sigma_{D_0}}{\sigma_{D_1}} + \frac{(d_k - \mu_{D_0})^2}{2\sigma_{D_0}^2} - \frac{(d_k - \mu_{D_1})^2}{2\sigma_{D_1}^2}$ ;
            $s'[k] = -\ln \frac{\sigma_{D_0}}{\sigma_{D_1}} - \frac{(d_k - \mu_{D_0})^2}{2\sigma_{D_0}^2} + \frac{(d_k - \mu_{D_1})^2}{2\sigma_{D_1}^2}$ ;
            Update *instant* test statistics:
            $S'[k] = S'[k-1] + s'[k]$ ;
            $S'[k] = S'[k-1] + s'[k]$ ;
            Update *cumulative* test statistics:
            $G'[k] = \{G'[k-1] + s'[k]\}^+$ ;
            $G'[k] = \{G'[k-1] + s'[k]\}^+$ ;
            if $(G'[k] > h > 0)$ then
                if $G'[k] > h > 0$ then
                    $n_c = \arg \min_{k-(W_1-1) \leq n_c \leq k} S'[n_c - 1]$ ;
                end
                if $G'[k] > h > 0$ then
                    $n_c = \arg \min_{k-(W_1-1) \leq n_c \leq k} S'[n_c - 1]$ ;
                end
                Predict $\hat{X} = \{x_j : j \in [i+1, \ldots, n_c] \}$ ;
                Slide reference window by $n_c$ to $i = i + n_c$ and restart alg. ;
            else
                $D_1 = D_1 \bigcup d_k$, $W_1 = W_1 + 1$
            end
        end
        Update reference data: $X'_0 = X_0 \bigcup X_1$ and restart alg. ;
    end
end
```

**Algorithm 1:** SSBD algorithm for a Gaussian model of the distances
5.4 Experiments

In this section we present experimental results for spatial localisation of activities, as well as temporal localisation and subsequent recognition. The experimental results take place on benchmark datasets to compare with the SoA, as well as on realistic datasets recorded in lab and home environments.

We evaluate the performance of spatial localisation on the benchmark UCF sports dataset, which comprises of videos of pre-segmented sports activities, and also provides groundtruth for spatial location of activities. Temporal localisation is tested on two long duration, realistic datasets recorded for the EU project Dem@Care in home-like environments, in the Centre Hospitalier Universitaire de Nice (CHUN) in Nice, France (CHUN dataset), and the day center of the Greek Association for Alzheimer’s Disease and Related Disorders (GAADRD) in Thessaloniki, Greece (Dem@Care dataset). The spatiotemporally localized activities in these two datasets are also recognized by calculating average accuracy over all classes for standard vocabulary sizes ranging from 32 to 256 for VLAD and Fisher. The maximum vocabulary size was set based on the results of previous works on activity recognition [46], [37], [21], where these vocabulary sizes led to optimal recognition rates. Indeed, we observed experimentally that outside this vocabulary range, improvements or degradations are only in the range of 1%. SSBD parameters $(W_0, h)$ are set after thorough experimental work on DemCare dataset and are presented in Section 5.4.2.1. Vocabulary sizes and encoding schemes are also computed separately for each dataset (DemCare-CHUN) in order to acquire optimal results for each one of them. Experiments were performed on a i5$-$3570K CPU, running at 3.4GHz, while no GPU was used to accelerate the processing time of our algorithm.

To evaluate spatial localisation accuracy, we consider the intersection of the results with the ground truth, instead of their union, as this metric is most commonly used in the literature [71], [70], [72]. We consider the OV20 evaluation criterion [13], which requires the Jaccard coefficient (intersection over union) to be over 20% between groundtruth and detected activities in the time domain, in order to consider a detected activity to be a true positive.

5.4.1 Spatial localisation

The UCF sports dataset is used to evaluate the spatial localisation accuracy of the MBAA mask. This dataset consists of 150 video samples recorded under different viewpoints and for varying camera motions. Frame per second processing time is also provided to evaluate the method’s computational cost. We achieved a processing time of 0.8732
frames per second (fps), because the image frames have high resolution. Table 5.1 shows the accuracy rates when spatial localisation was performed on the subset of UCF videos proposed by [71] and also used in [70] and [72]. As we can see, our algorithm outperforms the literature for almost all activities, except for ride and run which are localized more accurately by [71] which require a very robust pretrained detector, as in [71], in order to deal with severe background clutter and camera motion. We thus conclude that our spatial localisation algorithm is very accurate, with a very reasonable computational cost, outperforming more sophisticated and cumbersome spatial localisation techniques on most activities of the benchmark UCF dataset.

