Human action recognition in videos

Piotr Tadeusz Biliński

UNIVERSITY OF NICE - SOPHIA ANTIPOLIS

DOCTORAL SCHOOL STIC

SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

P H D T H E S I S

to obtain the title of

PhD of Science

of the University of Nice - Sophia Antipolis

Specialty : COMPUTER SCIENCE

Defended by

Piotr Tadeusz BILIŃSKI

Human Action Recognition in Videos

Thesis Advisor: François BRÉMOND

prepared at INRIA Sophia-Antipolis, STARS Team

defended on December 5, 2014

Jury :

President :	Frédéric PRECIOSO	-	Pr., Polytech Nice - Sophia Antipolis, France
Reviewers :	Ram NEVATIA	-	Pr., University of Southern California, USA
	Frédéric JURIE	-	Pr., University of Caen, France
Examinator :	Ivan LAPTEV	-	DR, INRIA Paris, France
Advisor :	François BRÉMOND	-	DR, INRIA Sophia - Antipolis, France

UNIVERSITE DE NICE-SOPHIA ANTIPOLIS

ECOLE DOCTORALE STIC

SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

T H E S E

pour l'obtention du grade de

Docteur en Sciences

de l'Université de Nice - Sophia Antipolis

Mention : INFORMATIQUE

presentée et soutenue par

Piotr Tadeusz BILIŃSKI

Reconnaissance d'action humaine dans des vidéos

Thèse dirigée par François BRÉMOND

INRIA Sophia-Antipolis, STARS

soutenue le 05/12/2014

Jury :

Président :	Frédéric PRECIOSO	-	Pr., Polytech Nice - Sophia Antipolis, France
Rapporteurs :	Ram NEVATIA	-	Pr., University of Southern California, USA
	Frédéric JURIE	-	Pr., University of Caen, France
Examinateur :	Ivan LAPTEV	-	DR, INRIA Paris, France
Directeur de thèse :	François BRÉMOND	-	DR, INRIA Sophia - Antipolis, France

Human action recognition in videos

Abstract: This thesis targets the automatic recognition of human actions in videos. Human action recognition is defined as a requirement to determine what human actions occur in videos. This problem is particularly hard due to enormous variations in visual and motion appearance of people and actions, camera viewpoint changes, moving background, occlusions, noise, and enormous amount of video data.

Firstly, we review, evaluate, and compare the most popular and the most prominent state-of-the-art techniques, and we propose our action recognition framework based on local features, which we use throughout this thesis work embedding the novel algorithms. Moreover, we introduce a new dataset (CHU Nice Hospital) with daily self care actions of elder patients in a hospital.

Then, we propose two local spatio-temporal descriptors for action recognition in videos. The first descriptor is based on a covariance matrix representation, and it models linear relations between low-level features. The second descriptor is based on a Brownian covariance, and it models all kinds of possible relations between low-level features.

Then, we propose three higher-level feature representations to go beyond the limitations of the local feature encoding techniques.

The first representation is based on the idea of relative dense trajectories. We propose an object-centric local feature representation of motion trajectories, which allows to use the spatial information by a local feature encoding technique.

The second representation encodes relations among local features as pairwise features. The main idea is to capture the appearance relations among features (both visual and motion), and use geometric information to describe how these appearance relations are mutually arranged in the spatio-temporal space.

The third representation captures statistics of pairwise co-occurring visual words within multi-scale feature-centric neighbourhoods. The proposed contextual features based representation encodes information about local density of features, local pairwise relations among the features, and spatio-temporal order among features.

Finally, we show that the proposed techniques obtain better or similar performance in comparison to the state-of-the-art on various, real, and challenging human action recognition datasets (Weizmann, KTH, URADL, MSR Daily Activity 3D, HMDB51, and CHU Nice Hospital).

Reconnaissance d'action humaine dans des vidéos

Résumé: Cette thèse traite de la reconnaissance automatique daction humaine dans des vidéos. La reconnaissance d'action humaine est indispensable pour déterminer quelles actions humaines se produisent dans des vidéos. Ce problème est particulièrement difficile en raison d'énormes variations dans les aspects visuels et de mouvement des personnes et des actions, les changements de point de vue de la caméra, le fond mobile, des occlusions, la présence de bruit, ainsi que l'énorme quantité de données vidéos.

Tout d'abord, nous passons en revue, évaluons et comparons les techniques les plus importantes et les plus populaires de l'état de l'art pour la reconnaissance d'action, ensuite, nous proposons une plateforme basée sur des caractéristiques locales, que nous utilisons tout au long de ce travail de thèse pour étudier de nouveaux algorithmes. En plus, nous introduisons une nouvelle base de données (Hôpital CHU de Nice) avec des actions de la vie quotidienne de patients âgés dans cet hôpital.

Ensuite, nous proposons deux descripteurs spatio-temporels locaux pour la reconnaissance d'action dans les vidéos. Le premier descripteur est basé sur une représentation des matrices de covariance, modélisant les relations linéaires entre les caractéristiques bas niveaux. Le deuxième descripteur est basé sur les covariances browniennes, et modélise tous les types de relations possibles entre les caractéristiques bas niveaux.

Après, nous proposons trois représentations de caractéristiques de hauts niveaux pour dépasser les limites des techniques utilisant l'encodage des sacs de mots.

La première représentation est basée sur le principe des trajectoires relatives denses. Nous proposons une représentation objet-centrée des caractéristiques locales des trajectoires de mouvement, ce qui permet d'utiliser l'information spatiale par une technique de codage des caractéristiques locales.

La deuxième représentation encode les relations entre les caractéristiques locales par paires. Le principe est dextraire les relations d'apparence entre les caractéristiques (à la fois visuelles et de mouvement), et dutiliser l'information géométrique pour décrire la façon dont ces relations d'apparence sont disposées mutuellement dans l'espace spatio-temporel.

La troisième représentation calcule les statistiques des paires concomitantes des mots visuels dans les voisinages multi-échelles centrées les caractéristiques. La représentation basée sur les caractéristiques contextuelles proposées encode linformation sur la densité locale de ces caractéristiques, les relations entre les paires des caractéristiques locales et leur ordre spatio-temporel.

Finalement, les techniques proposées permettent d'obtenir une performance meilleure ou semblable par rapport à l'état de l'art, sur des bases de données représentant une grande diversité dactions humaines (Weizmann, KTH, URADL, MSR Daily Activity 3D, HMDB51, et Hôpital CHU de Nice).

Acknowledgments

Contents

1 Introduction **1**
- 1.1 Introduction . . . 1
- 1.2 Motivation . . . 3
- 1.3 Taxonomy and Problem Statement . . . 6
- 1.4 Research Challenges . . . 8
- 1.5 Main Contributions . . . 11
- 1.6 Thesis Structure . . . 14

2 Related Work **17**
- 2.1 Human Body Model Based Methods . . . 18
- 2.2 Holistic Methods . . . 21
- 2.3 Local Feature Methods . . . 23
  - 2.3.1 Local Features . . . 23
    - 2.3.1.1 Local Feature Detectors . . . 23
    - 2.3.1.2 Local Feature Descriptors . . . 28
  - 2.3.2 Collections of Local Features . . . 30
    - 2.3.2.1 Pairwise Features . . . 30
    - 2.3.2.2 Contextual Features . . . 33
  - 2.3.3 Local Features Encoding . . . 36
- 2.4 Classifiers . . . 39
- 2.5 Conclusion . . . 43

3 Action Recognition Framework, Its Evaluation, and Analysis **47**
- 3.1 Introduction . . . 48
- 3.2 Action Recognition Framework . . . 48
  - 3.2.1 Local Spatio-Temporal Video Features . . . 49
    - 3.2.1.1 Spatio-Temporal Interest Points . . . 50
    - 3.2.1.2 Dense Trajectories . . . 53
  - 3.2.2 Video-Action Representation . . . 55
    - 3.2.2.1 Bag-of-Features . . . 56
    - 3.2.2.2 Fisher Vectors . . . 58
  - 3.2.3 Video-Action Recognition . . . 59
    - 3.2.3.1 Linear Support Vector Machines . . . 60
    - 3.2.3.2 Non-Linear Support Vector Machines . . . 61
    - 3.2.3.3 Fast Exponential χ^2 Kernel . . . 62
    - 3.2.3.4 From Binary to Multi-Class Classifcation . . . 63
  - 3.2.4 Action Recognition Approach Assessment . . . 64
    - 3.2.4.1 Cross-Validation . . . 64
    - 3.2.4.2 Dataset Splits . . . 64
    - 3.2.4.3 Mean Class Accuracy Metric . . . 65

3.3 Datasets . . . 65
3.3.1 Weizmann Dataset . . . 66
3.3.2 KTH Dataset . . . 67
3.3.3 URADL Dataset . . . 69
3.3.4 MSR Daily Activity 3D Dataset . . . 70
3.3.5 HMDB51 Dataset . . . 73
3.3.6 CHU Nice Hospital Dataset . . . 76
3.3.7 Datasets Summary . . . 79
3.4 Experiments, Comparison, and Analysis . . . 80
3.4.1 Spatio-Temporal Interest Points . . . 81
3.4.1.1 Weizmann Dataset . . . 81
3.4.1.2 URADL Dataset . . . 83
3.4.1.3 MSR Daily Activity 3D Dataset . . . 83
3.4.1.4 HMDB51 Dataset . . . 83
3.4.1.5 Overall . . . 87
3.4.1.6 Results Summary and Analysis . . . 87
3.4.2 Dense Trajectories . . . 89
3.4.2.1 Weizmann Dataset . . . 89
3.4.2.2 URADL Dataset . . . 92
3.4.2.3 MSR Daily Activity 3D Dataset . . . 92
3.4.2.4 HMDB51 Dataset . . . 92
3.4.2.5 Overall . . . 95
3.4.2.6 Results Summary and Analysis . . . 95
3.4.3 Summary and Conclusion . . . 96
3.5 Conclusion . . . 97

4 Video Covariance Matrix Logarithm 99
4.1 Introduction . . . 100
4.2 Video Covariance Matrix Logarithm Descriptor . . . 103
4.2.1 Video Frame Descriptor . . . 103
4.2.2 Pixel-Level Features . . . 105
4.2.3 Video Volume Descriptor . . . 108
4.2.4 Riemannian Geometry . . . 109
4.2.4.1 Affine-Invariant Riemannian Metric . . . 109
4.2.4.2 Log-Euclidean Riemannian Metric . . . 110
4.2.4.3 Riemannian Metric Selection . . . 110
4.2.4.4 Matrix Logarithm . . . 111
4.2.5 Fast Covariance Matrix Calculation . . . 112
4.3 Approach Overview . . . 114
4.4 Experiments . . . 116
4.4.1 Weizmann Dataset . . . 117
4.4.2 URADL Dataset . . . 119
4.4.3 MSR Daily Activity 3D Dataset . . . 123
4.4.4 HMDB51 Dataset . . . 123

4.4.5 Results Summary and Analysis . . . 126
4.5 Conclusion . . . 126

5 Video Brownian Covariance **129**
5.1 Introduction . . . 130
5.2 Video Brownian Covariance Descriptor . . . 131
5.2.1 Brownian Covariance . . . 132
5.2.1.1 Distance Covariance $\mathcal{V}^2$ . . . 132
5.2.1.2 Sample Distance Covariance $\mathcal{V}_n^2$ . . . 132
5.2.1.3 Distance Correlation $\mathcal{R}_n^2$ . . . 133
5.2.2 Video Frame Descriptor . . . 135
5.2.3 Pixel-Level Features . . . 136
5.2.4 Video Brownian Covariance . . . 137
5.2.5 Normalization . . . 137
5.2.6 Dimension Reduction . . . 138
5.3 Approach Overview . . . 138
5.4 Experiments . . . 139
5.4.1 Descriptor Evaluation . . . 140
5.4.1.1 Weizmann Dataset . . . 140
5.4.1.2 URADL Dataset . . . 141
5.4.1.3 MSR Daily Activity 3D Dataset . . . 142
5.4.1.4 Results summary and analysis . . . 143
5.4.2 Approach Evaluation . . . 144
5.4.2.1 Weizmann Dataset . . . 144
5.4.2.2 URADL Dataset . . . 145
5.4.2.3 MSR Daily Activity 3D Dataset . . . 147
5.4.2.4 HMDB51 Dataset . . . 148
5.4.2.5 Results Summary and Analysis . . . 149
5.5 Conclusion . . . 149

6 Relative Trajectories **151**
6.1 Introduction . . . 152
6.2 Relative Trajectories . . . 154
6.2.1 Trajectory Extraction . . . 154
6.2.2 Trajectory Shape and Relative Trajectory Shape Descriptors . . . 155
6.2.2.1 Trajectory Shape Descriptor . . . 155
6.2.2.2 Relative Trajectory Shape Descriptor . . . 156
6.2.3 Dynamic Coordinate System . . . 157
6.2.4 Trajectory Filtering . . . 159
6.3 Approach Overview . . . 161
6.4 Experiments . . . 162
6.4.1 URADL Dataset . . . 163
6.4.2 MSR Daily Activity 3D Dataset . . . 165
6.4.3 KTH Dataset . . . 169

6.4.4 CHU Nice Hospital Dataset . . . 170
6.4.5 Results Summary and Analysis . . . 171
6.5 Conclusion . . . 171

7 Geometric and Appearance Relations of Pairwise Features 173
7.1 Introduction . . . 174
7.2 Geometric and Appearance Relations of Pairwise Features . . . 176
7.2.1 Local Feature Extraction . . . 176
7.2.2 Pairwise Features . . . 176
7.2.3 Geometric and Appearance Relations of Pairwise Features . . . 177
7.3 Approach Overview . . . 178
7.4 Experiments . . . 180
7.4.1 Implementation Details . . . 181
7.4.2 KTH Dataset . . . 181
7.4.3 URADL Dataset . . . 182
7.4.4 HMDB51 Dataset . . . 182
7.4.5 Results Summary and Analysis . . . 183
7.5 Conclusion . . . 183

8 Spatio-Temporal Ordered Contextual Features 185
8.1 Introduction . . . 186
8.2 Spatio-Temporal Ordered Contextual Features . . . 187
8.2.1 Local Feature Quantization . . . 187
8.2.2 Feature-Centric Neighbourhood . . . 188
8.2.3 Spatio-Temporal Ordered Contextual Features . . . 189
8.3 Approach Overview . . . 191
8.4 Experiments . . . 192
8.4.1 Implementation Details . . . 193
8.4.2 KTH Dataset . . . 193
8.4.3 URADL Dataset . . . 194
8.4.4 Results Summary and Analysis . . . 195
8.5 Conclusion . . . 195

9 Comparison of Approaches 197
9.1 Spatio-Temporal Appearance Descriptors . . . 198
9.2 Trajectory Shape Descriptors . . . 199
9.3 Pairwise and Contextual Features . . . 200
9.4 Comparison with State-of-The-Art . . . 201
9.4.1 Weizmann Dataset . . . 201
9.4.2 KTH Dataset . . . 201
9.4.3 URADL Dataset . . . 204
9.4.4 MSR Daily Activity 3D Dataset . . . 204
9.4.5 HMDB51 Dataset . . . 205
9.5 Conclusion . . . 207

10 Conclusion and Perspectives **209**
10.1 Key Contributions . . . 209
10.2 Limitations . . . 210
10.3 Future Work . . . 211
10.3.1 Short-Term Perspectives . . . 211
10.3.2 Long-Term Perspectives . . . 212

A Video Covariance Matrix Logarithm: Additional Experiments **215**

B CHU Nice Hospital Dataset: Sample Video Frames **221**

Bibliography **227**

CHAPTER 1

Introduction

Contents

1.1 Introduction 1
1.2 Motivation 3
1.3 Taxonomy and Problem Statement 6
1.4 Research Challenges 8
1.5 Main Contributions 11
1.6 Thesis Structure 14

In this chapter, we introduce the topic of this PhD thesis, action recognition in videos (Section 1.1). We present the motivation of our work (Section 1.2), the taxonomy, which is used throughout this work, and the problem statement (Section 1.3). Then, we describe research challenges related to action recognition (Section 1.4), and we present our main contributions in this topic (Section 1.5). Finally, we close this chapter with the thesis structure (Section 1.6).

1.1 Introduction

Figure 1.1 – An early example of the phrase "One Look is Worth A Thousand Words" [Spe 1911].

90% of information that comes to our brain is visual, and our brain processes visual information 60000 times faster than text [1].

There is a popular adage "A picture is worth a thousand words", which refers to the notion that a complex idea can be explained using a single image. Many similar expressions share the same opinion about the importance of visual information, *e.g.* "One Look Is Worth A Thousand Words" [Spe 1911] (see Figure 1.1) and "Use a picture. It's worth a thousand words" [One 1913].

[1] http://visualteachingalliance.com/

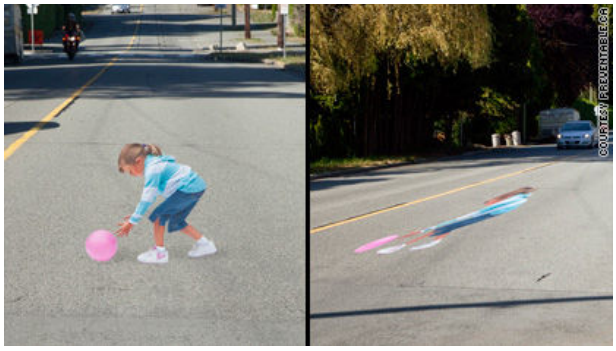


Figure 1.2 – The importance of videos in scene understanding. A single (left) image might mislead a viewer about the real scene, *i.e*. a painting on the street (see the right image). The images presents the 3D optical illusion of a young girl chasing a ball across the street, in Vancouver, Canada. The 2D painting becomes a 3D illusion as drivers approach it, to draw attention to the risk of children running into the street, and to make drivers slow down.

While a picture says a thousand words, there is another popular adage "One minute of a video is worth 1.8 million words", which is widely attributed to Dr. James McQuivery of Forrest Research. Usually, to understand the scene we need a video; a single image usually gives only a general information, and it might even outright mislead the viewer (see Figure 1.2). Therefore, over the years people use videos, *i.e*. moving visual images, more and more often to convey information, as even a short video contains a wealth of information.