5.4.2 Temporal localisation

The method proposed here for temporal localisation was applied to two datasets of long duration that were recorded during the Dem@Care project at CHUN and GAADRD. The SSBD parameters \((W_0, h)\) are determined empirically as described in Sec. 5.4.2.1. Vocabulary sizes and encoding schemes are also computed separately for each dataset, so as to acquire optimal results for each of them.

5.4.2.1 DemCare action dataset

The proposed method was also applied to two real-life datasets that were recorded during the Dem@Care project at CHUN and DemCare 2.3.6. The DemCare action dataset consists of 1 hour and 52 min recordings of 32 people with dementia that perform ADLs in a home-like environment. The camera viewpoint is in front of the person while they perform directed ADLs (i.e. activities dictated by a clinician). The recording frequency is 8 fps and our algorithm performance achieved 3.34 and 4.54 fps for simple sliding window and SSBD respectively, which is a near real time process achievement considering that for 1.5 – 2 video frames of the actual video, one frame is processed by our ADL detection

<table>
<thead>
<tr>
<th>Activity</th>
<th>All frames</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>[71]</td>
</tr>
<tr>
<td>Dive</td>
<td>37.0%</td>
</tr>
<tr>
<td>Golf</td>
<td>-</td>
</tr>
<tr>
<td>Kick</td>
<td>-</td>
</tr>
<tr>
<td>Ride</td>
<td>64.0%</td>
</tr>
<tr>
<td>Run</td>
<td>61.9%</td>
</tr>
<tr>
<td>Skate</td>
<td>-</td>
</tr>
<tr>
<td>Swing-b</td>
<td>-</td>
</tr>
<tr>
<td>Swing-s</td>
<td>-</td>
</tr>
<tr>
<td>Walk</td>
<td>-</td>
</tr>
<tr>
<td>Avg.</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 5.1: Intersection over Union over all classes for UCF action dataset
algorithm. The ADLs observed include: Cleaning 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) and Talk to Visitor (TV). They included large anthropometric differences and activity performance styles, while the ADLs took place continuously, introducing additional difficulty and increasing the computational cost needed for activity detection.

Table 5.2 aggregates the average accuracy rates over all ADL classes when our representation scheme was used in a one-subject-against-all scenario. Fig. 5.9 shows the classification performance of HOGHOF and HOGHOF+Traj. It is obvious that the Fisher encoding schema surpasses the VLAD recognition rates and the hybrid HOGHOF+Traj descriptor outperforms the local HOGHOF, even for small vocabularies. With VLAD, on the other hand, the inclusion of trajectory information does not significantly improve recognition rates. This is a reasonable result, as we have seen also in Chapters 3, 4, that Fisher encoding is the only representation scheme that can benefit from the proposed representation scheme.

Table 5.3 depicts the recognition rates achieved when leave-one-Subject-out was used for classifying the DemCare videos with Fisher encoding and a vocabulary of size 64. Most ADLs are distinguished quite clearly from each other, except for some that are very similar to each other, such as ES and PS which are confused with DB and SB (a similar action takes place in a similar location).
Table 5.2: Average accuracy over all classes for DemCare action dataset

<table>
<thead>
<tr>
<th>Encoding</th>
<th>Vocabulary size</th>
<th>Descriptor</th>
<th>Average Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>VLAD</td>
<td>32</td>
<td>HOGHOF</td>
<td>75.90%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HOGHOF+Traj</td>
<td>76.02%</td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>HOGHOF</td>
<td>80.69%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HOGHOF+Traj</td>
<td>77.13%</td>
</tr>
<tr>
<td></td>
<td>128</td>
<td>HOGHOF</td>
<td>80.66%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HOGHOF+Traj</td>
<td>78.15%</td>
</tr>
<tr>
<td></td>
<td>256</td>
<td>HOGHOF</td>
<td>81.75%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HOGHOF+Traj</td>
<td>80.54%</td>
</tr>
<tr>
<td>Fisher</td>
<td>32</td>
<td>HOGHOF</td>
<td>83.20%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HOGHOF+Traj</td>
<td>85.10%</td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>HOGHOF</td>
<td>81.80%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HOGHOF+Traj</td>
<td>88.10%</td>
</tr>
<tr>
<td></td>
<td>128</td>
<td>HOGHOF</td>
<td>83.02%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HOGHOF+Traj</td>
<td>87.60%</td>
</tr>
<tr>
<td></td>
<td>256</td>
<td>HOGHOF</td>
<td>81.83%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HOGHOF+Traj</td>
<td>87.20%</td>
</tr>
</tbody>
</table>

Figure 5.9: HOGHOF and HOGHOF+Traj comparison when VLAD(left) and Fisher(right) encoding is applied on DemCare dataset.