Nowadays, no one imagines the world without videos. They are a popular, accessible, frequent, natural, and meaningful form of conveying information.

Clearly, videos as well as video cameras have become an inseparable part of our lives. Video cameras are used almost everywhere. Cities, workplaces, homes, schools, hospitals, banks, shops, and many other (especially public) places widely introduce video surveillance cameras. This is becoming normal and widely accepted by society.

Computers, laptops, tablets, video cameras, and even mobile phones, they all can record, produce, store, and share videos. With every minute and every second, video devices are becoming more and more available in our lives. They are evolving rapidly, they provide good quality videos (often high-definition), and they are relatively cheap.

With a growing number of available videos, greater and easier access to them, the need for their understanding increases as well. From year to year the understanding of videos is getting more and more interest. The limited human capabilities of analyzing them in a natural way created a need for intelligent systems, that could analyze and recognize activities occurring in videos. With cheap, fast, and rapidly evolving computers, which are more and more affordable and powerful, such systems are already possible, and they can

perform video understanding and action recognition much faster than any human.

In this thesis, we address the problem of recognizing human actions in videos. Our goal is to automatically learn actions from training videos, and recognize them in unseen, diverse, and realistic video settings.

1.2 Motivation

There are a great many potential applications of action recognition systems. Action recognition plays a key role in many domains and applications, such as: Video Retrieval, Video Surveillance, Health Care, Human-Computer Interaction and Entertainment Industry.

Video Retrieval: Consumer Videos and Movies

Nowadays, creating, uploading and sharing videos have all become very popular, mostly due to: commonly available equipment that can generate videos, high-speed Internet access, and free storage servers. The number of consumer videos on the Internet is rapidly growing with every second. According to the most popular video-sharing website, *i.e.* YouTube [2]:

- 100 hours of video are uploaded to YouTube every minute.
- There are more than 1 billion unique users that visit this website each month.
- The users watch over 6 billion hours of videos each month, what is almost an hour for every person on the Earth.

This shows how significant in our lives the video content is and how important automatic video analysis and understanding remains, so as to properly retrieve a video relevant to the user.

Most of the YouTube content has been uploaded by individuals, and this is why we can find there a lot of consumer and amateur videos. A lot of professional videos may be found on the Internet as well, and they are viewed every day. For example, according to the statistics collected from and by YouTube:

- Batman movie is very popular, with more than 3 billion views of 71000 hours of video.
- The top 10 movies of superheroes account for more than 10 billion views and 234000 hours of video.

With huge amount of video data uploaded on the Internet with every second, and with such widespread popularity of watching videos and movies on the Internet, video understanding and action recognition systems have become extremely important and necessary for relevant video retrieval.

[2] http://www.youtube.com/yt/press/statistics.html [21 November 2013]

Video Surveillance

Figure 1.3 – Various video surveillance cameras often seen in the streets, and a sample center for monitoring video surveillance cameras.

Video surveillance cameras are part of our lives. Video surveillance is the process of monitoring people and objects of interest using video cameras. With each passing day, a growing number of people have daily contact with video surveillance systems. They became normal and widely accepted by society. Video surveillance cameras are used almost everywhere (see Figure 1.3): at airports, subways, train stations, bus terminals, shopping malls, banks, post offices, casinos, swimming pools, cinemas, parking lots, but also in the streets for traffic monitoring, and in a great many buildings, for intrusion detection and analysis of human behavior. The number of possible examples of applications is enormous.

Today's increase in threats to the security in cities and towns around the world makes the use of video cameras to monitor people necessary. The attacks on humans, thefts, fights, vandalism, and harassment are just some cases where action recognition systems are needed. Transmission of high quality audio and video streams allows police forces and security operatives to have eyes and ears everywhere. Video surveillance cameras are more and more often used not only to detect security violations, but also to prevent them.

With the rapid increase in the number of installed video surveillance cameras in cities and towns (*e.g.* currently around 500000 security cameras in London and around 4 million in Great Britain [3]; around 800000 security cameras in Beijing and around 30 million in China [4]), the demand for action recognition systems increases as well.

[3] http://www.slate.com/articles/news_and_politics/explainer/2010/05/big_apple_is_watching_you.html

[4] http://thediplomat.com/2013/11/the-limits-of-chinas-surveillance-state/

A Social Issue and Health Care

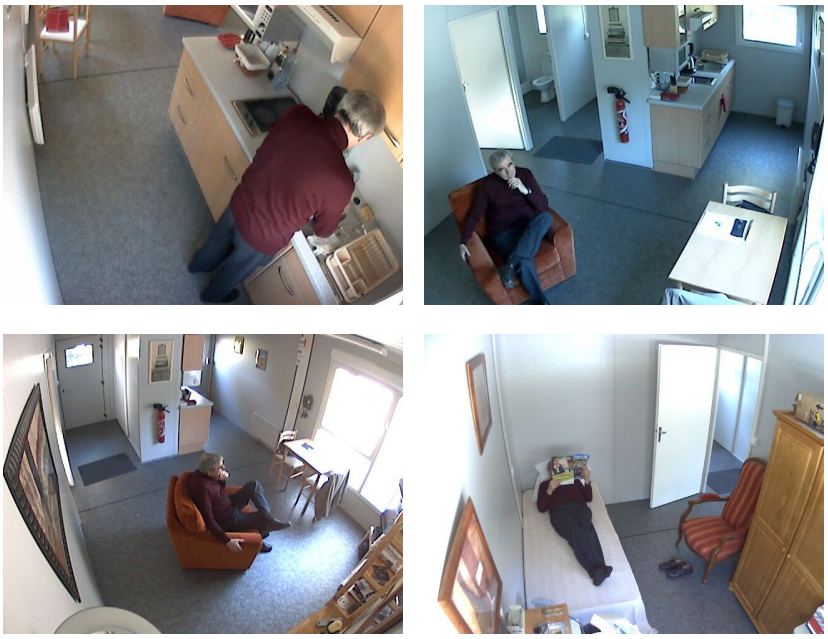


Figure 1.4 – Sample video frames acquired by four video surveillance cameras monitoring people at home (Gerhome Laboratory in France).

According to the Global AgeWatch [5], around the world there were 810 million people aged 60 or more in 2012. It is projected that in fewer than 10 years this number will reach 1 billion, and 1.375 billion by 2030, what means that 16% of the global population will be people aged 60 and more. There will be more people aged 60+ than children under 10. The proportion of older people is growing faster than the general population. Population aging is occurring, and is occurring worldwide.

Therefore, the recognition of human actions and behavior is becoming more and more important and frequently used in medicine, especially for the purpose of health care monitoring of elderly people. There is a rapidly increasing demand for systems that allow to recognize human actions, and early detect and emerge about upcoming and existing physical and mental health problems of patients. Identifying changes in human everyday behavior such as food preparation, walking, housekeeping, exercise or sleeping, allows medical scientists to propose strategies related to diet, exercise and medication adherence. This is especially important for elderly people, for whom such systems allow to live at home longer, healthier and safer.

Sample video frames acquired by four video surveillance cameras monitoring people at home (Gerhome Laboratory [6] in France) are presented in Figure 1.4.

[5] http://www.helpage.org/download/52e7ed72b4a68
[6] http://gerhome.cstb.fr/

Human-Computer Interaction and Entertainment Industry

Figure 1.5 – Microsoft Kinect sensor (left image) and a sample image presenting a human-computer interaction (right image).

Nowadays, video cameras are used not only for video surveillance, and not only in mobile phones and laptops, but they are also used for human-computer interaction and in the entertainment industry. Video cameras provide a natural and intuitive way of human communication with a device. They have become so cheap that with each passing day, a rapidly growing number of people have daily contact with them. An example of popular video camera is the Microsoft Kinect sensor (see Figure 1.5), which is available in millions of homes around the world. Just since November 2010, more than 24 million Kinect sensors have been sold worldwide [7]. Kinect is a motion sensing input device that allows users to control and interact with it through a natural user interface, using gestures and spoken commands. Therefore, one of the most important aspects for this sensor is the recognition of gestures and short actions.

1.3 Taxonomy and Problem Statement

Action recognition can be performed at different levels of abstraction. Over the last years, various taxonomies have been proposed to define these levels of abstraction, and different names for them have been used interchangeably.

In this dissertation we use a common terminology, which was described by Moeslund *et al*. [Moeslund 2006]. Depending on the complexity, we categorize a motion as:

- **Action primitive** - it is an atomic, elementary movement of a person's body part, such as "raising a hand" and "stretching an arm". Action primitive is also often referred to as **gesture**, and sometimes is also referred to as **concept**.

- **Action** - it is a more complex body movement that consists of several action primitives (gestures), which are organized temporally, such as "kicking a ball" and "running".

[7] http://www.microsoft.com/en-us/news/bythenumbers/index.html

Figure 1.6 – Various actions and various video settings.

- **Activity** - it is even more complex body movement that consists of a number of subsequent actions, such as "playing football" and "baking a cake". Activity is also often referred to as **event**.

Within this work we also use the two following terms:

- **human-object interaction** - we use this term to highlight that a motion is intrinsically linked to some objects, *e.g*. "human-cup interaction" during the "drinking a tea" action.
- **human-human interaction** - we use this term to highlight that a motion is intrinsically linked to at least two people, *e.g*. "discussion" requires at least two people.

In this dissertation we address the issue of automatic recognition of human actions in videos via supervised learning. This means that we have available training videos with annotations, *i.e*. for every training video we know which action or actions it contains. Using training videos we learn action models, and then we try to recognize these actions in new, unseen videos, *i.e*. videos for which we do not have annotations.

We consider various types of actions and videos, *e.g*. surveillance videos, movies, and consumer videos (*e.g*. like YouTube videos). Figure 1.6 illustrates various types of actions

and videos that we are interested in and we try to recognize. We focus on the recognition of actions. However, some actions that we recognize during our experiments may also be classified as gestures and simple activities, as sometimes boundaries between the motion categories are difficult to set and the names are used interchangeably in the state-of-the-art datasets that we use.

Several state-of-the-art techniques divide the activity (sometimes also called action) recognition problem to the detection of concepts, and then these approaches recognize activities using the detected concepts, *e.g.* [Yang 2012, Izadinia 2012]. In this thesis work, we do not consider such types of approaches as: (1) the detection of concepts usually requires additional annotations for the training dataset and we assume that we do not have them, and (2) we focus on action recognition and not activity recognition (*i.e.* we consider our actions as relatively short and atomic), and some actions - videos in our datasets are too short to divide them into concepts (*i.e.* they last just a few seconds).

1.4 Research Challenges

Human action recognition is a very important and challenging research topic. One of the main issues with action recognition is that the same action can be performed in many different ways, even by the same person. Thus, the main challenge is to find a proper representation of actions, which is both discriminative (so we can separate different actions) and general (so we can classify together videos with the same actions, performed by different looking and moving people). Due to enormous variations in visual and motion appearance of both people and actions, camera view point, occlusions, noise, and enormous amount of video data, action recognition still remains a challenging problem.

Action recognition based on local features is one of the most active research topics. The main advantage of local feature based methods is that no information on human body model or localization of people is required, in contrast to other techniques (*e.g.* human body model based methods and holistic methods, which are described in Chapter 2). Therefore, in this dissertation we mainly focus on local feature based methods.

Over the last years many techniques have been proposed for action recognition. Therefore:

- Our first goal is to review and compare the existing local feature based methods in order to decide which local features and which video representations work the best.
- Our second goal is to understand the limitations of the existing techniques and to propose new techniques for action recognition in videos to go beyond the state-of-the-art limitations.

Existing local feature based methods

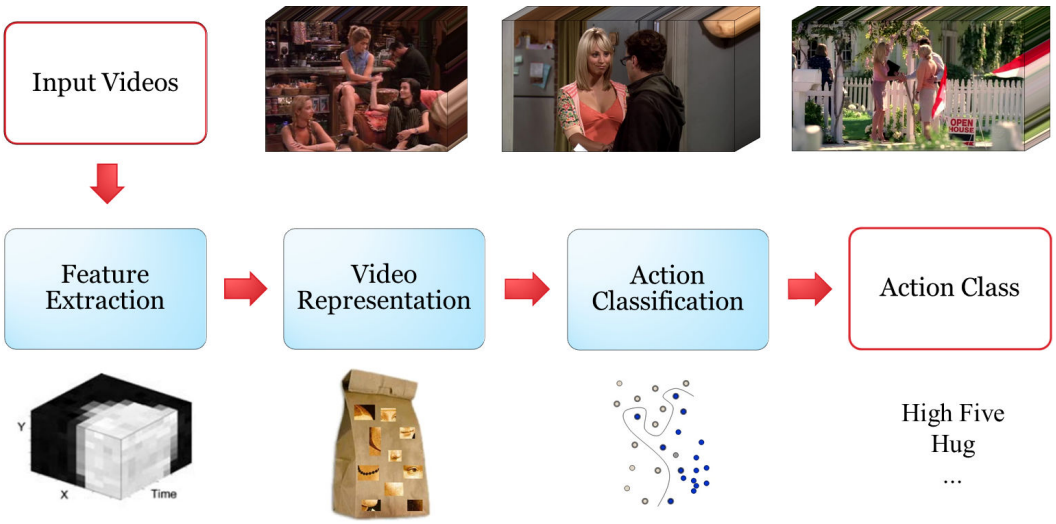


Figure 1.7 – Overview of a typical action recognition / video classification approach.

Our first goal is to review and compare the existing local feature based methods in order to decide which local features and which video representations work the best.

Typical action recognition approach based on local features consists of three phases: feature extraction, video representation, and action recognition, see Figure 1.7.

The most popular local spatio-temporal video features (see Section 2.3.1) are:

- Spatio-Temporal Interest Points [Laptev 2005]:
  - detector: Harris3D,
  - descriptors: Histogram of Oriented Gradients and Histogram of Optical Flow.
- Dense Trajectories [Wang 2011a]:
  - detector: dense sampling,
  - descriptors: Trajectory shape descriptor, Histogram of Oriented Gradients, Histogram of Optical Flow, and Motion Boundary Histograms.

The most popular video representation techniques (see Section 2.3.3) are:

- Bag-of-features approach - it is the most popular technique for encoding local features. Its representation requires a small amount of memory to store a video sequence. It represents a video using a histogram of quantized local features, followed by the normalization step (the L1 norm or the L2 norm).

- Fisher vector encoding - another encoding technique, which has shown superior results over the bag-of-features for many Computer Vision tasks. Its representation requires a large amount of memory to store a video sequence.

The existing literature provides a limited evaluation, comparison, and analysis of these techniques together, so our first goal is to to review and compare the Spatio-Temporal Interest Points and the Dense Trajectories with the bag-of-features approach and the Fisher vector encoding.

Moreover, an interesting issue with the bag-of-features approach is that many of the state-of-the-art techniques use the L1 normalization but so many use the L2. Therefore, we evaluate and compare the bag-of-features approach with the L1 and the L2 norm.

State-of-The-Art Limitations and Challenges

Our second goal is to understand the limitations of the existing techniques and to propose new techniques for action recognition in videos to go beyond the state-of-the-art limitations.

The most popular state-of-the-art descriptors for local features are: Histogram of Oriented Gradients, Histogram of Optical Flow, and Motion Boundary Histograms. All these descriptors are based on a 1-dimensional histogram representation of individual features, and they directly model values of given features. However, the joint statistics between individual features, *i.e.* low-level features such as intensity and gradient, are ignored by these descriptors, whereas such information may be informative. Therefore, **a new local feature descriptor for videos is required to capture relations between different low-level features**.

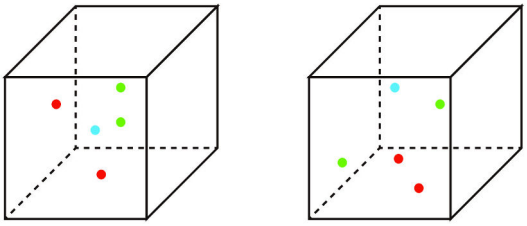


Figure 1.8 – Wang *et al.* [Wang 2011b]: Two spatio-temporal video volumes with local features. Different colors represent different visual words. Although these two volumes may express different actions, the bag-of-features and Fisher vector encoding consider these volumes to be identical.

Although the bag-of-features and the Fisher vector encoding are very successful models in many domains, including Computer Vision, they also contain limitations. One of the main limitations of these models is that they simplify the structure of spatio-temporal video data assuming conditional independence across spatial and temporal domains (see

Figure 1.8). They compute only global statistics of local features, ignoring information about the spatio-temporal positions of features, relations among the features, and local density of features. Thus, not using all the available information, they may fail to distinguish similar actions. A common way to overcome this limitation is to use either spatio-temporal grids [Laptev 2008] or multi-scale pyramids [Lazebnik 2006]. However, these methods are still limited in terms of a detailed description providing only a coarse representation. Recently, several representations of local features were proposed, which are often called pairwise features and contextual features.

- Pairwise Features - they encode relations among local spatio-temporal features (like spatio-temporal interest points and trajectories). They are very useful to distinguish videos with similar global distributions of local features, but with different placements and orders of local features. The existing pairwise features based techniques use the discriminative power of individual features and the capture visual relations among features. However, they typically ignore information about the spatio-temporal geometric relations between features (*i.e.* Δx, Δy, Δt). Moreover, some of the above techniques can only handle small codebooks, and they ignore associations among geometric and appearance relations among features.

- Contextual Features - they encode local distributions of features in feature-centric neighborhoods. They are very useful to distinguish videos with similar global distributions of local features, but with different local distributions of local features. The existing contextual features based techniques use the discriminative power of individual features, and they capture local densities of features in feature-centric neighborhoods. To capture structural information in feature-centric neighborhoods, they use the spatio-temporal grid; however, as mentioned before, the spatio-temporal grid is limited in terms of detailed description providing only a coarse representation. Moreover, the existing techniques ignore information about the spatio-temporal order among features.

Therefore, **new representations of local features are required to capture spatio-temporal positions of features, pairwise relations among the features, and local densities of features**.