Table 5.3: Confusion matrix for ADL recognition, using average accuracy scores over all classes, on the DemCare dataset

<table>
<thead>
<tr>
<th></th>
<th>CU</th>
<th>DB</th>
<th>EP</th>
<th>ER</th>
<th>ES</th>
<th>HS</th>
<th>PS</th>
<th>RP</th>
<th>SB</th>
<th>SP</th>
<th>TV</th>
</tr>
</thead>
<tbody>
<tr>
<td>CU</td>
<td>0.853</td>
<td>0.029</td>
<td>0.029</td>
<td>0.029</td>
<td>0.881</td>
<td>0.029</td>
<td>0.029</td>
<td>0.829</td>
<td>0.029</td>
<td>0.029</td>
<td>1.00</td>
</tr>
<tr>
<td>DB</td>
<td>0.02</td>
<td>0.939</td>
<td>0.041</td>
<td>1.00</td>
<td>0.022</td>
<td>0.029</td>
<td>0.029</td>
<td>0.029</td>
<td>0.029</td>
<td>0.029</td>
<td>0.029</td>
</tr>
<tr>
<td>EP</td>
<td>0.031</td>
<td>0.844</td>
<td>0.031</td>
<td>0.029</td>
<td>0.029</td>
<td>0.029</td>
<td>0.029</td>
<td>0.029</td>
<td>0.029</td>
<td>0.029</td>
<td>0.029</td>
</tr>
<tr>
<td>ER</td>
<td></td>
<td></td>
<td>1.00</td>
<td>0.244</td>
<td>0.711</td>
<td>0.029</td>
<td>0.029</td>
<td>0.029</td>
<td>0.029</td>
<td>0.029</td>
<td>0.029</td>
</tr>
<tr>
<td>ES</td>
<td>0.244</td>
<td>0.022</td>
<td>1.00</td>
<td>0.029</td>
<td>0.029</td>
<td>0.029</td>
<td>0.029</td>
<td>0.029</td>
<td>0.029</td>
<td>0.029</td>
<td>0.029</td>
</tr>
<tr>
<td>HS</td>
<td>0.059</td>
<td>0.029</td>
<td>0.022</td>
<td>0.711</td>
<td>0.906</td>
<td>0.029</td>
<td>0.029</td>
<td>0.029</td>
<td>0.029</td>
<td>0.029</td>
<td>0.029</td>
</tr>
<tr>
<td>PS</td>
<td>0.029</td>
<td>0.086</td>
<td>0.022</td>
<td>0.022</td>
<td>0.022</td>
<td>0.022</td>
<td>0.022</td>
<td>0.022</td>
<td>0.022</td>
<td>0.022</td>
<td>0.022</td>
</tr>
<tr>
<td>RP</td>
<td>0.031</td>
<td>0.031</td>
<td>0.029</td>
<td>0.029</td>
<td>0.029</td>
<td>0.029</td>
<td>0.029</td>
<td>0.029</td>
<td>0.029</td>
<td>0.029</td>
<td>0.029</td>
</tr>
<tr>
<td>SB</td>
<td>0.061</td>
<td>0.061</td>
<td>0.061</td>
<td>0.061</td>
<td>0.061</td>
<td>0.061</td>
<td>0.061</td>
<td>0.061</td>
<td>0.061</td>
<td>0.061</td>
<td>0.061</td>
</tr>
<tr>
<td>SP</td>
<td>0.029</td>
<td>0.059</td>
<td>0.029</td>
<td>0.059</td>
<td>0.059</td>
<td>0.059</td>
<td>0.059</td>
<td>0.059</td>
<td>0.059</td>
<td>0.059</td>
<td>0.059</td>
</tr>
<tr>
<td>TV</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.00</td>
</tr>
<tr>
<td>AA</td>
<td>0.881</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Experiments with varying $W_0$ are depicted in Fig. 5.10. When $W_0$ is set at the frequency of the camera recording rate ($FPS = 8$), the average accuracy is increased significantly. Lower $W_0 < FPS = 8$ provides the worst results as the number of samples in SSBD is not sufficient to compute the pdfs, while larger window $W_0 > FPS = 8$ worsens the average accuracy rates, as the boundaries are coarser and fast changes are missed.