1.5 Main Contributions

To go beyond the limitations of the state-of-the-art, we address the issues presented in the previous section, and we introduce the following contributions:

Action recognition framework, its evaluation, and analysis

We introduce the action recognition framework, which is used throughout this thesis work, embedding the novel algorithms for action recognition. Our framework is based on local features, and it consists of three main steps: local spatio-temporal video features extraction,

video-action representation, and video-action recognition. We select and review the most popular and the most prominent techniques for each of these steps, *i.e.*:

- Extraction of local features: Spatio-Temporal Interest Points [Laptev 2005] and Dense Trajectories [Wang 2011a].
- Video-action representation: bag-of-features and Fisher vector encoding.
- Video-action classification: Support Vector Machines [Cortes 1995], with the exponential χ^2 kernel for the bag-of-features, and the linear kernel for the Fisher vector encoding.

We also review five popular state-of-the-art datasets (*e.g.* HMDB51, MSR Daily Activity 3D, and URADL datasets) and we present their statistical analysis. Moreover, we propose a new dataset for the recognition of realistic human actions of daily living. Then, we present an extensive evaluation of the selected techniques, we compare these techniques, and present an analysis of the results. Using the experiments, we try to find the answers to unexplained questions in the state-of-the-art literature:

- Does the Spatio-Temporal Interest Points or the Dense Trajectories give better results with the bag-of-features approach and the Fisher vector encoding?
- Does the bag-of-features with the L1 or the L2 norm give better results?

The above contributions are presented in Chapter 3, and a part of this work was published in [Bilinski 2011].

New video-related local spatio-temporal descriptors: (1) Video Covariance Matrix Logarithm descriptor, and (2) Video Brownian Covariance descriptor

The popular state-of-the-art descriptors, *i.e.* Histogram of Oriented Gradients, Histogram of Optical Flow, and Motion Boundary Histograms, are based on a 1-dimensional histogram representation of individual features, and they directly model values of given features. The joint statistics between individual features are ignored, whereas such information may be informative. Therefore, we propose two novel descriptors, (1) Video Covariance Matrix Logarithm, and (2) Video Brownian Covariance, to model relationships between different pixel-level appearance features such as intensity and gradient. We propose a method to compute these representations on space-time video volumes extracted from a video sequence.

- Video Covariance Matrix Logarithm Descriptor:
  We present a new descriptor called Video Covariance Matrix Logarithm, which is based on a covariance matrix representation, and it models linear relationships between pixel-level features. This contribution is presented in Chapter 4.
- Video Brownian Covariance Descriptor:
  We present a new descriptor called Video Brownian Covariance, which is based on a Brownian covariance, a natural extension of the classical covariance measure,

and it measures all types of dependence between pixel-level features in an arbitrary dimension. This contribution is presented in Chapter 5, and it was published in [Bilinski 2014].

New higher-level feature representations: Person-Centric Dense Tracklets, Pairwise Features, and Contextual Features

The existing local feature encoding techniques, *e.g.* bag-of-features and Fisher vector encoding, ignore information about the spatio-temporal positions of features, relations among the features, and local densities of features. Therefore, we propose three techniques to overcome these three limitations:

- Person-Centric Dense Tracklets:
  The existing local feature encoding techniques ignore spatial positions of local features, whereas such information may be very useful for action recognition. Therefore, we introduce the idea of person-centric dense trajectories for action recognition. We focus on short motion trajectories (often called tracklets), which in a natural way describe moving objects in a video sequence. Our main idea is that action recognition ought to be performed using a dynamic coordinate system corresponding to an object of interest. We propose an object-centric local feature representation of motion trajectories, which allows to add spatial information of features to a local feature encoding technique. This contribution is presented in Chapter 6, and it was published in [Bilinski 2013].

- Pairwise Features: GARPF - Geometric and Appearance Relations of Pairwise Features
  The existing local feature encoding techniques ignore relations among local features, whereas such information may also be very useful for action recognition. To overcome this limitation we propose new pairwise features. The existing pairwise features based techniques use the discriminative power of individual features and capture visual relations among features. However, they ignore information about the spatio-temporal geometric relations between features (*i.e.* Δx, Δy, Δt) and the spatio-temporal orders between features. Moreover, some of the above techniques can only handle small codebooks, and they ignore associations between geometric and appearance relations among features.
  Therefore, we propose much finer representation of pairwise features, called Geometric and Appearance Relations of Pairwise Features (GARPF), that overcomes the above limitations. The GARPF representation captures statistics of pairwise co-occurring local spatio-temporal features. It encodes geometric and appearance relations among features in a single descriptor. Our video representation captures not only global distribution of features but also focuses on geometric and appearance relations among the features. Calculating video representations with different geometrical arrangements among the features, we keep important association between appearance and geometric information. This contribution is presented in Chapter 7, and it was published in [Bilinski 2012b].

- Contextual Features: STOCF - Space-Time Ordered Contextual Features:
  The existing local feature encoding techniques ignore information about local distribution of features, whereas such information also may be very useful for action recognition. To overcome this limitation we propose new contextual features. The existing contextual features based techniques use the discriminative power of individual features and capture local densities of features in feature-centric neighborhoods. To capture structural information they use the spatio-temporal grids; however, the spatio-temporal grid is limited in terms of detailed description providing only a coarse representation. Moreover, the existing techniques ignore information about the spatio-temporal orders among features.
  Therefore, we propose a new, much finer representation of contextual features, called Spatio-Temporal Ordered Contextual Features (STOCF), that overcomes the above limitations. We propose contextual features which capture both local densities of features and statistics of space-time ordered features. Moreover, our representation encodes information about the order of local features. This contribution is presented in Chapter 8, and it was published in [Bilinski 2012a].

Evaluation, comparison, and analysis of the proposed approaches

We present an extensive evaluation of all the proposed approaches in this manuscript. As before, the experiments are conducted on various challenging datasets, including HMDB51 [Kuehne 2011], MSR Daily Activity 3D [Wang 2012], and URADL [Messing 2009] datasets. We compare our techniques with each other and with state-of-the-art approaches. Moreover, we present a deep analysis of the results. Using the experiments, we try to find an answer to the question "When and which approach should we apply depending on videos and actions?". This contribution is mainly presented in Chapter 9, and partially in Chapters $3-8$.

1.6 Thesis Structure

In order to organize this dissertation in logical and coherent parts, we have divided this manuscript into ten chapters. A brief description of the consecutive chapters is as follows:

- *Chapter 2* – **Related Work**:
  In this chapter, we provide a brief literature overview of action recognition approaches. We review existing literature focusing on the most related and prominent state-of-the-art research techniques related to our work. We also discuss the critical points where the existing methods succeed and fail to recognize actions in videos.

- *Chapter 3* – **Action Recognition Framework, Its Evaluation, and Analysis**:
  In this chapter, we introduce the action recognition framework, which is used throughout this thesis work, embedding the novel algorithms for action recognition. It consists of three steps: local spatio-temporal video features extraction, video-action representation, and video-action recognition. We review the popular techniques for each of the steps. Then, we present five state-of-the-art action recognition

datasets that we use in the following evaluations, and we present the statistical analysis of these datasets. Moreover, we propose a new dataset for the recognition of realistic human actions of daily living. Finally, we present the extensive evaluation, comparison, and analysis of the presented techniques.

- *Chapter 4* – **Video Covariance Matrix Logarithm**:
  In this chapter, we present a new descriptor called Video Covariance Matrix Logarithm. The new descriptor is based on a covariance matrix representation, and it models linear relationships between different pixel-level appearance features such as intensity and gradient. We apply this descriptor to encode neighborhoods of local space-time video volumes.

- *Chapter 5* – **Video Brownian Covariance**:
  In this chapter, we present a new descriptor called Video Brownian Covariance. The new descriptor is based on a Brownian covariance, a natural extension of the classical covariance measure, and it measures all types of dependence between pixel-level features in an arbitrary dimension. Similarly to the Video Covariance Matrix Logarithm descriptor, we use pixel-level appearance features such as intensity and gradient, and we apply this descriptor to encode neighborhoods of local space-time video volumes.

- *Chapter 6* – **Relative Trajectories**:
  In this chapter, we introduce the idea of person-centric dense trajectories for action recognition. We focus on short motion trajectories (often called tracklets), which in a natural way describe moving objects in a video sequence. Our main idea is that action recognition ought to be performed using a dynamic coordinate system corresponding to an object of interest. We propose an object-centric local feature representation of motion trajectories, which allows to add spatial information of features to a local feature encoding technique.

- *Chapter 7* – **Geometric and Appearance Relations of Pairwise Features**:
  In this chapter, we propose a fine representation of pairwise features, which captures statistics of pairwise co-occurring local spatio-temporal features. The proposed representation encodes geometric and appearance relations among features in a single descriptor. Our video representation captures not only global distribution of features but also focuses on geometric and appearance (both visual and motion) relations among the features. Calculating video representations with different geometrical arrangements among the features, we keep important association between appearance and geometric information.

- *Chapter 8* – **Spatio-Temporal Ordered Contextual Features**:
  In this chapter, we propose a fine representation of contextual features, which could be applied to any spatio-temporal features, pairwise features, and even contextual features. We propose contextual features, which capture both local densities of features and statistics of space-time ordered features. The proposed representation encodes information about the order of local features.

- *Chapter 9* – **Comparison of Approaches**:
  In this chapter, we present an extensive evaluation of all the proposed in this manuscript approaches. We compare our techniques with each other and with the state-of-the-art approaches. Moreover, we present a deep analysis of the proposed methods and the obtained results. Using the experiments, we try to find an answer to the question "When and which approach should we apply depending on actions and videos?".

- *Chapter 10* – **Conclusions and Perspectives**:
  In this chapter, we conclude our dissertation, summarizing the main advantages and disadvantages of all the proposed approaches, and we present possible future directions, extensions, and perspectives of the proposed techniques.

CHAPTER 2

Related Work

Contents

2.1 Human Body Model Based Methods 18
2.2 Holistic Methods 21
2.3 Local Feature Methods 23
2.3.1 Local Features 23
2.3.2 Collections of Local Features 30
2.3.3 Local Features Encoding 36
2.4 Classifiers 39
2.5 Conclusion 43

To present the context of our work in the domain of action recognition and to emphasize our main contributions in this research area, we begin by reviewing the existing literature on action recognition in videos. We provide a brief literature overview, describing the most relevant and the most prominent state-of-the-art research techniques related to our work. We also discuss the critical points where the existing methods succeed and fail to recognize actions in videos.

Over the last years various techniques have been proposed for action recognition in videos. In this dissertation, we divide the existing techniques into three main categories:

- **Human body model based methods** (Section 2.1) - action recognition is based on the extraction of 2D or 3D information on human body parts, such as body part configuration, body part positions, and movements.
- **Holistic methods** (Section 2.2) - action recognition is based on the extraction of information on people localization in videos, and a global representation of human body structure, shape and movements is used for action recognition. Holistic techniques do not use information on human body parts.
- **Local feature methods** (Section 2.3) - action recognition is based on local features. No information on human body model or localization of people is required.

In this chapter we focus on the most relevant approaches related to our work. For general surveys on action recognition we refer to the work of Weinland *et al*. [Weinland 2011], Aggarwal *et al*. [Aggarwal 2011], Vishwakarma *et al*. [Vishwakarma 2013], and Poppe *et*

al. [Poppe 2010].

Then, in Section 2.4, we present a brief overview of popular machine learning classification algorithms, which can be applied to any of the above-mentioned category of action recognition techniques. Finally, we conclude this chapter in Section 2.5.

2.1 Human Body Model Based Methods

Human body model based methods for action recognition use 2D or 3D information on human body parts, such as body part positions and movements. Typically, the pose of a human body is recovered and action recognition is based on pose estimation, human body parts, trajectories of joint positions, or landmark points.

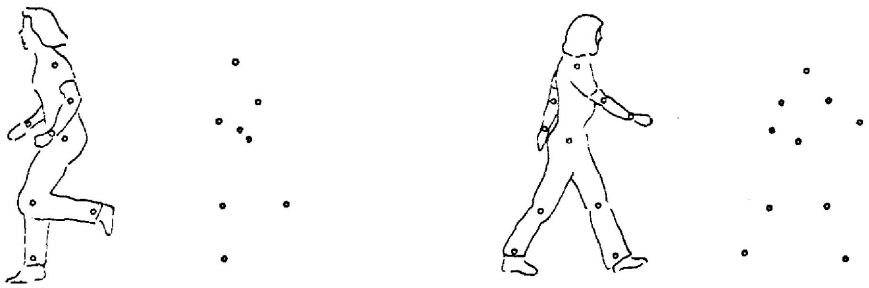


Figure 2.1 – Johansson [Johansson 1973]: Outline contours of a running and a walking subject, and the corresponding moving light displays attached to the human body.

Human body model based methods are inspired by a psychophysical research work of Johansson [Johansson 1973] on visual perception of motion patterns characteristics of living organisms in locomotion. Johansson has shown that humans can recognize actions from the motion of a few moving light displays attached to the human body, describing the motions of the main human body joints. He has found that between 10 and 12 moving light displays in adequate motion combinations in proximal stimulus evoke an impression of human walking, running, dancing, *etc*. (see Figure 2.1).

Yilmaz and Shah [Yilma 2005] have proposed an approach for recognition of human actions in videos captured by uncalibrated moving cameras. The proposed approach is based on trajectories of human joint points. In order to handle camera motion and different viewpoints of the same action in different environments, they use the multi-view geometry between two actions and they propose to extend the standard epipolar geometry to the geometry of dynamic scenes where the cameras are moving. Sample trajectories of the walking actions captured using a stationary camera and a moving camera are presented in Figure 2.2.

Ali *et al*. [Ali 2007] have also proposed an approach based on trajectories of reference

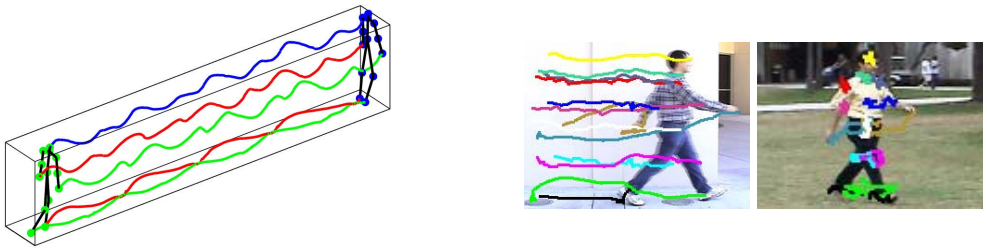


Figure 2.2 – On the left: Ali *et al*. [Ali 2007]: 3D trajectories generated by a head, two hands and two feet are shown for the running action. On the right: Yilmaz and Shah [Yilma 2005]: Trajectories of the walking actions are captured using a stationary camera and a moving camera.

joint points. These trajectories are used as the representation of the non-linear dynamical system that is generating the action, and they use them to reconstruct a phase space of appropriate dimension by employing a delay-embedding scheme. The properties of the phase space are captured in terms of dynamical and metric invariants that include Lyapunov exponent, correlation integral and correlation dimension. Finally, they represent an action by a feature vector which is a combination of these invariants over all the reference trajectories. Sample 3D trajectories generated by a head, two hands and two feet for the running action are presented in Figure 2.2.

Although all these techniques have shown to be promising, they have a big limitation. The extraction of human body model and body joint points in realistic and unconstrained videos is still a very difficult problem, and therefore these techniques remain limited in applicability.

Figure 2.3 – Microsoft Kinect on the left and ASUS Xtion PRO LIVE on the right.

The recent introduction of the cost-effective depth cameras helps in the extraction of human body joint points. The two most popular depth cameras are Microsoft Kinect and ASUS Xtion PRO LIVE motion sensor (see Figure 2.3). Both these sensors consist of an infrared pattern projector and an infrared camera to capture depth data, and a RGB camera to capture color images, see Figure 2.4. The depth cameras provide 3D depth data of the scene, which largely helps in people segmentation and in obtaining the 3D joint positions of the human skeleton.

Several techniques that use such depth cameras and the extracted human skeleton have

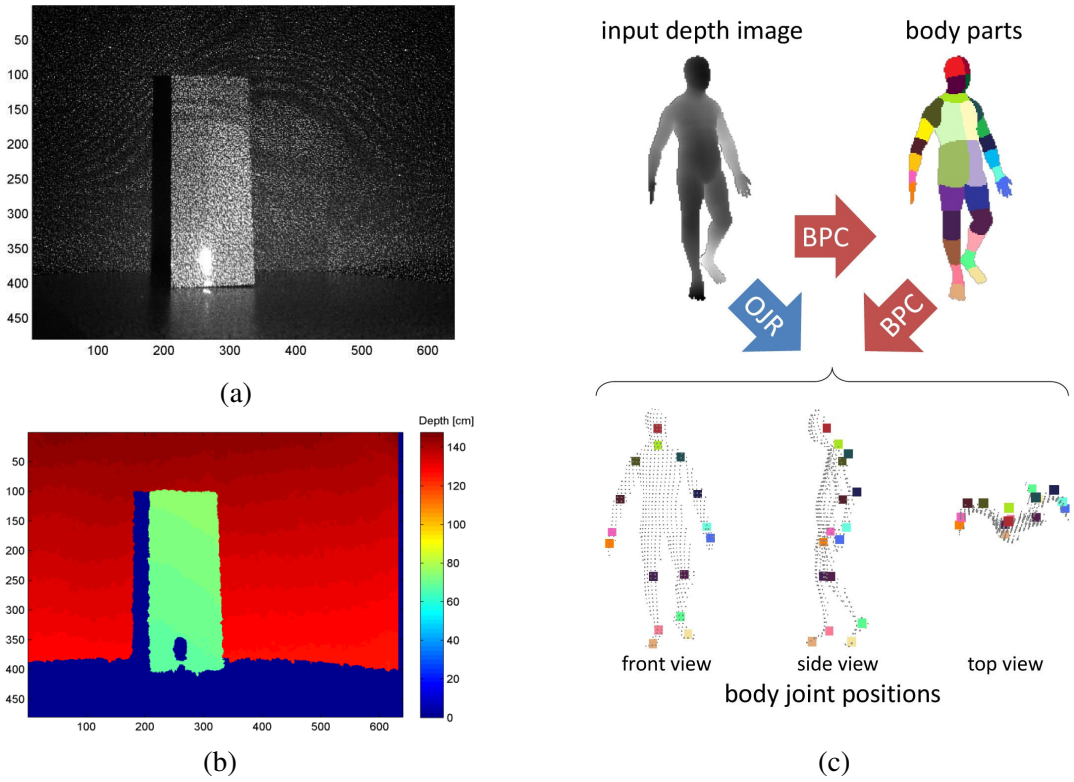


Figure 2.4 – (a) Sample infrared image, from the Microsoft Kinect, presenting pattern of speckles projected on a sample scene [Litomisky 2012]. (b) The resulting depth image [Litomisky 2012]. (c) Two sample approaches for estimating human pose from single depth images [Shotton 2012]. Body part classification (BPC) predicts a body part label at each pixel (labels are represented by colors), and then uses these labels to localize the body joints. Offset joint regression (OJR) more directly regresses the positions of the joints.

been proposed, *e.g*. Raptis *et al*. [Raptis 2011] and Wang *et al*. [Wang 2012]. However, the cost-effective depth cameras also have some limitations.