Several experiments were also carried out with SSBD’s sensitivity threshold $h$ (Table 5.4) in order to evaluate the detection performance of SSBD algorithm. Activity classification after detection over all classes is measured with OV20 (i.e. 20% overlap between the groundtruth and detected intervals), and we report the number of detections and computational time over FPS. Generally we observed a slight decrease in accuracy when $h$ increased, while no variations were observed when $h$ changed from 20 to 50 and then to 100. This is attributed to the fact that a small $h$ can detect more changes, as it increases sensitivity, while large $h$ only finds fast, abrupt ADL changes, so some smoother ADL changes may be missed or detected with delay. This can be also observed in Table 5.4, where a low $h$, with $h = 10$, leads to nearly 200 detections less than $h = 100$ (i.e. 5% more). However, this phenomenon does not have any effect on the computational cost of the algorithm, so we choose to have a low sensitivity threshold ($h = 10$) in our experiments instead.

Fig. 5.11 shows the performance of the HOGHOF without SSBD and HOGHOF+Traj with/without SSBD for different Jaccard Coefficient values. Trajectory coordinates boost HOGHOF descriptor and lead to better classification rates in comparison with simple HOGHOF. In addition, SSBD improves HOGHOF+Traj even more by increasing
recognition rates by almost 10% at a lower computational cost, as it achieved 6.66 fps against the 3.34 fps of HOGHOF+Traj without SSBD.

Fig. 5.12 shows classification by detection over all classes for the HOGHOF and HOGHOF+Traj descriptors applied to the DemCare action dataset. The OV20 Jaccard coefficient was used as a comparison metric and a Fisher descriptor with 64 cluster centers was picked for the vocabulary size.
Chapter 5. Activity detection using SSBD and sliding window

The results are quite similar for distinct classes (see ER/DB/ES/CU), while great differences are noted in the recognition of static ADLs, where trajectory coordinates make a big difference (see TV/RP/HS), with HOGHOF+Traj outperforming HOGHOF. Other activities (classes), such as SB/PS and SP/EP, located in the same region and including very similar actions were usually confused and performed quite poorly with both descriptors. SSBD improved accuracy rates almost in all ADL categories achieving the best recognition rate in our experiments.

Precision-Recall curves are also provided for SSBD detection algorithm in Fig. 5.13, where we can see how our detection algorithm performs over all activities. Some activities are detected with very high average precision score, such as clean up, eat snack, enter room and talk to visitor, while others contain a lot false alarms in their predictions, such as prepare snack, serve beverage and start phonecall leading to a relatively small average precision. In the figure, we also include a precision-recall curve for all detections, where we can see the overall evaluation of our detection algorithm.

5.4.2.2 CHUN action dataset

The CHUN dataset consists of 15 hr and 10 min recordings of 64 PwD that perform ADLs in a Lab environment. The camera viewpoint is such that it monitors the whole room where the PwD is performing semi-directed ADLs (i.e. they followed instructions for performing a set of ADLs listed on a paper). The recording frequency is at 8 fps and our algorithm performance achieved 3.55 fps without SSBD and 4.24 fps with SSBD for video analysis, which is a near real time process achievement considering that for 2 video frames of the actual video, one is processed by our ADL detection algorithm.
The ADLs observed are: (AP) answering phone and (DP) dialing phone, (LoM) look on map, (PB) pay bill, (PD) prepare drugs, (PT) prepare tea, (RP) read paper, (WP) water plant and (WtV) watch TV. They included large anthropometric variations and activity performance styles, while severe occlusions introduced great difficulty in discriminating actions. Figure 5.14 depicts some activities with their trajectory and HOGHOF rectangles. Despite these challenges, Table 5.15 shows that high accuracy results were achieved, proving the applicability of our technique in realistic scenarios.