- First of all, the range of the depth sensor is limited, *e.g*. Microsoft recommends to use the Kinect sensor in the range between 0.5m and 4m [1], and ASUS recommends to use the Xtion PRO LIVE motion sensor in the range between 0.8m and 3.5m [2]. Although it is possible to use the depth data at larger distances, the quality of the data is degraded by the noise and low resolution of the depth measurements. For example, it is possible to get the depth data even up to 10 meters from the Microsoft Kinect [Litomisky 2012], but Khoshelham and Elberink [Khoshelham 2012] show that: (a) the random error of depth data increases quadratically with increasing distance and reaches 4 cm at the range of 5 meters, (b) the depth resolution decreases quadratically with increasing distance and the point spacing in the depth direction reaches 7 cm at the range of 5 meters, and (c) for indoor mapping applications the data should be acquired within 1-3 m distance to the sensor. Human pose estimation in such motion sensors is typically extracted from depth images, *e.g*. Shotton *et al*. [Shotton 2012] (see Figure 2.4), and as a result the quality of human pose estimation algorithms decreases with increasing distance.

- Second of all, skeleton tracking and the estimated 3D joint positions are noisy and can produce inaccurate results or even fails when serious occlusion occurs [Wang 2012], *e.g*. when one leg is in front of the other, a hand is touching another body part, or two hands are crossing.

Therefore, in this thesis we focus on action recognition using RGB cameras due to many potential applications of such sensors.

2.2 Holistic Methods

Shape and silhouette information based features are one of the very first characteristics, which were used to represent human body structure and its dynamics for action recognition in videos.

One of the first approaches using silhouette images and features for action recognition is the work of Yamato *et al*. [Yamato 1992]. They extract a human shape mask for each image, calculate a grid over the silhouette, and for each cell of the grid calculate the ratio of foreground to background pixels (see Figure 2.5). Then, each grid representation of an image is assigned to a symbol, which corresponds to a codeword in the codebook created by the Vector Quantization technique. Finally, Hidden Markov Models (HMMs) are applied for action recognition and the model which best matches the observed symbol sequence is chosen as the recognized action category.

[1] http://msdn.microsoft.com/en-us/library/hh438998.aspx
[2] http://www.asus.com/Multimedia/Xtion_PRO_LIVE/specifications/

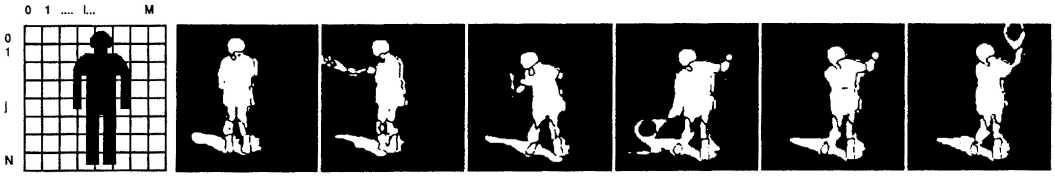


Figure 2.5 – Yamato *et al.* [Yamato 1992]: Mesh feature (the first image), and the sample shape masks for the forehand stroke action from the tennis action (the remaining images).

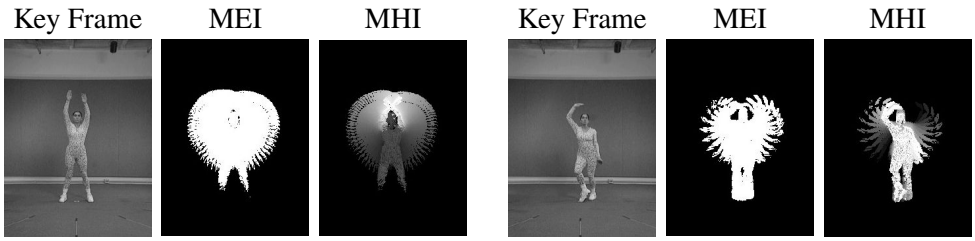


Figure 2.6 – Bobick and Davis [Bobick 2001]: MEI and MHI representations calculated for two sample actions ("move 4" and "move 17") together with sample key frames.

Bobick and Davis [Bobick 2001] were first to introduce the idea of temporal templates for action recognition. They extract human shape masks from images and accumulate their differences between consecutive frames. These differences are then used to construct a binary motion-energy image (MEI) and a scalar-valued motion-history image (MHI) (see Figure 2.6). The former indicates the presence of motion, and the latter represents the recency of motion, *i.e.* the pixel intensity is a function of the temporal history of motion at that point. Then, they proposed a recognition method matching temporal templates against stored instances of actions. The MEI and MHI together can be considered as a two component version of a temporal template.

Figure 2.7 – Blank *et al.* [Blank 2005]: Sample spatio-temporal volumes constructed by stacking silhouettes over a given sequence.

Blank *et al.* [Blank 2005] proposed a model based on three-dimensional shapes induced by the silhouettes in the space-time volume. At each frame, they compute a silhouette information using a background subtraction technique. They stack silhouettes

over a given sequence to form a spatio-temporal volume (see Figure 2.7). Then, they use properties of the solution to the Poisson equation to extract space-time features such as local space-time saliency, action dynamics, shape structure and orientation. They use chunks of 10 frames length and match these chunks using a sliding window approach. The action classification is done using simple nearest neighbour algorithm with an Euclidean distance.

The main disadvantage of the holistic based method is the requirement of shape, silhouette extraction, what is typically done by segmentation. The accuracy of these techniques is highly related to the correctness of the segmentation and the precise segmentation is very difficult to obtain in real world videos.

2.3 Local Feature Methods

Action recognition based on local features is one of the most active research topics. The main advantage of the local features based methods is that no information on human body model or localization of people is required.

In this section, we focus on local feature methods. In Section 2.3.1, we present spatio-temporal local feature detectors and descriptors for videos. Then, in Section 2.3.2 we present higher-order features, the collections of local features. Finally, in Section 2.3.3 we present various video - action encoding techniques based on local features.

2.3.1 Local Features

Local features are extracted by applying a local feature detector and then by encoding spatio-temporal neighbourhoods around the detected features using a local feature descriptor. In this section we describe the most popular local spatio-temporal detectors (see Section 2.3.1.1) and descriptors (see Section 2.3.1.2) for action recognition in videos.

2.3.1.1 Local Feature Detectors

Local feature detectors for videos can be divided into two categories: spatio-temporal interest point detectors and trajectory detectors.

Spatio-Temporal Interest Point Detector

One of the first works on local feature detectors for videos is the work of Laptev and Lindeberg [Laptev 2003]. They proposed the Harris3D interest point detector, which is an extension of the Harris detector [Harris 1988] to the spatio-temporal domain by requiring the video values in space-time to have large variations in both the spatial and the temporal dimensions. The Harris3D detector calculates a spatio-temporal second-moment matrix at each video point and searches for regions that have significant eigenvalues of

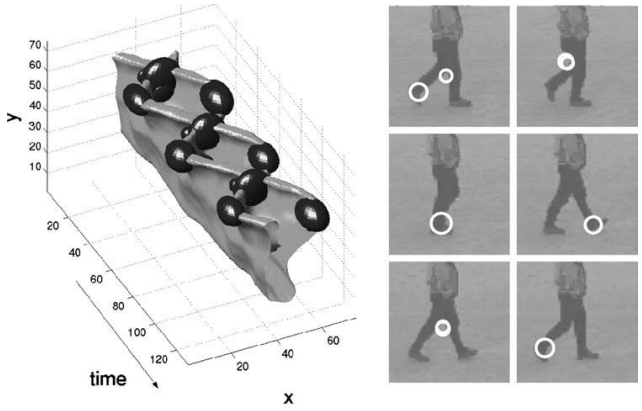


Figure 2.8 – Laptev [Laptev 2005]: the Harris3D interest points from the motion of the legs of a walking person; left image: 3D plot with a thresholded level surface of a leg pattern (upside down) and the detected points (ellipsoids); right image: interest points overlayed on single frames in the original video sequence.

the matrix. The final spatio-temporal points are detected as local positive spatio-temporal maxima. Moreover, the detected points have to be the local extrema of the normalized spatio-temporal Laplace operator, which is defined to select the spatio-temporal scales of points.

Dollar *et al.* [Dollar 2005] observed that sometimes true spatio-temporal corners are rare, even when interesting motion occurs, and might be too rare in certain cases, *e.g.* for face expression recognition. Therefore, they proposed the Gabor detector, which gives denser results than the Harris3D. The Gabor detector applies a set of spatial Gaussian kernels and temporal Gabor filters. The final spatio-temporal points are detected as local maxima of the defined response function.

Different from the above, Oikonomopoulos *et al.* [Oikonomopoulos 2005] proposed a space-time extension of a salient region detector [Kad 2003] using entropy. The proposed detector selects the scales at which the entropy achieves local maxima and forms spatio-temporal salient regions by clustering spatio-temporal points with similar location and scale.

Willems *et al.* [Willems 2008] proposed the Hessian3D interest point detector, which is a spatio-temporal extension of the Hessian saliency measure for blob detection in images [Beaudet 1978]. The Hessian3D detector calculates the Hessian matrix at each interest point and uses the determinant of the Hessian matrix for point localization and scale selection. The detector uses integral video to speed up computations by approximating derivatives with box-filter operations. The detected points are scale-invariant and dense, typically they are denser than from the Harris3D detector but not that dense as from the Gabor detector.

Most of the techniques use local information to detect spatio-temporal interest points. Wong and Cipolla [Wong 2007] proposed an interest point detector which uses global information, *i.e.* the organisation of pixels in a whole video sequence, by applying non-negative matrix factorization on the entire video sequence. The proposed detector is based on the extraction of dynamic textures, which are used to synthesize motion and identify important regions in motion. The detector extracts structural information, *e.g.* the location of moving parts in a video, and searches for regions that have a large probability of containing the relevant motion.

Different from the above techniques, Wang *et al.* [Wang 2009] proposed to apply dense sampling. The dense sampling extracts interest points at regular positions and scales in space and time. The sampling is done using 5 dimensions (x, y, t, σ, τ), where (x, y, t) is the spatio-temporal position of a point, σ is the spatial scale, and τ is the temporal scale. This detector extracts a big amount of features but is also able to extract relevant video features.

When faced with the decision "Which Spatio-Temporal Interest Point detector gives the best results?", there is no clear answer. Wang *et al.* [Wang 2009] compared Harris3D, Gabor detector, Hessian3D, and dense sampling. The comparison was done on three datasets: (a) KTH dataset, where Harris3D achieved the best results, (b) UCF dataset, where dense sampling achieved the best results, and (c) Hollywood2 dataset, where dense sampling achieved the best results using reference videos, but Harris3D with full resolution videos achieved better results than the dense sampling with reference videos. Therefore, according to that evaluation, there is no single detector that always achieves the best results, but among the four selected detectors (*i.e.* Harris3D, Gabor detector, Hessian3D, and dense sampling), the best results per dataset are achieved either by Harris3D or dense sampling.

In this thesis, we use the Harris3D detector to extract local spatio-temporal interest points in videos. The main reasons are:

- The Harris3D detector is the most popular and the most widely used local spatio-temporal interest point detector. This allows for fair comparison with many state-of-the-art approaches.
- There is no single detector that always achieves the best results and the Harris3D detector receives very good results [Wang 2009].
- Moreover, the Harris3D detector extracts relatively sparse number of interest points, while the dense sampling produces a very large number of features (typically $15 - 20$ times more than other detectors [Wang 2009]). This means that the Harris3D detector requires less memory to store the calculated points, and thus less time is required to represent a video sequence.

Trajectory Detector

Trajectories are typically extracted by detecting interest points and tracking them in the consecutive frames.

One of the best-known feature tracking algorithm is the KLT (Kanade-Lucas-Tomasi) [Lucas 1981, Tomasi 1991, Shi 1994]. The KLT algorithm locates good features for tracking by examining the minimum eigenvalue of each 2×2 gradient matrix, and then features are tracked using a Newton-Raphson method of minimizing the difference between the two windows.

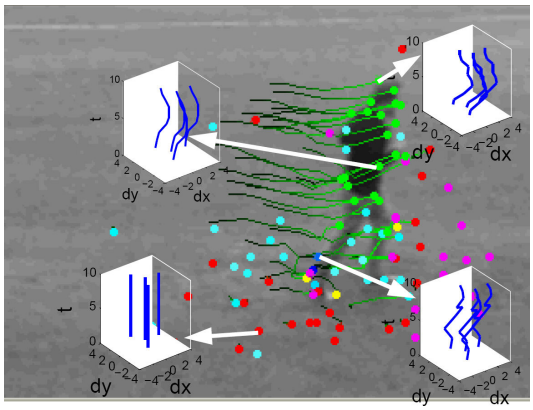


Figure 2.9 – Matikainen *et al.* [Matikainen 2009]: Feature points are tracked using the KLT tracking algorithm, and then the trajectories are clustered and assigned to the library of trajectories.

Sample work using the KLT tracker is the work of Matikainen *et al.* [Matikainen 2009], where they extract trajectories of fixed length using a standard KLT tracker and then cluster the trajectories. They compute an affine transformation matrix for each cluster center, and the elements of the matrix are then used to represent the trajectories.

Messing *et al.* [Messing 2009] proposed to apply a different detector of points, Harris3D detector, and track points with the KLT tracker. Then, the trajectories, which vary in length, are represented as sequences of log-polar quantized velocities and used for action classification.

Kaaniche and Bremond [Kaaniche 2009] proposed to detect interest points using Shi and Thomasi corner detector [Shi 1994] or Features from Accelerated Segment Test (FAST) corner detector [Rosten 2006], and then track points using matching the HOG descriptors over consecutive frames. The obtained trajectories vary in length and according to the authors are less sensitive to the noise than the trajectories from the KLT tracker.

Different from the above techniques, Sun *et al.* [Sun 2009] proposed to extract

trajectories based on the pairwise SIFT matching over consecutive frames. They claim that scale-invariant properties of the SIFT descriptor is a better choice when compared to the Harris and KLT based feature trackers.

In [Sun 2010], Sun *et al.* proposed to combine the tracking results of the KLT and the SIFT trackers, and formulated the visual matching and tracking in a unified constrained optimization problem. In order to extract dense trajectories, the authors add interior points that are neither corner points tracked by the KLT nor by the SIFT trackers by interpolating of the surrounding flows, subject to block-matching constraints.

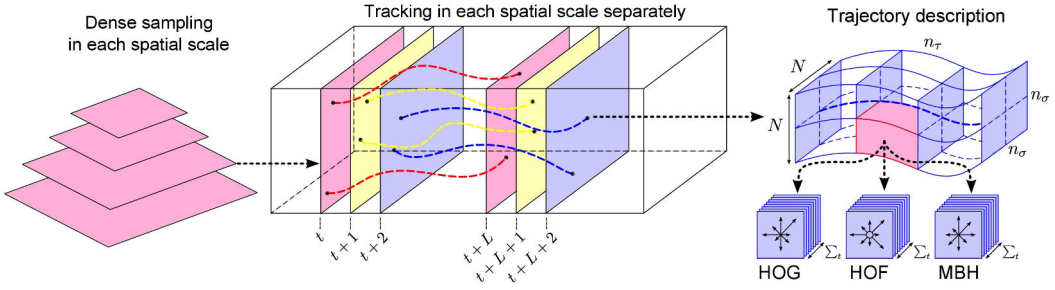


Figure 2.10 – Wang *et al.* [Wang 2010]: Overview of the dense trajectories; left image: dense sampling on multiple spatial scales; middle image: feature tracking in the corresponding spatial scale over L frames; right image: descriptors calculated around a trajectory.

Wang *et al.* [Wang 2010] also proposed to extract dense trajectories. They apply dense sampling to extract interest points and track them using a dense optical flow field. Then, the trajectories are represented using the trajectory shape, HOG, HOF and MBH descriptors. This technique gives a great number of trajectories but according to the authors they obtain better results than the trajectories from the KLT and the SIFT tracking algorithms.

When faced with the decision "Which trajectory detector gives the best results?", we refer to the work of Wang *et al.* [Wang 2013a], where:

- the dense trajectories were compared with trajectories extracted by Kanade-Lucas-Tomasi (KLT) tracker and by SIFT descriptor matching. In all cases, *i.e.* on nine datasets, the dense trajectories outperformed the other trajectories.
- the dense trajectories were compared with the trajectories from [Sun 2010] and from [Messing 2009] on the KTH dataset, and the dense trajectories outperformed the other trajectories.

Although the dense trajectories detector extracts a large amount of features, it outperforms other trajectory detectors [Wang 2013a], and therefore we use it to extract features in videos.

2.3.1.2 Local Feature Descriptors

Local feature descriptors capture shape and motion information in a local neighborhood surrounding interest points and trajectories.

One of the first works on local feature descriptors for videos is the work of Laptev and Lindeberg [Laptev 2006]. They presented and compared several descriptors based on motion representations in terms of spatio-temporal jets (higher-order derivatives), position dependent histograms, position independent histograms, and principal component analysis computed for either spatio-temporal gradients or optical flow. They reported the best results for descriptors based on histogram of spatio-temporal gradients and optical flow.