Table 5.6 shows that the inclusion of trajectory coordinates in the HOGHOF descriptor improves recognition performance with both the VLAD and Fisher encoding schemas. The differences between HOGHOF and HOGHOF+Traj are depicted clearly in Fig. 5.15.
VLAD achieved very accurate recognition rates, similar to those achieved by Fisher encoding, but at a lower computational and memory cost (i.e., fewer distance computations for the same vocabulary size). This motivated us to test our detection schema on the CHUN dataset with a VLAD descriptor of 64 cluster centers, which achieves a fair balance among accuracy and computational cost, and used the OV20 Jaccard coefficient as a comparison metric, with the resulting average accuracy shown in Fig. 5.16. The hybrid HOGHOF, i.e., including trajectory coordinates, outperformed simple HOGHOF in all categories, rendering it more appropriate for this challenging, real-life dataset. The inclusion of SSBD algorithm outperformed both of the former ones in all ADL categories except Water Plant (WP).

Table 5.7 depicts the recognition rates achieved when leave-one-Subject-out was used to classify the CHUN videos. The robustness of our recognition algorithm is obvious, as all activities, except for AP (which, as before, is confused with the very similar DP), are very accurately classified.

Fig. 5.17 shows the performance of the HOGHOF without SSBD and HOGHOF + Traj with and without SSBD when varying the Jaccard Coefficient. For all percentages of overlap, we have a consistent improvement in accuracy with the inclusion of trajectory coordinates in the HOGHOF descriptor, while SSBD outperform both HOGHOF and HOGHOF + Traj without SSBD in all overlap ratios. Considering that SSBD achieves
### Table 5.6: Average accuracy over all classes for the CHUN dataset

<table>
<thead>
<tr>
<th>Encoding</th>
<th>Vocabulary size</th>
<th>Descriptor</th>
<th>Average Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>VLAD</td>
<td>32</td>
<td>HOGHOF</td>
<td>90.65%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HOGHOF+Traj</td>
<td>96.43%</td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>HOGHOF</td>
<td>91.66%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HOGHOF+Traj</td>
<td>96.43%</td>
</tr>
<tr>
<td></td>
<td>128</td>
<td>HOGHOF</td>
<td>91.32%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HOGHOF+Traj</td>
<td>96.66%</td>
</tr>
<tr>
<td></td>
<td>256</td>
<td>HOGHOF</td>
<td>93.47%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HOGHOF+Traj</td>
<td>96.78%</td>
</tr>
<tr>
<td>Fisher</td>
<td>32</td>
<td>HOGHOF</td>
<td>92.12%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HOGHOF+Traj</td>
<td>95.96%</td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>HOGHOF</td>
<td>93.34%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HOGHOF+Traj</td>
<td>95.75%</td>
</tr>
<tr>
<td></td>
<td>128</td>
<td>HOGHOF</td>
<td>92.36%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HOGHOF+Traj</td>
<td>95.55%</td>
</tr>
<tr>
<td></td>
<td>256</td>
<td>HOGHOF</td>
<td>91.57%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HOGHOF+Traj</td>
<td>95.35%</td>
</tr>
</tbody>
</table>

**Figure 5.15:** HOGHOF and HOGHOF+Traj comparison when VLAD(left) and Fisher(right) encoding is applied on the CHUN action dataset.

### Table 5.7: ADL recognition in terms of average accuracy on the CHUN dataset

<table>
<thead>
<tr>
<th></th>
<th>AP</th>
<th>DP</th>
<th>LoM</th>
<th>PB</th>
<th>PD</th>
<th>PT</th>
<th>RP</th>
<th>WP</th>
<th>WtV</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP</td>
<td>0.891</td>
<td>0.091</td>
<td>0.018</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DP</td>
<td>0.03</td>
<td>0.97</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LoM</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PB</td>
<td>0.009</td>
<td>0.982</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PD</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PT</td>
<td></td>
<td></td>
<td></td>
<td>0.987</td>
<td>0.013</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.948</td>
<td>0.95</td>
<td>0.983</td>
</tr>
<tr>
<td>WP</td>
<td>0.025</td>
<td></td>
<td></td>
<td></td>
<td>0.025</td>
<td>0.017</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WtV</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.968</td>
</tr>
</tbody>
</table>

**AA** 0.968
better recognition rates with lower computational cost than the former, we can safely say that it is a better solution. As we can see from Table 5.8 HOGHOF+Traj SSBD surpass HOGHOF+Traj without SSBD by almost 2.5% in almost 6 hours faster, analysing on average almost 4 frames per second (the half of the camera recording rate, which is 8 fps).