Dollar *et al*. [Dollar 2005] also proposed several local feature descriptors. They considered three transformations to local neighborhoods: normalized pixel values, the brightness gradient, and windowed optical flow. They also considered three methods to create a feature vector: flattening the local neighborhood into a vector, histogramming the values in the local neighborhood, and dividing the local neighborhood into a grid and histogramming the values in each cell of the grid. For all methods, the PCA was applied to reduce the dimensionality of the final descriptors. They reported the best results for descriptors based on concatenated gradient information.

The HOG (Histogram of Oriented Gradients) and HOF (Histogram of Optical Flow) are the popular local feature descriptors for videos proposed by Laptev *et al*. [Laptev 2008]. The HOG descriptor for videos is the variant of the HOG image descriptor [Dalal 2005]. In order to embed structure information in a descriptor, the local neighborhood surrounding a local feature is divided into a spatio-temporal grid. For each cell of the grid, a histogram descriptor is calculated. Then, the histograms are normalized and concatenated into the final descriptor. The HOG descriptor encodes visual appearance and shape information; the edge orientations are calculated and quantized into histogram bins. The HOF descriptor encodes motion information; the optical flow is calculated and quantized into histogram bins.

The 3DSIFT (3-Dimensional SIFT) is an extension of the SIFT (Scale Invariant Feature Transform) image descriptor [Lowe 2004] to the spatio-temporal domain proposed by Scovanner *et al*. [Scovanner 2007]. It is based on the spatio-temporal grid idea and spatio-temporal gradients. Each pixel is weighted by a Gaussian centered on the given position and votes into a grid of histograms of oriented gradients. A Gaussian weighting is applied to give less importance to gradients farther away from the local feature center. To be rotation-invariant, a dominant orientation is determined and is used for orienting the grid descriptor.

The HOG3D descriptor is another extension of the HOG image descriptor [Dalal 2005] to the spatio-temporal domain proposed by Klaser *et al*. [Klaser 2008]. The HOG3D is based on the spatio-temporal grid idea and 3D gradients, which are calculated and quan-

tized to the histograms of 3D gradient orientations based on convex regular polyhedrons.

The main differences between the HOG, ESIFT, and the HOG3D spatio-temporal descriptors are: (1) the HOG descriptor only considers spatial gradients, and the ESIFT and the HOG3D descriptors consider spatio-temporal 3D gradient orientation, and (2) the ESIFT descriptor uses regular binning based on spherical coordinates, and the HOG3D descriptor uses regular polyhedrons and spherical coordinates for which the amount of bins can be controlled separately for spatial and temporal gradient orientations.

The ESURF (Extended SURF) is an extension of the SURF (Speeded Up Robust Features) image descriptor [Bay 2006] to the spatio-temporal domain proposed by Willems *et al*. [Willems 2008]. The ESURF divides the local neighborhood surrounding a local feature into a spatio-temporal grid, and it represents each cell of the grid by a vector of weighted sums of uniformly sampled responses of Haar-wavelets along the three (x, y, z) axes.

The MBH (Motion Boundary Histogram) is an extension of the MBH image descriptor [Dalal 2006] to the spatio-temporal domain proposed by Wang *et al*. [Wang 2011a]. The MBH descriptor separates the optical flow field into its x and y components. Spatial derivatives are computed separately for the horizontal and vertical components of the optical flow, and orientation information is quantized into histograms, similarly to the HOG descriptor. The MBH descriptor is also based on the spatio-temporal grid idea.

The Trajectory shape descriptor was proposed by Wang *et al*. [Wang 2011a] to encode a shape of the extracted dense trajectories. It describes a shape of a trajectory by a sequence of displacement vectors normalized by the sum of displacement vector magnitudes.

When faced with the decision "Which local feature descriptor should we use?", we refer to the work of:

- [Wang 2009], which recommends using the combination of HOG and HOF descriptors for the Spatio-Temporal Interest Points.

- [Wang 2013a], which recommends using the combination of Trajectory shape descriptor, HOG, HOF, and MBH descriptors for the Dense Trajectories. This combination achieved the best results on 8 out of 9 datasets, when compared with each of the descriptors separately; the best result for the remaining dataset was achieved by the MBH descriptor alone. The authors underline the importance of the MBH descriptor, which is robust to camera motion.

Therefore, in this thesis we also use the HOG and HOF descriptors for the Spatio-Temporal Interest Points, and the Trajectory shape descriptor, HOG, HOF, and MBH descriptors for the Dense Trajectories, as it was recommended by the above papers.

2.3.2 Collections of Local Features

The methods based on local features presented in the previous section (Section 2.3.1) are based on the discriminative power of individual local features and global statistics of individual local features. Although these techniques have shown very good results in action recognition, they also have a few limitations:

- they ignore position of features,
- they ignore local density of features,
- they ignore relations among the features (*i.e.* visual appearance and motion relations, spatio-temporal order among features, and spatio-temporal geometric relations among features (*i.e.* Δx, Δy, Δt)).

These techniques might distinguish various actions but may fail to distinguish similar actions as they do not use all the available information.

A common way to overcome these limitations is to use either spatio-temporal grids [Laptev 2008] or multi-scale pyramids [Lazebnik 2006]. However, these techniques are still limited in terms of detailed description providing only a coarse representation.

In order to cope with these problems, several solutions have been proposed, most of which try to create higher-level feature representations and use them together with the bag-of-features approach. These higher-level feature representations we can divide into 2 categories:

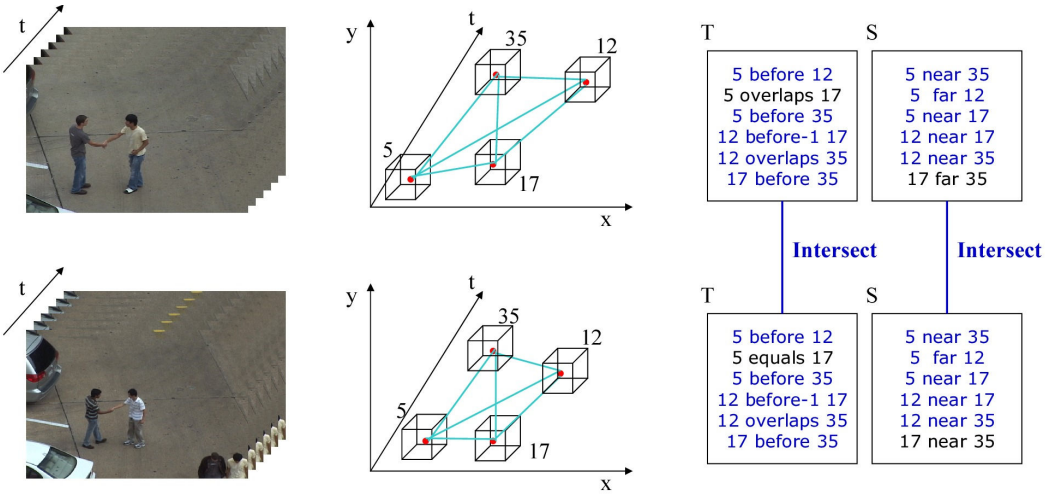
- **Pairwise Features** (Section 2.3.2.1) - features capturing pairwise relations among features.
- **Contextual Features** (Section 2.3.2.2) - features capturing relations among any number of neighbouring features.

These higher-level feature representations have shown to enhance the discriminative power of individual local features and improve action recognition accuracy.

2.3.2.1 Pairwise Features

One of the first studies on pairwise features is the work of Liu *et al.* [Liu 2008]. They proposed to explore the correlation of the compact visual word clusters using a modified correlogram. Firstly, they extract local features using the detector and the descriptor proposed by the Dollar *et al.* [Dollar 2005]. Then, they represent a video sequence using the bag-of-features approach. Instead of using the k-means algorithm, they apply Maximization of Mutual Information to discover the optimal number of codewords. Then, to capture the structural information they explore the correlation of the codewords. They apply the modified correlogram, which is somewhat scale invariant, translation and rotation invariant. As they calculate the probability of co-occurence between every pair of

codewords, they use small codebooks.



(a) 3-D XYT Video volumes (b) Spatio-temporal feature points (c) Relationships in histograms

Figure 2.11 – Ryoo *et al.* [Ryoo 2009]: Spatio-temporal relationship matching process: (a) two given videos, (b) extraction of local features and calculation of pairwise relations, (c) calculation of a relationship histogram per input video, and similarity between relationship histograms calculated as intersection.

Ryoo *et al.* [Ryoo 2009] proposed a spatio-temporal relationship matching technique, which is designed to measure structural similarity between sets of features extracted from two videos (see Figure 2.11). Firstly, the authors extract local features for every video sequence. Then, they create pairwise relations among features, and represent each video sequence using relationship histograms. The relationship histogram is created separately both for the spatial and the temporal order, and it is based on simple, constant and limited predicates indicating the order of features. Then, the authors compute the relationship histograms intersection to measure similarity between two videos. The main limitations of this technique are: (a) the relationship histograms use only simple predicates (*e.g.* before and after) to encode pairwise relations between local features, (b) the spatial and the temporal orders between local features are encoded independently and not both at the same time, and (c) the spatio-temporal geometric relations (*i.e.* Δx, Δy, Δt) among features are ignored.

Ta *et al.* [Ta 2010] proposed pairwise features, which encode both appearance and spatio-temporal relations of local features (see Figure 2.12). Firstly, the authors extract the Spatio-Temporal Interest Points (STIPs) from a video sequence. Then, the pairwise features are created by grouping pairs of STIPs, which are both close in space and close in time. The pairwise features are encoded by appearance and spatio-temporal relations of local features. The appearance relations are captured by concatenating the appearance

Figure 2.12 – On the left: Ta *et al*. [Ta 2010]: Sample pairwise features are presented as local features [Dollar 2005] detected as close in time and close in space. On the right: Matikainen *et al*. [Matikainen 2010]: Sample pairwise features are presented as pairs of local features selected to be discriminative for a specific action class.

descriptors of STIPs. The spatio-temporal relations are captured by a spatio-temporal distance between STIPs. Then, for each type of relations the bag-of-features approach is applied independently and the two obtained representations are concatenated. The main limitations of this technique are: (a) it is difficult to correctly set the spatial and temporal thresholds to decide which STIPs are both close in space and close in time, (b) spatio-temporal order between features is lost, and (c) association between appearance and the spatio-temporal geometric information is lost by calculating two independent codebooks.

Matikainen *et al*. [Matikainen 2010] also proposed a method for representing spatio-temporal relationships between features in the bag-of-features approach (see Figure 2.12). The authors use both the Spatio-Temporal Interest Points (STIPs) and trajectories to extract local features from a video sequence. Then, they combine the power of discriminative representations with key aspects of Naive Bayes. As the number of all possible pairs and relationships between features is big, they reduce the number of relationships to the size of the codebook. Moreover, they show that the combination of both the appearance and motion base features improves the action recognition accuracy. The main limitation of this technique is that it encodes the appearance and motion relations among features but it does not use information about the spatio-temporal geometric relations between features.

Banerjee *et al*. [Banerjee 2011] proposed to model pairwise co-occurrence statistics of visual worlds. Firstly, the authors extract local features and they create a codebook of local features represented by local descriptors. Instead of selecting the most discriminative relations between features, they use small codebooks, *i.e*. the codebook size is smaller than 20. They model local neighborhood relationships between local features in terms of a count function which measures the pairwise co-occurrence frequency of codewords. Then, the count function is transformed to the edges connecting the latent variables of a

Conditional Random Field classifier, and they explicitly learn the co-occurrence statistics as a part of its maximum likelihood objective function. The main limitations of this technique are: (a) it can only use small codebooks, and (b) it uses discriminative power of individual (appearance) features but information about the spatio-temporal geometric relations and spatio-temporal order between features is ignored.

In summary, most of the above pairwise features based techniques use the discriminative power of individual features and capture visual relations among features. However, the existing techniques ignore information about spatio-temporal geometric relations between features (*i.e.* Δx, Δy, Δt) and spatio-temporal order between features. Moreover, some of the above techniques can only handle small codebooks [Liu 2008, Banerjee 2011] due to quadratic processing time. Therefore, a new and optimized representation is needed to create a finer description of pairwise features.

2.3.2.2 Contextual Features

Pairwise features only capture relations between two features. Contextual features are able to capture relations among many features.

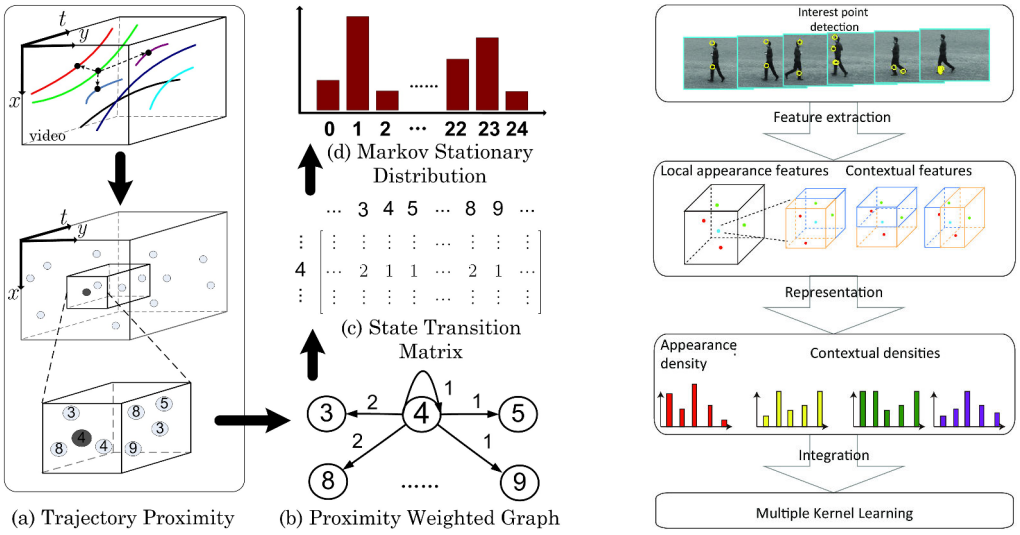


Figure 2.13 – Sun *et al.* [Sun 2009]: The proposed inter-trajectory context representation (left image). Wang *et al.* [Wang 2011b]: Overview of the proposed approach (right image).

One of the first studies on contextual features is the work of Sun *et al.* [Sun 2009] (see Figure 2.13). They proposed to model the spatio-temporal context information of video sequences based on the SIFT based trajectories. The spatio-temporal context is represented in a hierarchical way: point-level, intra-trajectory, and inter-trajectory context. The point-level context is measured as the average of all the SIFT features extracted around the trajectory. The intra-trajectory context is encoded as the transition and dynamics of the trajectory in spatio-temporal domain. The inter-trajectory context (see Figure 2.13)

is represented as contextual features and captures local occurrence statistics of quantized trajectories within figure-centric neighbourhoods. The intra-trajectory and inter-trajectory context encode the spatio-temporal context information into the transition matrix of a Markov process, and extract its stationary distribution as the final context descriptor. The main limitations of the proposed contextual features are: (a) they ignore pairwise relations among features, and (b) they ignore spatio-temporal geometric relations among features.

Similarly, Wang *et al*. [Wang 2011b] proposed to capture contextual statistics among interest points based on the density of features observed in each interest point's contextual domain (see Figure 2.13). Firstly, the authors extract local features for a given video sequence. Then, they create spatio-temporal contextual features that capture contextual interactions between interest points, *i.e*. they capture the density of all features observed in each interest point's mutliscale spatio-temporal contextual domain. Then, they apply the bag-of-features approach for local features and contextual features, and augment the obtained video representations using Multiple Kernel Learning approach. The main limitations of the proposed contextual features are: (a) they ignore pairwise relations among features, and (b) they ignore spatio-temporal geometric relations and spatio-temporal order among features.

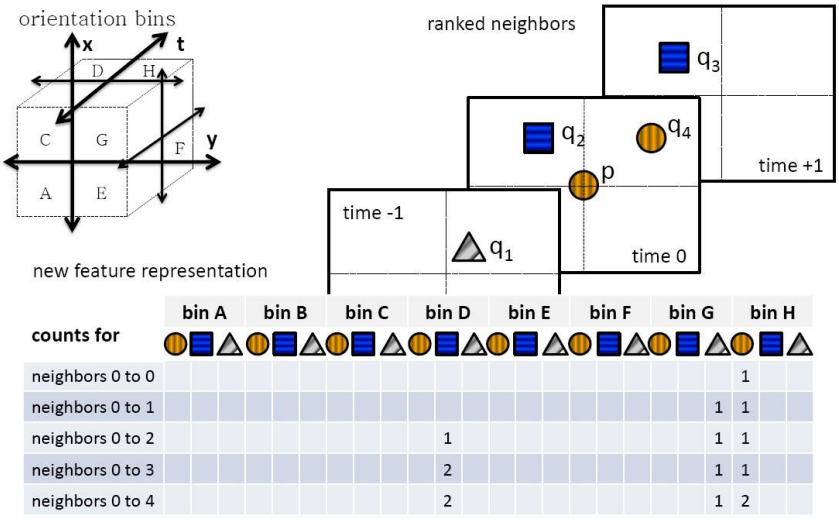


Figure 2.14 – Kovashka *et al*. [Kovashka 2010]: Contextual Features. A figure-centric neighbourhood divided into 8 orientations, three frames with sample features, and the histogram representation of the neighbourhood.

Kovashka *et al*. [Kovashka 2010] proposed figure-centric statistics that capture the orientation among features. Firstly, the authors extract local features from videos, *i.e*. they either apply: (1) dense sampling and represent interest points using HOG3D descriptors, or (2) Harris3D detector and represent interest points using HOG and HOF descriptors. Then, they create a visual vocabulary, they quantize local features, and use the quantized features to create figure-centric features consisting of the words associ-

ated with nearby points and their orientation with respect to the central interest point. Such figure-centric features are then recursively mapped to higher-level vocabularies, producing a hierarchy of figure-centric features. Moreover, the authors propose to learn the shapes of space-time feature neighborhoods that are the most discriminative for a given action category. The main limitations of this technique are: (a) only the orientation among feature points is captured and not the spatio-temporal distance relations among features, and (b) the contextual features are of high dimension ($40 \times k$, where k is the codebook size) and the clustering process of these contextual features might be time consuming.

All techniques presented above use the bag-of-features approach in order to encode the created contextual features. The techniques presented below create contextual features but do not use the bag-of-features approach.