Precision-Recall curves are also provided for SSBD detection algorithm in Fig. 5.18, where we can see how our detection algorithm performs over all activities. In this case, some activities that dominate over others, such as Call Phone does to Answer Phone, leading to a very low Answer Phone score and a relatively high Call Phone. Furthermore, activities that are performed close to the camera like Prepare Drugs and Pay Bill have the advantage of a more accurate appearance and motion descriptor (i.e. HOGHOF) and thus achieve an almost 100% average precision score, while others, which are far away from the camera cannot be distinguished clearly and contain a lot false positives, such as Look on Map. In the figure, we also include a precision-recall curve for all detections, where we can see the overall evaluation of our detection algorithm.
Figure 5.17: Average accuracy for different overlap ratios when HOGHOF and HOGHOF+Traj and SSBD are applied on the CHUN dataset

5.5 Conclusion

In this work, a novel method for accurate activity detection and recognition is proposed, offering a reduced computational cost. Regions of interest, that is the Motion Boundary Activity Areas (MBAAs) are located by separating the moving pixels from the static ones in a video, and dense, multi-scale sampling takes place in the MBAAs in order to extract interest points. HOGHOF descriptors characterise the interest points, which are tracked over time using a KLT tracker supplemented by a RANSAC homography outlier estimator. The resulting trajectories are used to determine when an activity starts and ends, providing Activity Detection in long videos. SoA encoding techniques are used in combination with a BoVW framework in order to recognise the activities taking place, and also introduce the characterisation of “ambiguity intervals”, located between recognised activities. This method is tested on well known benchmark data (URADL) where it is shown to achieve very high, SoA, performance. Experiments also take place with videos recorded in more challenging, real life datasets, recorded in home-like environments from CHUN and DemCare, where it can be seen that the proposed method leads to very accurate activity detection and recognition rates.
Chapter 5. Activity detection using SSBD and sliding window

Answer Phone − AP: 5.88%

Call Phone − AP: 73.30%

Look on Map − AP: 28.22%

Pay Bill − AP: 96.38%

Prepare Drugs − AP: 100.00%

Prepare Tea − AP: 85.65%

Read paper − AP: 63.81%

Water plant − AP: 73.75%

Watch TV − AP: 73.31%

All − AP: 60.59%

Figure 5.18: Precision Recall curves over all classes for CHUN dataset.
Chapter 6

Conclusions and Future Work

Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1 Conclusions</td>
<td>117</td>
</tr>
<tr>
<td>6.2 Future work</td>
<td>118</td>
</tr>
</tbody>
</table>

6.1 Conclusions

This thesis addressed the problem of activity detection and recognition, presenting several methods to solve these problems.

In Chapter 3, we discussed the problem of activity detection in videos recorded by a moving camera and proposed a representation scheme, which define SoA accuracy rates. For accurate representation, we initially used a motion compensation scheme which was based on a quadratic motion model in order to extract human motion from the global motion pattern. Compensated Activity Areas (cAA) helped diminish the computational cost and improve recognition rates, while, a hybrid descriptor, which combined local (HOGHOF) with holistic (trajectory) attributes, led to improved recognition rates when combined with the Fisher encoding scheme.

In Chapter 4 a framework was designed for ADL recognition with a realistic computational cost. To achieve this, we introduced a novel activity mask, referred to as Motion Boundary Activity Area (MBAA), which extracted interest points around human boundaries and significantly decreased the computational cost without compromising the recognition accuracy rate. Next, optimal trajectories were proposed in order to deal with temporal variations in motion patterns. We introduced a novel algorithm, referred to as Statistical Sequential Change Detection (SSCD), that detects changes in the HOF descriptors, and consequently the motion patterns, in nearly real time. Numerous
experiments showed the validity of this method by its application to benchmark and in-house recorded datasets.

Chapter 5 presents novel methods for detecting and recognising ADLs simultaneously, at a near-real time computational cost. MBAAs and HOGHOF+Traj were also used here in order to describe the actions, while two algorithms were proposed for activity detection, i.e. ADL localisation in unsegmented video samples. Sequential Statistical Boundary Detection (SSBD) provided similar accuracy rates to the sliding window method used in the literature, but at a lower computational cost, rendering it a more appropriate solution for realistic scenarios.

6.2 Future work

In future work, we plan to improve the motion compensation scheme that is used in Chapter 3 with a novel RANSAC-based motion model. Our goal is to replace the binary mask which separates human from camera motion by a more sophisticated one that will segment background in several “super-planes” and produce separate motion model for each one. These larger areas are approximately planar and are connected via a projective transformation (expressed by the homography matrix) between different frames. The technique will use superpixel technology (i.e. SLICO) on colour images in order to segment image in similar colour cues and merge them in larger areas based on the motion flow that they follow. Accurate motion flow can be provided by either tracking densely sampled interest points or by matching SIFT descriptors and then using RANSAC homography for outlier extraction.

A motion saliency algorithm is currently designed in order to index the objects that move within the video scene. Several Bag-of-Visual-Words (BoVW), one for each saliency region, can afterwards be used in order to encode action descriptors in a more sophisticated way and tackle the absence of geometric properties that is observed in the BoVW encoding schemes.

Finally, we plan to use depth maps from Kinect cameras and decrease even more the computational cost that is required for ADL detection and recognition, so that we can achieve real time processing. In this scenario, we will work on subtracting human subject from the background using depth map analysis and create regions of interest in the body parts using superpixel technology. By providing a skeleton for each visible body part, we plan to use histograms of depth gradients in conjunction with skeleton correspondences to describe the desired activity.
Appendix A

Original publications and references

This thesis has been partially published in the following papers:

- **Journals**

- **Books**

- **Conferences**


15. K. Avgerinakis, A. Briassouli, I. Kompatsiaris, “Video processing for judicial applications”, ict4justice, September 2009 publication, Skopje, FYROM.
Appendix B

Clustering algorithms for Visual Vocabulary

Local based approaches for activity representation demand to quantize the activity feature space and create visual vocabularies (i.e. K-Means cluster centers, GMM means) in order to build fixed size action descriptors. Here we present the most common ones: KMeans and GMM.

B.1 K-means clustering

K-means is the most common and simple technique used for constructing visual vocabularies [49], [12], [21], [61], [20]. Its most popular implementation is based on least squares quantization [104], but is computationally cumbersome. Although improved versions [105] have succeeded in minimizing its high computational cost, K-means in [105] still requires storage space proportional to the square of the number of cluster centers, making it impractical for many cluster centres.

Given a set of feature vectors: $X = \{\bar{x}_1, \bar{x}_2, ..., \bar{x}_L\}$, where $\bar{x}_i \in R^D, i = \{1, 2, ..., L\}$ derived from the training set videos, K-Means looks for the $K$ optimal cluster centers $CC = \{\bar{c}_1, \bar{c}_2, ..., \bar{c}_K\}, \bar{c}_l \in R^D, l = \{1, 2, ..., K\}$ that partition the feature space in the best possible way. For an initial set of K cluster centers $\{\bar{c}^1_1, \bar{c}^1_2, ..., \bar{c}^1_K\}$ at iteration $t = 1$, the method of [104] alternates between assignment and update steps. In the assignment step, at iteration $t$, each observation $\bar{x}_i, i \in \{1, ..., L\}$ is assigned to its closest cluster center $\bar{c}^1_k, k \in \{1, ..., K\}$, which yields the lowest within cluster sum of squares (WCSS):

$$k = \arg \min_{j \in \{1,2,\ldots,K\}} \|\bar{x}_i - \bar{c}^1_j\|^2,$$  \hspace{1cm} (B.1)
so the set \( S^t_k \) of observations in each cluster center \( c^t_k \) at iteration \( t \) is:

\[
S^t_k = \{ \bar{x}_i : \| \bar{x}_i - c^t_k \|_2^2 \leq \| \bar{x}_i - c^t_j \|_2^2, \ 1 \leq j \leq K, j \neq k \}
\]

At the update step, new cluster centers \( \bar{c}^{t+1}_k \), \( k \in \{1,...K\} \) are computed after each new assignment as follows:

\[
\bar{c}^{t+1}_k = \frac{1}{|S^t_k|} \sum_{\bar{x}_i \in S^t_k} \bar{x}_i
\]

### B.2 Gaussian Mixture Model (GMM) clustering

Gaussian Mixture Model (GMM) clustering can also be used for quantizing the visual feature space. Unlike K-means, GMM use has not been as widespread in the computer vision community for partitioning visual feature spaces, although recent results in image classification are very promising [46], [37].