Gilbert *et al*. [Gilbert 2009] proposed to use dense corner features that are spatially and temporally grouped in a hierarchical process. They build compound hierarchical features, which can be seen as contextual features, based on relationships of detected interest points, and they find frequently reoccurring patterns of features using data mining. The local features are represented only using scale, channel, and dominant orientation of features. The main limitations of this technique are: (a) it does not use visual and motion appearance information, and (b) it ignores pairwise relations among features and information about spatio-temporal order among features.

Oshin *et al*. [Oshin 2011] proposed another contextual features and they use the spatio-temporal distribution of features alone, *i.e*. without explicit appearance information. Their approach makes use of locations and strengths of interest points only, and it discards appearance information. In order to automatically discover and reject outlier samples within classes, they use Random Sampling Consensus (RANSAC). The main limitations of this technique are: (a) it does not use visual and motion appearance information, and (b) it ignores pairwise relations among features and information about spatio-temporal order among features.

In summary, most of the above contextual features based techniques use the discriminative power of individual features and capture local density of features in feature-centric neighbourhoods. To capture structural information in contextual features, the spatio-temporal grid has been applied in some of the above approaches, however the spatio-temporal grid is limited in terms of detailed description providing only a coarse representation. Moreover, the existing techniques ignore information about the spatio-temporal order among features. Therefore, a new representation is needed to create a finer description of contextual features.

2.3.3 Local Features Encoding

Once local features are extracted, they are used to represent videos - actions.

The most popular representation technique encoding local features is the bag-of-features model. The bag-of-features is a very popular representation used in Natural Language Processing, Information Retrieval, and also Computer Vision. It was originally proposed for document retrieval, where text is represented as the bag of its words (bag-of-words) [Salton 1968].

One of the first and important studies using bag-of-features model in Computer Vision are: Cula and Dana [Cula 2001] for texture classification, Sivic and Zisserman [Sivic 2003] for object and scene retrieval, Csurka *et al*. [Csurka 2004] for image categorization, Lazebnik *et al*. [Lazebnik 2006] for scene categorization, Sivic *et al*. [Sivic 2005] for object localization, and Schuldt *et al*. [Schuldt 2004], Dollar *et al*. [Dollar 2005], and Niebles *et al*. [Niebles 2006] for action recognition.

The bag-of-features model encodes global statistics of local features, computing a spatial histogram of local feature occurrences in a video sequence. Firstly, it creates a visual vocabulary using unsupervised learning over local features extracted from the training videos. The learning is typically done with k-means clustering algorithm. Then, the bag-of-features quantizes local features to a visual vocabulary, and it represents a video using histogram of quantized local features, followed by the L1 or the L2 norm; both norms are popular and there is no clear answer which one is the best. The advantage of the L1 norm is that it requires less computation time. The normalization step is applied to reduce effects of variable video size and variable number of detected local features in videos.

The bag-of-features model uses hard quantization of local features (*i.e*. uses histogram encoding) to represent local features. Recent approaches replace the hard quantization of local features with alternative encoding techniques that retain more information about the local features. This has been done in two ways: (1) by representing features as a combination of visual words (*e.g*. Kernel codebook encoding [Philbin 2008, Gemert 2008] and Locality-constrained Linear Coding [Wang 2010]), and (2) by representing differences between features and visual words (*e.g*. Fisher vector encoding [Perronnin 2010b], Super-vector encoding [Zhou 2010], BossaNova encoding [Avila 2013], and Vector of Locally Aggregated Descriptors encoding [Jegou 2010]). A good description of various encoding techniques is provided in [Chatfield 2011], where the encoding techniques are applied for object recognition (but can be applied for action recognition as well).

The following techniques are based on visual vocabulary, which is typically created in the same manner as in the bag-of-features model, unless otherwise stated.

Kernel codebook encoding [Philbin 2008, Gemert 2008] is a variant of the bag-

of-features model, where local features are assigned to visual vocabulary in a soft manner. The local features are associated with several nearby visual words instead of a single nearest visual word, and they are mapped to a weighted combination of visual words.

Locality-constrained Linear Coding [Wang 2010, Zhou 2013] is another variant of the bag-of-features approach. It projects each local feature into its local-coordinate system, and the projected coordinates are integrated by max pooling technique to generate the final representation. Features are projected down to the local linear subspace spanned by several closest visual words.

Fisher vector encoding (Fisher vectors) [Perronnin 2010b, Oneata 2013] does not represent features as a combination of visual words but instead it represents differences between features and visual words. Firstly, it creates a visual vocabulary by clustering local features extracted from the training videos, where clustering is done with Gaussian Mixture Model clustering. Then, it captures the average first and second order differences between local features and visual vocabulary, *i.e*. Gaussian components.

Super-vector encoding [Zhou 2010] is another variant of the Fisher encoding. There are two variants of the support vector encoding: (1) with hard assignment of local features to the nearest visual word, and (2) with soft assignment of local features to several nearest visual words. The visual vocabulary is created using k-means algorithm. Then, the video is encoded using (1) the first order differences between local features and visual words and (2) the components representing the mass of each visual word.

Vector of Locally Aggregated Descriptors (VLAD) encoding [Jegou 2010, Jain 2013] is another variant of the bag-of-features model. It accumulates the residual of each local feature with respect to its assigned visual word. Then, it matches each local feature to its closest visual word. Finally, for each cluster it stores the sum of the differences of the descriptors assigned to the cluster and the centroid of the cluster.

BossaNova encoding [Avila 2013] is very similar to the Vector of Locally Aggregated Descriptors encoding technique. It enriches the bag-of-features representation with a histogram of distances between the local features and visual words, preserving information about the distribution of the local feature around each visual word.

Most of the above techniques were invented for image classification, image retrieval, and object recognition. However, they can be applied for any domain and any task using local features.

Local Features Encoding: Memory Requirements

Let's denote the size of codebook as K and the size of local descriptors as D. Then:

- The size of the bag-of-features representation, Kernel codebook encoding, and Locality-constrained Linear Coding is K.
- The size of the Fisher vector encoding is $2KD$.
- The size of the VLAD encoding is KD.
- The size of the BossaNova encoding is $K(B+1)$, where B is the number of discretized distances between codewords and local descriptors [Avila 2013].

We observe that the bag-of-features, Kernel codebook encoding, and Locality-constrained Linear Coding representations require the smallest amount of memory to store a video sequence. The Fisher vector encoding requires the greatest amount of memory to store a video sequence.

Local Features Encoding: Accuracy

Various comparisons between local feature encoding techniques have been presented in the literature, *e.g.*:

- Chatfield *et al.* [Chatfield 2011] compared the bag-of-features, Kernel codebook encoding, Locality-constrained Linear Coding, Fisher vector encoding, and Super-vector encoding, and for the task of object recognition Fisher vector encoding gave the best results.
- Avila *et al.* [Avila 2013] compared the bag-of-features, BOSSA encoding [Avila 2011], BossaNova encoding (improved version of the BOSSA encoding), and Fisher vector encoding, and for the task of image classification Fisher vector encoding gave the best results (not counting the combination of the Fisher vector encoding and BossaNova encoding which shown superior results).
- Moreover, Jegou *et al.* [Jegou 2012] compared the bag-of-features, Fisher vector encoding and VLAD encoding, and for large-scale image search again Fisher vector encoding gave the best results.
- Krapac *et al.* [Krapac 2011] compared the bag-of-features and the Fisher vector encoding, and for image categorization again Fisher vector encoding gave the best results.
- For large-scale web video event classification, Sun and Nevatia [Sun 2013] presented that the Fisher vector encoding obtained better results than the bag-of-features and the VLAD encoding.
- Similarly, for the action recognition task, Oneata *et al.* [Oneata 2013] presented that the Fisher vector encoding obtained better results than the bag-of-features representation.

Local Features Encoding: Conclusion

The bag-of-features approach is the most popular technique for encoding local features and its representation requires a small amount of memory to store a video sequence. The recent Fisher vector encoding seems to be very powerful technique, it has shown superior results for many Computer Vision tasks, but its representation requires a large amount of memory to store a video sequence. Fortunately, it has been shown [Perronnin 2010b] that the Fisher vector encoding can be used with linear classifiers and it still outperforms the bag-of-features representation, which should be applied with non-linear classifiers to give a good classification performance.

Therefore, we will use both these encoding techniques, *i.e*. bag-of-features and Fisher vector encoding, in this thesis for the representation of local features and videos. We will evaluate, compare, and analyze them to better understand these models.

2.4 Classifiers

Once we represent video sequences, *e.g*. using any of the above techniques, we would like to decide which actions they contain. We are given a set of actions and our goal is to recognize these actions in videos.

There are many successful machine learning algorithms. If instances in a dataset are given with known labels, *i.e*. with the information about the correct output, then the learning is called supervised. If instances are unlabeled, then the learning is called unsupervised [Jain 1999]. K-means [MacQueen 1967], Gaussian Mixture Models [Reynolds 1995], Latent Semantic Analysis [Deerwester 1990], Probabilistic Latent Semantic Analysis [Saul 1997, Hofmann 1999], and latent Dirichlet allocation [Blei 2003]) are a few representative algorithms which belong to the unsupervised learning category. Reinforcement learning is another category of machine learning, where algorithms are taught what to do, how to map situations to actions to maximize a numerical reward signal [Sutton 1998]. The learner is not told which actions to take but it must discover which actions yield the most reward, by trying each action. Several algorithms which belong to the reinforcement learning category are: temporal difference [Sutton 1988] and Q-learning [Watkins 1992].

Due to the nature of our task, *i.e*. we know which actions we would like to recognize (instances available with labels), we only focus on supervised learning algorithms in this work. Due to a large number of machine learning algorithms, we only briefly present several popular classification algorithms.

The goal of the supervised learning is to build a model of the distribution of class labels in terms of input features. Then, the obtained classifier assigns class labels to the testing instances, where the values of the input features are known, but the value of the class label is unknown.

An excellent description of the supervised learning classification algorithms is provided in [Kotsiantis 2007], and we refer to that article for further information, comparisons, and references.

Statistical approaches [Jensen 1996] provide a probability that a given instance belongs to a particular class. Naive Bayes classifier is the simplest Bayesian classifier, which is based on Bayes' theory with strong (naive) assumption that all variables contribute toward classification and are mutually correlated. A Bayesian network is another classifier, it is a probabilistic graphical model that represents a set of random variables and their conditional dependencies via a directed acyclic graph, where the nodes are in one-to-one correspondence with the features.

Another examples of the graphical models are Hidden Markov Model (HMM) [Rabiner 1990] and Conditional Random Field (CRF) [Lafferty 2001]. The former is a generative model and it gives the output directly by modeling the transition matrix based on the training data. It assumes that the system being modeled is a Markov process with unobserved (hidden) states. The latter is a discriminative model which outputs a confidence measure. It can be considered as a generalization of HMM.

Instance based learners is another category of classifiers. One of the simplest classifier is the k-Nearest Neighbour (k-NN) [Cover 2006]. It locates the k nearest instances to the given query instance and determines its label by selecting the single most frequent label of nearest instances. The main limitation of this classifier is that it requires to store all the instances and it is sensitive to the choice of the similarity function to compare instances. Moreover, there is general agreement that it is very sensitive to irrelevant features.

There are many variants of the k-NN algorithm, *e.g.* CNN and UNN. Condensed nearest neighbor (CNN) [Hart 1968] is designed to reduce the data set for classification. It selects the set of prototypes from the training data to classify samples almost as accurately as the nearest neighbour with the whole data set. Another variant of the k-NN algorithm is the Universal Nearest Neighbors [Piro 2010], which is a boosting algorithm for inducing a leveraged k-NN rule. This rule generalizes the k-NN to weighted voting, *i.e.* the votes of nearest neighbors are weighted by means of real coefficients, where the weights (called leveraging coefficients) are iteratively learned from training data.

Decision trees [Murthy 1998] belong to logic category of classifiers. Decision trees are trees, which classify instances by sorting them based on feature values. Each node in a decision tree represents a test on a feature, each branch represents an outcome of the test, and each leaf node represents the class label. There are several measures for finding the best features for the construction of a decision tree: Information gain, Gain ratio, Gini index, ReliefF algorithm, Chi square, and others. However, no measure is significantly better than others. The construction of the optimal decision tree is an NP-complete problem. The popular decision tree algorithms are: Iterative Dichotomiser 3 (ID3), C4.5, and Classification and Regression Trees (CART).

The main limitation of the decision trees is that they tend to overfit the training data. Random forest classifier [Breiman 2001, Genuer 2008] solves this problem. It uses a multitude of decision trees and outputs the class label based on the votes from all the individual decision trees. Moreover, it uses a random selection of features to split each node.

Another category of classifiers is perceptron based techniques. Artificial Neural Networks (ANNs) [Rumelhart 1986, Zhang 2000] are multi-layer neural networks, which consist of a number of connected units (neurons). ANNs consist of three types of layers: input layer with input units which receive information to be processed, output layer with output units which give the result of the algorithm, and hidden layers with hidden units which process the data. An ANN learns the weights of the connections between neurons in order to determine the mapping between the input and the output. They are many types of ANNs: single layer perceptron, RBF network, DNNs, CNNs, and others. A single layer perceptron is the simpliest neural network based on a linear combination of a set of weights with the feature vector. A Deep Neural Network (DNN) is a neural network with at least one hidden layer of units between the input and output layers. A Radial Basis Function (RBF) is a three-layer feedback network, in which each hidden unit implements a radial activation function and each output unit applies a weighted sum of hidden units outputs. A Convolutional Neural Network (CNN) is another type of a neural network that can be directly applied on the raw input, thus automating the process of feature construction.

Boosting [Schapire 1999] is a machine learning meta algorithm, which creates a strong classifier from a set of weak classifiers. The algorithm iteratively learns weak classifiers and adds them to a final strong classifier with weights which are typically corresponding to the accuracy. After a weak classifier is added, the data is reweighted, and typically the correctly classified samples loose weight, and the misclassified samples gain weight so the boosting algorithm will focus on them in the next iteration step. A weak classifier is defined as a classifier which works at least as well as a random classifier. A strong classifier should be well correlated with the true classification. The popular boosting algorithms are: AdaBoost, GentleBoost, BrownBoost, LogitBoost, Bootstraping, and others.

Support Vector Machines (SVMs) [Vapnik 1995, Burges 1998, Cristianini 2010] belong to another category of classifiers. They are maximizing the distance between a hyperplane that separates two classes of data and instances on either side of it. They can perform linear separation and also non-linear separation using a kernel function. Moreover, they reach the global minimum and avoid ending in a local minimum, what may happen in other search algorithms such as neural networks. Finally, they typically provide very good results.

Many of the above classification techniques have been successfully applied to action recognition in videos, *e.g.* HMMs [Yamato 1992], k-NN [Efros 2003, Blank 2005, Thurau 2008], ANNs [Iosifidis 2012], CNNs [Karpathy 2014], Boosting [Nowozin 2007, Fathi 2008], and SVMs [Dollar 2005, Laptev 2005, Laptev 2008, Liu 2008, Wang 2011a]. Over the last years, SVMs is the most popular classification technique used in action recognition in videos.

All the above classification algorithms have pros and cons and we refer to the work of Kotsiantis [Kotsiantis 2007] for the details. According to that work, SVMs achieve the best accuracy in general, in comparison with the Decision Trees, Neural Networks, Naive Bayes, k-NN, and Rule-learners. They are also at least as good as others in speed of classification, tolerance to irrelevant attributes, tolerance to redundant attributes, and tolerance to highly interdependent attributes. However, there is no single learning algorithm that can uniformly outperform other techniques over all datasets. SVMs have a sound theoretical foundation, and they are considered as a "must try" [Wu 2008] as they are one of the most robust and accurate methods. However, SVMs also have cons [Kotsiantis 2007], *e.g.* their performance highly relies on the selection of an appropriate kernel function, they have low speed of learning w.r.t. the number of attributes and the number of instances, and they do not handle well model parameters. For action recognition in videos, SVMs are the most widely used supervised learning classifiers. They achieve very good results, there exist kernel functions that give good results (see Section 2.3.3), the number of instances and the number of attributes are typically not large (up to several thousands), and there are typically not many classifier parameters to learn. Therefore, for action recognition in videos we will also use SVMs.

Ensemble of Classifiers

The above supervised learning techniques use an individual method to perform a classification. Another type of approaches creates an ensemble of classifiers to obtain better predictive performance. Over the last years, numerous methods have been proposed for that [Kotsiantis 2007, Dietterich 2000, Rokach 2010], and these methods typically use: (a) various subsets of training data with a single learning approach, (b) various training parameters with a single training approach, and/or (c) various learning approaches. Although many ensemble methods have been proposed, there is no clear picture which technique is the best [Kotsiantis 2007, Vilalta 2002]. An ensemble of classifiers have been used by several action recognition approaches, *e.g.* [Yang 2012, Izadinia 2012, Oh 2014]. Finding the right ensemble method is still an open machine learning research problem, that we do not try to solve it here. An ensemble of classifiers may improve results for some features, techniques, and decrease results for others. Therefore, to better understand the first steps of action recognition (*i.e.* feature extraction and video-action representation) we will only use SVMs while performing our action recognition experiments.

2.5 Conclusion

We presented the most relevant and the most prominent state-of-the-art research techniques related to our work. All the techniques have shown promising results. The presented techniques can be divide into three main categories:

- Human body model based methods - these techniques are based on the extraction of 2D or 3D information on human body parts, which is very difficult to achieve in realistic and unconstrained videos. Although cost-effective depth cameras help in the extraction of human body joint points, they also bring large limitations, *e.g.* to the range of the depth sensor, and this is exactly why we do not use them in this thesis.
- Holistic methods - these techniques are based on people localization, which is often done using segmentation. Precise segmentation is very difficult to obtain in realistic and unconstrained videos.
- Local feature methods - these techniques are based on local features and no information about human body model or people localization is required. These techniques can be applied to any scenario, therefore we mainly focus on local feature methods in this thesis.