For GMM clustering, a set of \( L \) feature vectors \( X = \{ \bar{x}_1, \bar{x}_2, ..., \bar{x}_L \} \), where \( \bar{x}_i \in R^D, i = \{1,2,...,L\} \) derived from training videos is modeled by a mixture of \( m \) Gaussian distributions that are completely specified by the set of parameters \( \Theta \). \( \Theta = \{\pi_1, \mu_1, \Sigma_1, ..., \pi_m, \mu_m, \Sigma_m\} \) comprises of prior probabilities \( \pi_l \in R_+ \), mean values \( \mu_l \in R^D \) and covariance matrices \( \Sigma_l \in R^{DxD} \), where \( l = \{1,2,...,m\} \). A sample \( \bar{x}_i, i = \{1,2,...,L\} \), derived from the \( X \) set of feature vectors, is characterized by its density \( p(\bar{x}_i;\Theta) \):

\[
p(\bar{x}_i|\pi_l, \mu_l, \Sigma_l) = \sum_{l=1}^{m} p_l(\bar{x}_i) \cdot \pi_l, \pi_l \geq 0, \quad \sum_{l=1}^{m} \pi_l = 1,
\]

\[
p_l(\bar{x}_i) = \frac{1}{\sqrt{(2\pi)^D\sqrt{\det\Sigma_l}}} \exp\left\{-\frac{1}{2}(\bar{x}_i - \mu_l)^T\Sigma_l^{-1}(\bar{x}_i - \mu_l)\right\},
\]

where \( p_l, l = \{1,2,...,m\} \) is a probability density function (pdf) with parameters \( \{\pi_1, \mu_1, \Sigma_1, ..., \pi_m, \mu_m, \Sigma_m\} \), prior probabilities \( \pi_l \in R_+ \), mean values \( \mu_l \in R^D \) and covariance matrices \( \Sigma_l \in R^{DxD} \) for \( m \) Gaussian components. We make the assumption that the data is uncorrelated, leading to diagonal covariance matrices, so the GMM is fully described by \((2D + 1)m\) scalar parameters. A GMM is fit to the data \( X = \{\bar{x}_1, ..., \bar{x}_n\} \) by maximizing the data log-likelihood:

\[
L(\Theta; X) = \ln p(X; \Theta) = \frac{1}{L} \sum_{i=1}^{L} \ln \left( \sum_{l=1}^{m} \pi_l p(\bar{x}_i|\mu_l, \Sigma_l) \right)
\]
Expectation Maximization (EM) is then initialized to learn these parameters. At the assignment step of the algorithm, soft data \( i = \{1, 2, ..., L\} \) to distribution \( k = \{1, 2, ..., m\} \) assignments are defined as follows:

\[
q_{ik} = \frac{p_k(\bar{x}_i) \cdot \pi_k}{\sum_{l=1}^{m} p(\bar{x}_i | \mu_l, \Sigma_l) \cdot \pi_l}
\]

At the update step (Maximization step or M-step), the mixture parameter estimates are refined using the computed probabilities:

\[
\pi_k = \frac{1}{L} \sum_{i=1}^{L} q_{ik}, \quad \mu_k = \frac{\sum_{i=1}^{L} q_{ik} \bar{x}_i}{\sum_{i=1}^{L} q_{ik}}
\]

\[
\Sigma_k = \frac{\sum_{i=1}^{L} q_{ik} (\bar{x}_i - \bar{\mu}_k)(\bar{x}_i - \bar{\mu}_k)^T}{\sum_{i=1}^{L} q_{ik}}, \quad k = \{1, 2, ..., m\}
\]
Bibliography