Local feature methods have shown very good results across many and various datasets. These techniques usually consist of three phases:

1. Extraction of local features - based on popularity and published results we selected:
   - The Harris3D detector as the spatio-temporal interest point detector. This Harris3D detector is typically used with the HOG and HOF descriptors.
   - The dense trajectories as the trajectories detector. The dense trajectories are typically used with the Trajectory shape descriptor, HOG, HOF, and MBH descriptors.
2. Video - action representation - based on popularity and published results we selected:
   - The bag-of-features approach, which is the most popular technique for encoding local features and its representation requires a small amount of memory to store a video sequence.
   - Fisher vector encoding, which seems to be a very powerful technique, it has shown superior results for many Computer Vision tasks, but its representation requires a large amount of memory to store a video sequence.
3. Video - action classification - based on popularity and published results we selected the Support Vector Machines (SVMs) as a classifier, and we use the recommended kernels for SVMs, *i.e.*:
   - χ^2 kernel for the bag-of-features approach.

- linear kernel for the Fisher vector encoding.

The limitations of the state-of-the-art and our main contributions are as follows:

- There are several unanswered questions related to local features, such as: What is the accuracy of the Spatio-Temporal Interest Points and the Dense Trajectories using the bag-of-features approach and Fisher vector encoding? Moreover, does the bag-of-features with the L1 or the L2 norm give better results? Therefore, in Chapter 3 we present an extended evaluation, comparison, and analysis of local feature methods to answer the above questions.

- The popular state-of-the-art descriptors, *i.e*. Histogram of Oriented Gradients, Histogram of Optical Flow, and Motion Boundary Histograms, are based on a 1-dimensional histogram representation of individual features, and they directly model values of given features. The joint statistics between individual features are ignored, whereas such information may be informative. Therefore, we propose two novel descriptors to model relationships between different pixel-level appearance features such as intensity or gradient. We propose a method to compute these representations on space-time volumes extracted from a video sequence using the dense trajectories.

  - In Chapter 4 we present a new descriptor called Video Covariance Matrix Logarithm, which is based on a covariance matrix, and it models linear relationships between pixel-level features.
  - In Chapter 5 we present a new descriptor called Video Brownian Covariance, which is based on a Brownian covariance, a natural extension of the classical covariance measure, and it measures all types of dependence between pixel-level features in an arbitrary dimension.

- The existing local feature encoding techniques ignore: information about the spatio-temporal positions of features, relations among the features, and local densities of features, whereas such information may be very useful for action recognition. Therefore, we propose three techniques to overcome these limitations:

  - In Chapter 6 we introduce the idea of person-centric dense trajectories. We focus on short motion trajectories (often called tracklets), which in a natural way describe moving objects in a video sequence. The main idea is that action recognition ought to be performed using a dynamic coordinate system corresponding to an object of interest. Therefore, we propose an object-centric local feature representation of motion trajectories, which allows to add spatial information of features to a local feature encoding technique.
  - The existing pairwise features based techniques use the discriminative power of individual features and capture visual relations among features. However, they ignore information about the spatio-temporal geometric relations between features (*i.e*. Δx, Δy, Δt) and the spatio-temporal order between features.

Moreover, some of the above techniques can only handle small codebooks, and they ignore associations between geometric and appearance relations among features. Therefore, in Chapter 7 we propose a new representation of pairwise features, called Geometric and Appearance Relations of Pairwise Features (GARPF). Our pairwise features capture statistics of pairwise co-occurring local spatio-temporal features. They encode geometric and appearance relations among features in a single descriptor. Calculating video representations with different geometrical arrangements among the features, we preserve an important associations between appearance and geometric information. Our video representation captures not only global distribution of features but also focuses on geometric and appearance (visual and motion) relations among the features.

- The existing contextual features based techniques use the discriminative power of individual features and capture local densities of features in feature-centric neighborhoods. To capture structural information they use the spatio-temporal grids; however the spatio-temporal grid is limited in terms of detailed description providing only a coarse representation. Moreover, the existing techniques ignore information about the spatio-temporal order among features. Therefore, in Chapter 8 we propose a new, much finer representation of contextual features, called Spatio-Temporal Ordered Contextual Features (STOCF), that overcomes the above limitations. We propose contextual features which capture both local densities of features and statistics of space-time ordered features. Our representation encodes information about the order of local features.

- In Chapter 9 we present a comparison of all the proposed methods. We compare our techniques with each other and with state-of-the-art approaches, and we present an analysis of the results. Using the experiments, we try to find an answer to the question "When and which approach should we apply depending on videos and actions?".

CHAPTER 3

Action Recognition Framework, Its Evaluation, and Analysis

Contents

3.1 Introduction 48
3.2 Action Recognition Framework 48
3.2.1 Local Spatio-Temporal Video Features 49
3.2.2 Video-Action Representation 55
3.2.3 Video-Action Recognition 59
3.2.4 Action Recognition Approach Assessment 64
3.3 Datasets 65
3.3.1 Weizmann Dataset 66
3.3.2 KTH Dataset 67
3.3.3 URADL Dataset 69
3.3.4 MSR Daily Activity 3D Dataset 70
3.3.5 HMDB51 Dataset 73
3.3.6 CHU Nice Hospital Dataset 76
3.3.7 Datasets Summary 79
3.4 Experiments, Comparison, and Analysis 80
3.4.1 Spatio-Temporal Interest Points 81
3.4.2 Dense Trajectories 89
3.4.3 Summary and Conclusion 96
3.5 Conclusion 97

In this chapter we introduce the action recognition framework, which is used throughout this thesis work, embedding the novel algorithms for action recognition. It consists of three steps: local spatio-temporal video features extraction, video-action representation, and video-action recognition. We review the popular techniques for each of the steps. Then, we present five state-of-the-art action recognition datasets, and we propose a new, locally collected action recognition dataset. The dataset are presented with the statistical analysis, and they are used in the following evaluations. Finally, we show the extensive evaluation, comparison, and analysis of the presented techniques.

3.1 Introduction

Over the last decade, many different action recognition techniques have been proposed, most of which belong to the group of local feature based methods (Section 2.3). Action recognition based on local features is one of the most active research topics.

The local feature based methods have shown good accuracy in action recognition over various datasets [Bilinski 2011, Wang 2009, Stöttinger 2010]. Local features are able to capture appearance and motion. They are robust to viewpoint and scale changes. Moreover, they are easy to implement and quick to calculate.

In this chapter, we introduce the action recognition framework based on local features, which is used throughout this thesis work, embedding the novel algorithms for action recognition. We review the most popular state-of-the-art techniques for each of the steps of the typical action recognition framework based on local features (local spatio-temporal video features extraction, video-action representation, and video-action recognition).

Then, we present five popular state-of-the-art action recognition datasets that we use in the following evaluations, and we present the statistical analysis of these datasets. The datasets vary in challenges, the number of action categories, videos, and people participating in actions. We start with relatively simple one person actions and we finish with complicated multi person actions. Moreover, we propose a new, locally collected action recognition dataset, the CHU Nice Hospital dataset.

Finally, we present the extensive evaluation, comparison, and analysis of the presented techniques.

The remainder of the chapter is organized as follows. In Section 3.2, we describe our action recognition framework. Section 3.3 presents five popular state-of-the-art action recognition datasets, and our CHU Nice Hospital dataset. The datasets are presented along with their statistical analysis. In Section 3.4, we present experimental results, comparison, and analysis. Finally, we conclude in Section 3.5.

3.2 Action Recognition Framework

In this section, we present our action recognition framework, which is used throughout this thesis work, embedding the novel algorithms. The action recognition framework consists of three following consecutive steps (see Figure 3.1):

1. **Local spatio-temporal video features extraction** (Section 3.2.1) – we apply a feature detector to extract local spatio-temporal video volumes from a given video sequence, and we represent each local spatio-temporal video volume by a feature descriptor. In this thesis, we use the Spatio-Temporal Interest Points and the Dense Trajectories.

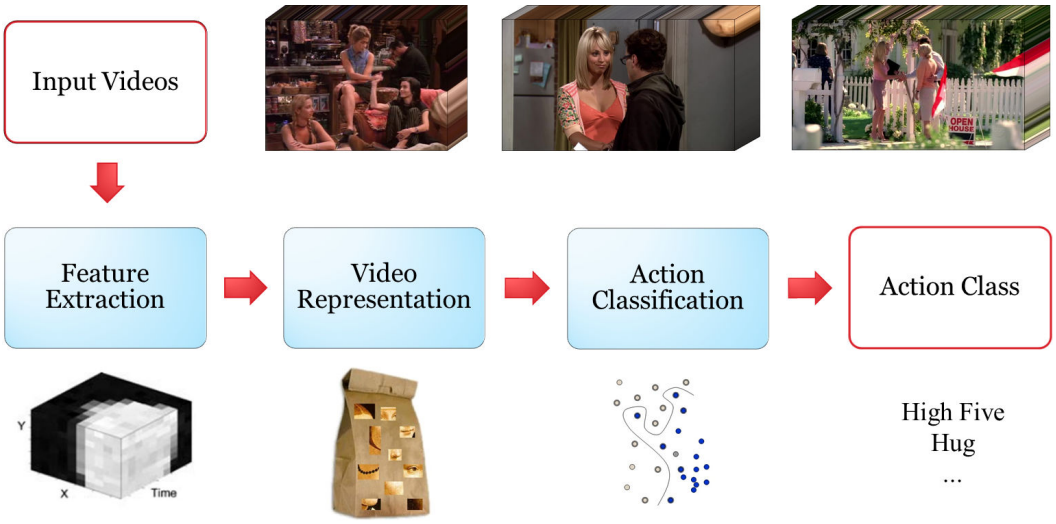


Figure 3.1 – Overview of our action recognition framework. Firstly, we extract local spatio-temporal features from a given video sequence. Then, we represent the given video sequence using a local feature encoding technique and the extracted local spatio-temporal video features. Finally, we apply a classifier to determine the action class for the given video.

2. **Video-action representation** (Section 3.2.2) – we create a video representation using a local feature encoding technique and the extracted local spatio-temporal video features. In this work, we use the bag-of-features approach and the Fisher vector encoding to represent videos.

3. **Video-action recognition** (Section 3.2.3) – we apply a classifier to determine the action class of a given video representation. In this thesis, we use the multi-class Support Vector Machines, with the exponential χ^2 kernel for the bag-of-features approach, and the linear kernel for the Fisher vector encoding.

The information about the training and the assessment of an action recognition approach is included in Section 3.2.4.

3.2.1 Local Spatio-Temporal Video Features

This section describes local spatio-temporal video feature detectors and descriptors.

Two methods were selected based on their use in the literature and good results provided by their authors: Spatio-Temporal Interest Points and Dense Trajectories.

- The Spatio-Temporal Interest Points (Section 3.2.1.1) were proposed by Laptev [Laptev 2003, Laptev 2008]. Firstly, the Harris3D detector is applied to extract points of interest, and then neighbourhood of each point is described by two descriptors: Histogram of Oriented Gradients and Histogram of Optical Flow.

- The Dense Trajectories (Section 3.2.1.2) were proposed by Wang *et al.* [Wang 2011a]. They are based on dense sampling of feature points and on tracking detected points in subsequent video frames. Then, the neighbourhood around each trajectory is described by four descriptors: Trajectory Shape, Histogram of Oriented Gradients, Histogram of Optical Flow, and Motion Boundary Histogram.

3.2.1.1 Spatio-Temporal Interest Points

Spatio-Temporal Interest Points - Detector: Harris3D

The Harris detector is an interest point detector for images, which was proposed by Harris and Stephens [Harris 1988]. It finds locations in a spatial image, where image values have significant variations in both directions (x and y).

The Harris3D detector is an interest point detector for videos, which was proposed by Laptev and Lindeberg [Laptev 2003]. It is the extension of the Harris detector into the spatio-temporal domain by requiring the video values in space-time to have large variations in both the spatial and the temporal dimensions.

The Harris3D detector calculates a spatio-temporal second-moment matrix, which is a 3×3 matrix composed of first order spatial and temporal derivatives averaged with a Gaussian weighting function $g(\cdot; s\sigma_l^2, s\tau_l^2)$:

$$\mu = g(\cdot; s\sigma_l^2, s\tau_l^2) * \begin{pmatrix} L_x^2 & L_xL_y & L_xL_t \\ L_xL_y & L_y^2 & L_yL_t \\ L_xL_t & L_yL_t & L_t^2 \end{pmatrix}, \tag{3.1}$$

where $s\sigma_l^2$ and $s\tau_l^2$ are the integration scales which relate to the local scales σ_l^2 and τ_l^2, and the first-order derivatives of the video sequence f are defined as:

$$L_x(\cdot; \sigma_l^2, \tau_l^2) = \partial_x(g * f), \tag{3.2}$$

$$L_y(\cdot; \sigma_l^2, \tau_l^2) = \partial_y(g * f), \tag{3.3}$$

$$L_t(\cdot; \sigma_l^2, \tau_l^2) = \partial_t(g * f). \tag{3.4}$$

The spatio-temporal separable Gaussian kernel g is defined as:

$$g(x, y, t; \sigma^2, \tau^2) = \frac{1}{\sqrt{(2\pi)^3\sigma^4\tau^2}} \times \exp\left(-\frac{(x^2+y^2)}{2\sigma^2} - \frac{t^2}{2\tau^2}\right). \tag{3.5}$$

The Harris3D detector searches for regions that have significant eigenvalues λ_1, λ_2, and λ_3 of the matrix μ. The spatio-temporal points are detected as local positive spatio-temporal maxima of H:

$$H = \det(\mu) - k\,\mathrm{trace}^3(\mu) = \lambda_1\lambda_2\lambda_3 - k(\lambda_1 + \lambda_2 + \lambda_3)^3, \tag{3.6}$$

which combines the determinant $\det(\cdot)$ and the trace $\mathrm{trace}(\cdot)$ of the matrix μ.

The authors also defined normalized spatio-temporal Laplace operator to select the spatio-temporal scales of points:

$$\Delta^2_{norm} L = L_{xx,norm} + L_{yy,norm} + L_{tt,norm} = \sigma^2\tau^{1/2}(L_{xx} + L_{yy}) + \sigma\tau^{3/2}L_{tt}, \tag{3.7}$$

where $L_{xx,norm}$, $L_{yy,norm}$, and $L_{tt,norm}$ are the second-order derivatives L_{xx}, L_{yy}, and L_{tt} of L normalized by the scale parameters.

The final spatio-temporal interest points returned by the Harris3D detector are the local maxima of the Harris cornerness criterion (Equation 3.6) and the local extrema of the normalized spatio-temporal Laplace operator (Equation 3.7).

In [Laptev 2008] the authors have proposed to use a multi-scale approach and extract features at multiple levels of spatio-temporal scales instead of performing scale selection. This reduces the computational complexity and the Harris3D detectors still provides a good recognition performance using dense scale sampling.

In the following evaluations, we use the default parameters of the Harris3D detector provided by the authors [Laptev 2008] [1], *i.e.* $k = 0.0005$, $\sigma^2 \in \{4, 8, 16, 32, 64, 128, 256, 512\}$, and $\tau^2 \in \{2, 4\}$. Sample video frames with the extracted Harris3D interest points are presented in Figure 3.2.

Spatio-Temporal Interest Points - Descriptors: HOG and HOF

The Histogram of Oriented Gradients (HOG) and Histogram of Optical Flow (HOF) descriptors for the Spatio-Temporal Interest Points were proposed by Laptev *et al.* [Laptev 2008]. The HOG descriptor for the Spatio-Temporal Interest Points is a variant of the HOG descriptor for human detection, which was introduced by Dalal and Triggs [Dalal 2005]. The HOG encodes visual appearance and shape information of a local feature; the edge orientations are computed in the neighbourhood of a detected point and quantized into 4 bins (4 directions). The HOF encodes motion information of a local feature; the optical flow is computed in the neighbourhood of a detected point and quantized into 5 bins (4 directions with an additional zero bin).

The authors compute descriptors of space-time volumes in the neighborhood of detected points. The size of a local space-time volume (Δx,Δy,Δt) is related to the

[1] http://www.di.ens.fr/~laptev/download.html#stip

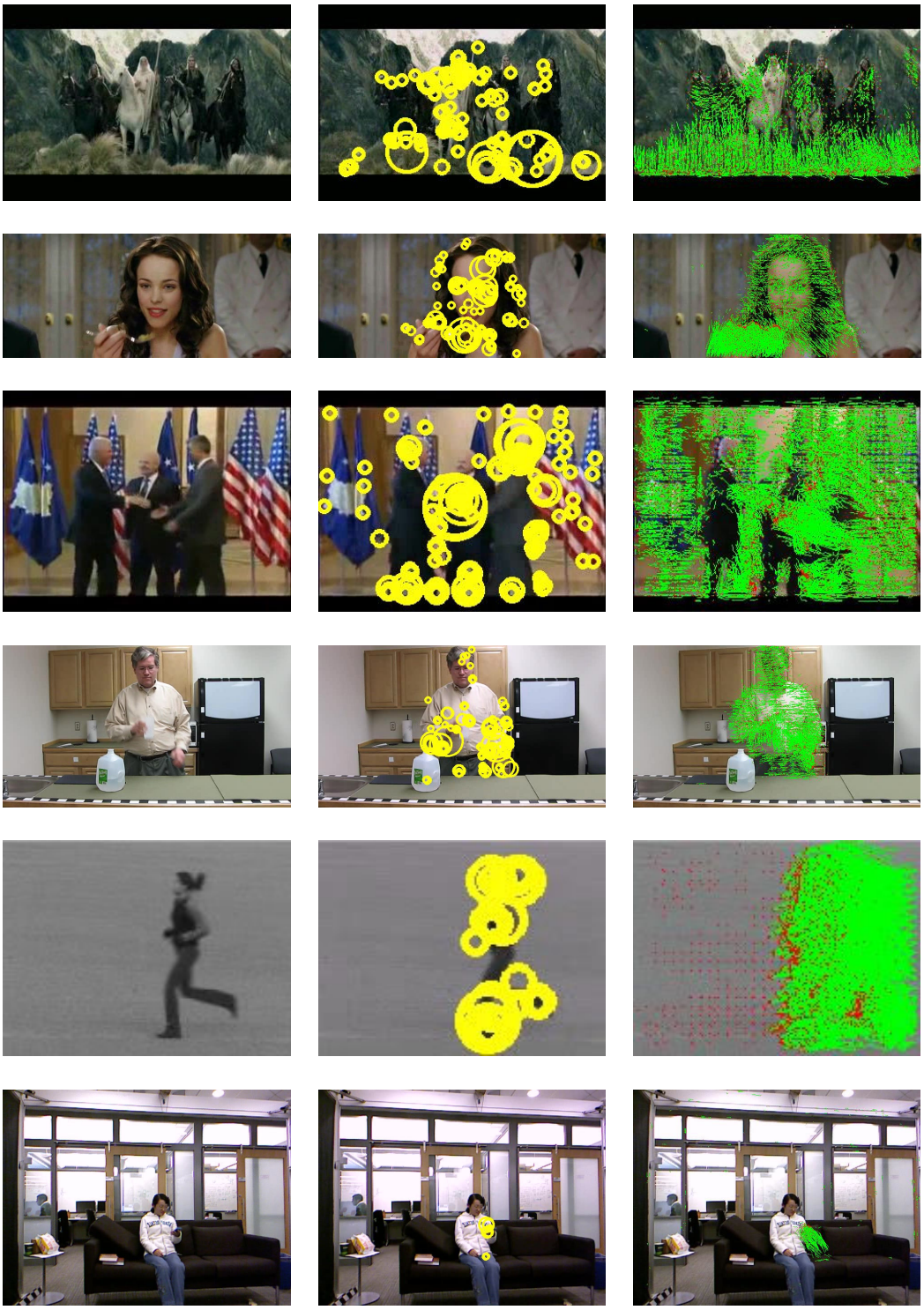


Figure 3.2 – Sample video frames (first column) with the extracted Harris3D interest points (second column) and the Dense Trajectories (third column) presented on four datasets: HMDB51 (first three rows), URADL (fourth row), KTH (fifth row), and MSR Daily Activity 3D (sixth row).

detection scales by Δx, $\Delta y = 2k\sigma$, and $\Delta t = 2k\tau$. To embed structure information, each local volume is subdivided into a grid with $n_x \times n_y \times n_t$ spatio-temporal cells; for each cell of the grid, coarse histograms are computed. Then, the normalized histograms from cells are concatenated into the final descriptors.

In the following evaluations, we use the default parameters provided by the authors, *i.e.* $k = 9$, n_x, $n_y = 3$, and $n_t = 2$. Therefore, the size of the HOG descriptor is 72 and the size of the HOF descriptor is 90.

3.2.1.2 Dense Trajectories

Dense Trajectories - Detector

The Dense Trajectories were introduced by Wang *et al.* [Wang 2011a]. They are based on a dense sampling of feature points and on tracking the detected points in subsequent video frames.

Dense Trajectories - Detector: Dense Sampling

Feature points are sampled in each frame of a video, on a dense grid, with the distance between points (*i.e.* the step size) of W pixels. The parameter W is usually set to 5 pixels, which allows to obtain sufficiently dense trajectories and catch significant motion in a video. It is difficult to determine the best scale(s) to track feature points; therefore, the dense sampling is applied on multiple spatial scales, on each spatial scale separately, to ensure extraction of meaningful features. Depending on the video resolution, at most 8 spatial scales are computed. The spatial scale increases by a factor of $1/\sqrt{2}$. This approach guarantees that feature points equally cover spatial positions and scales of a video sequence.

Typically, feature point tracking algorithms try to capture a structure around a feature point. However, in homogeneous regions it is just not possible. Therefore, the criterion of [Shi 1994] is used to remove points in homogeneous areas, *i.e.* the points on the grid are removed, if the eigenvalues of the auto-correlation matrix are very small, smaller than a threshold T:

$$T = 0.001 \times \max_{i \in I} \min(\lambda_i^1, \lambda_i^2), \tag{3.8}$$

where λ_i^1 and λ_i^2 are the eigenvalues of the point i in the image I. The parameter 0.001 was set experimentally, and it represents a good compromise between saliency and density of the sampled points.

In the following evaluations, we use the default parameter W provided by the authors, *i.e.* $W = 5$.

Dense Trajectories - Detector: Point Tracking

Once local feature points are extracted, they are tracked in the subsequent video frames. Feature points are detected on multiple spatial scales and they are tracked on each spatial scale separately.

Firstly, dense optical flow field w_t is computed from frame I_t to the following frame I_{t+1}, using the algorithm proposed by Farneback in [Farnebäck 2003]. Then, given the point $P_t = (x_t, y_t)$ in a frame I_t, its new position in the next frame I_{t+1} is calculated as:

$$P_{t+1} = (x_{t+1}, y_{t+1}) = (x_t, y_t) + (M * w_t)|_{(x_t, y_t)}, \tag{3.9}$$

where M is the median filtering kernel [Sundaram 2010] of size 3×3, and $w_t = (u_t, v_t)$ is the dense optical flow field, where u_t and v_t are the horizontal and vertical components of the optical flow.

Short trajectories are necessary to recognize short actions and gestures, such as smiling and doing a high five. Moreover, it has been shown that short trajectories are more robust in case of presence of fast irregular motions. Therefore, the length of trajectories is limited to L frames, which was set experimentally to $L = 15$ frames.

Moreover, static trajectories are removed as they do not contain any motion information. A trajectory is classified as static if the standard deviation of horizontal and vertical positions is less than σ_{min} (both standard deviations have to be lower than the threshold).

Also, trajectories with sudden large displacements, which are most likely to be erroneous, are removed. A trajectory is classified as a trajectory with sudden large displacement if the standard deviation of horizontal or vertical positions is grater than σ_{max} (any standard deviation has to be grater than the threshold), or if the displacement vector between two consecutive frames is larger than T_{max} of the overall displacement of this trajectory.

As a result of the above steps, sometimes there is no tracked point in $W \times W$ neighborhood, and in that case, a new feature point is sampled and tracked in the subsequent video frames.

In the following evaluations, we use the default parameters provided by the authors [Wang 2011a] [2], *i.e.* $L = 15$, $\sigma_{min} = \sqrt{3}$, $\sigma_{max} = 50$, and $T_{max} = 70\%$.

[2] http://lear.inrialpes.fr/~wang/dense_trajectories (ver. 1.1)

Dense Trajectories - Descriptors: Trajectory shape, HOG, HOF, and MBH

The Trajectory shape descriptor was proposed by Wang *et al*. [Wang 2011a] to encode a shape of the extracted dense trajectories. It describes a shape of a trajectory by a sequence of displacement vectors normalized by the sum of displacement vector magnitudes.

Three descriptors (HOG, HOF, and MBH) are computed within a space-time volume around a trajectory. To embed structure information, each local volume is subdivided into a grid with $n_x \times n_y \times n_t$ spatio-temporal cells; for each cell of the grid, a histogram descriptor is calculated. Then, the histograms are normalized with their L_2 norm, and the normalized histograms from cells are concatenated into the final descriptors.

Similarly to the Spatio-Temporal Interest Points, the HOG and HOF descriptors are computed. In this case, the orientations (*i.e*. edge and optical flow orientations) are quantized into 8 bins using full orientations, with an additional zero bin for the HOF descriptor.

The Motion Boundary Histogram (MBH) descriptor was introduced by Dalal *et al*. [Dalal 2006] for human detection and was proposed for dense trajectories by Wang *et al*. [Wang 2011a]. The MBH descriptor separates the optical flow field $I_w = (I_x, I_y)$ into its x and y component. Spatial derivatives are computed separately for the horizontal and vertical components of the optical flow, and orientation information is quantized into histograms, similarly to the HOG descriptor. The MBH descriptor encodes the relative motion between pixels. Constant motion information is suppressed and only information about changes in the flow field (*i.e*. motion boundaries) is kept, what eliminates noise due to background motion.

In the following evaluations, we use the default parameters provided by the authors, *i.e*. the spatial size of the volume is 32×32, n_x, $n_y = 3$, and $n_t = 2$. Therefore, the size of the Trajectory shape descriptor is 30, the size of the HOG descriptor is 96, the size of the HOF descriptor is 108, and the size of the MBH descriptor for the horizontal (vertical) component is 96.

3.2.2 Video-Action Representation

Once local features are extracted, they are used to represent videos - actions.

This section describes two video-action representation models that we selected based on their use in the literature and good results provided by many authors (see Section 2.3.3). The selected techniques are bag-of-features and Fisher vectors.

The bag-of-features approach is the most popular technique for encoding local features (see Section 3.2.2.1). Over the last years it has demonstrated impressive levels of performance with local spatio-temporal video features.

The bag-of-features model computes a histogram of feature occurrences in a video sequence. However, due to the hard vector quantization, some loss of information might occur.

Therefore, Perronnin *et al*. [Perronnin 2007] have introduced a new model called Fisher vectors (see Section 3.2.2.2). Fisher vector encoding represents differences between features and visual words, and it has shown to improve recognition results over the bag-of-features representation [Chatfield 2011, Avila 2013, Oneata 2013].

3.2.2.1 Bag-of-Features

The bag-of-features is a very popular representation used in Natural Language Processing, Information Retrieval, as well as in Computer Vision. It operates on extracted local features and a video is represented as a bag of its local features, disregarding their order, but keeping multiplicity. This representation encodes global statistics of local features, computing a histogram of feature occurrences in a video sequence. The bag-of-features model is commonly used in action recognition, where the frequency of local feature occurrence is calculated and used for training a classifier.

In the first step, the bag-of-features model builds a visual vocabulary, called codebook. The codebook is generated using local features extracted from the training videos. Local features extracted from the testing videos are not used in the process of creating the codebook.

Typically, the codebook is generated using the k-means algorithm. The k-means algorithm was first used by James B. MacQueen [MacQueen 1967]. It is an unsupervised learning algorithm that partition given a set of n feature vectors (f_1, f_2, ..., f_n) into k clusters ($k \leq n$, where k is fixed a priori), in which each feature vector belongs to the cluster with the nearest mean. The objective of the k-means clustering technique is to minimize the sum of the squares of the distances between the features and their closest cluster centers:

$$\arg\min_{\mathbf{C}} \sum_{i=1}^{k} \sum_{f_j \in C_i} \|f_j - \mu_i\|^2 , \tag{3.10}$$

where μ_i is the mean of feature vectors belonging to the cluster C_i.

An important element of the k-means clustering algorithm is to select the appropriate distance between a feature vector and the center of a cluster (*i.e.* $\|f_j - \mu_i\|$ of Eq. 3.10). Although there are many state-of-the-art distance measurements (*e.g.* Euclidean, City-block (also called Manhattan), Chebychev, Cosine, Jaccard, Power, Percent disagreement), typically the best results are achieved by using the Euclidean distance.

After generating the visual vocabulary (codebook), every video can be represented by the bag-of-features model. The bag-of-features model represents a video sequence by assigning its features to the nearest elements of the created visual vocabulary, *i.e.* to the nearest cluster centers.

Therefore, given a set of local features (f_1, f_2, ..., f_n) extracted from a video sequence, we assign each feature vector to the closest cluster center using the nearest neighbour algorithm. We use the same distance between a feature vector and a center of a cluster that was used to create the codebook. Local features assigned to the i-th cluster are defined as:

$$C_i = \left\{ f_k \mid i = \arg\min_j \|f_k - \mu_j\|^2 \right\}, \tag{3.11}$$

where μ_j is the center of the j-th cluster C_j.

The bag-of-features model represents a video sequence as a histogram of local feature occurrences, and more precisely as a k-elements feature vector $H = \{H_1, H_2, ..., H_k\}$ of quantized local features, where k is the number of codebook elements. Each element of the feature vector H represents the number of features assigned to the corresponding cluster ($H_i = |C_i|$).

Finally, we normalize the histogram representation so that the video size does not significantly change the bag-of-features magnitude. The two popular normalization methods that we consider, compare, and analyze in the experimental section are:

- L1 norm:
$$H_{L1} = \frac{H}{(||H||_1 + e)}, \tag{3.12}$$
- L2 norm:
$$H_{L2} = \frac{H}{\sqrt{||H||_2^2 + e^2}}, \tag{3.13}$$

where H is the non normalized histogram representation, $||H||_k$ is the k-norm of the vector H (where $k \in \{1, 2\}$), and e is a very small constant (*e.g.* $e = 1e - 20$).

Typically, the bag-of-features representation is used together with the Support Vector Machines using exponential χ^2 kernel, which has shown very good results for action recognition in videos [Laptev 2008, Marszalek 2009, Wang 2011a]. The exponential χ^2 kernel combines the advantages of the χ^2 kernel (which is designed to compare histograms), and the exponential kernel (which nonlinearly maps samples into a higher dimensional space to handle nonlinear relations between class labels and attributes). More details are presented in Section 3.2.3.2 and Section 3.2.3.3.

3.2.2.2 Fisher Vectors

In Computer Vision, the most popular image and video representation model is the bag-of-features approach with the histogram encoding.

Recently, Perronnin *et al.* [Perronnin 2007, Perronnin 2010a, Perronnin 2010b] have introduced a new model called Fisher vectors (or Fisher vector encoding), which is an extension of the bag-of-features representation. Fisher vector encoding has shown to achieve excellent results as a global descriptor both for image classification [Perronnin 2010b] and for image retrieval [Perronnin 2010a], outperforming the common bag-of-features approach. It encodes a video sequence using first and second order statistics of a distribution of a feature set $\mathbb{X}$. Fisher vector encoding models features with a generative model and computes the gradient of their likelihood with respect to the parameters of the model, *i.e.* $\Delta_\lambda \log p(\mathbb{X}|\lambda)$. It describes how the set of features deviates from an average distribution of features, modeled by a parametric generative model.

Firstly, during the preliminary learning stage, we fit a M-centroid Gaussian Mixture Model (GMM) to our training features, which can be regarded as a soft visual vocabulary:

$$p(x_i|\lambda) = \sum_{j=1}^{M} w_j g(x_i|\mu_j, \Sigma_j), \tag{3.14}$$

where $x_i \in \mathbb{X}$ is a D-dimensional feature vector, $\{g(x_i|\mu_j, \Sigma_j)\}_{j=1}^{M}$ are the component Gaussian densities, and $\lambda = \{w_j, \mu_j, \Sigma_j\}_{j=1}^{M}$ are the parameters of the model, *i.e.* the mixture weights $w_j \in \mathbb{R}_+$, the mean vector $\mu_j \in \mathbb{R}^D$, and the positive definite covariance matrices $\Sigma_j \in \mathbb{R}^{D\times D}$ of each Gaussian component, respectively. The Gaussian g is defined as:

$$g(x_i|\mu_j, \Sigma_j) = \frac{1}{\sqrt{(2\pi)^D |\Sigma_j|}} \exp\left\{-\frac{1}{2}(x_i - \mu_j)^T \Sigma_j^{-1}(x_i - \mu_j)\right\}, \tag{3.15}$$

and we require:

$$\forall_j : w_j \geq 0, \quad \sum_{j=1}^{M} w_j = 1, \tag{3.16}$$

to ensure that g is a valid distribution. We learn the parameters λ using the Expectation Maximization restricting the covariance of the distribution to be diagonal. We define the soft assignment of a descriptor x_i to the Gaussian j as the posteriori probability $\gamma(j|x_i, \lambda)$ for the component j:

$$\gamma(j|x_i, \lambda) = \frac{w_j g(x_i|\mu_j, \Sigma_j)}{\sum_{l=1}^{M} w_l g(x_i|\mu_l, \Sigma_l)}. \tag{3.17}$$

Then, we compute the gradients of the j-th component with respect to the mean vector μ_j and the variance vector σ_j^2, using the following derivations:

$$G^{\mathbb{X}}_{\mu,j} = \frac{1}{N_x\sqrt{w_j}} \sum_{l=1}^{N_x} \gamma(j|x_l,\lambda) \left(\frac{x_l - \mu_j}{\sigma_j} \right), \tag{3.18}$$

$$G^{\mathbb{X}}_{\sigma,j} = \frac{1}{N_x\sqrt{2w_j}} \sum_{l=1}^{N_x} \gamma(j|x_l,\lambda) \left(\frac{(x_l - \mu_j)^2}{\sigma_j^2} - 1 \right), \tag{3.19}$$

where N_x is the cardinality of the set $\mathbb{X}$. Finally, we encode the set of local descriptors $\mathbb{X}$ as a concatenation of partial derivatives with respect to the mean $G^{\mathbb{X}}_{\mu,j}$ and standard deviation $G^{\mathbb{X}}_{\sigma,j}$ parameters for all M components:

$$V = [G^{\mathbb{X}}_{\mu,1}, G^{\mathbb{X}}_{\sigma,1}, ..., G^{\mathbb{X}}_{\mu,M}, G^{\mathbb{X}}_{\sigma,M}]^T. \tag{3.20}$$

As a final step, we apply the power normalization and the L2 normalization:

- Power normalization: We apply the signed squared rooting, *i.e.* we apply the function $f(z) = \text{sign}(z)\sqrt{|z|}$ to each dimension of the vector V, where $\text{sign}(\cdot)$ is the signum function. The power normalization can be viewed as explicitly applying non-linear additive kernel, the Hellinger's kernel (also called the Battacharyya's kernel).
- L2 normalization: The obtained vector is further normalized by the L2 norm.

Both the power normalization and the L2 normalization have shown superior results [Perronnin 2010b] over the Fisher vector encoding without them [Perronnin 2010a].

Usually, the Fisher vector representation is applied with the linear Support Vector Machines [Perronnin 2010a, Perronnin 2010b]. Typically, if the number of features is large, there is no need to map data to a higher dimensial space (see Section 3.2.3.2), *i.e.* the non-linear mapping does not improve the accuracy [Hsu 2003].

The dimension of the Fisher vector representation is $2DM$, where D is the size of local descriptors and M is the size of the codebook. Therefore, the Fisher vector encoding can be seen as embedding local descriptors in a higher dimensional space, what allows to use the linear SVMs for classification of Fisher vectors.

Fisher vector representation with linear SVMs have shown to outperform the bag-of-features approach in many Computer Vision applications (see Section 2.3.3).

3.2.3 Video-Action Recognition

In this section, we focus on how to use the given video representations and classify them into action categories.

Support Vector Machines (SVMs) are among the most prominent machine learning algorithms that analyze data and recognize patterns. They are widely used for classification

and regression. SVMs are one of the most robust and accurate Machine Learning methods. They have a sound theoretical foundation and they have efficient training algorithms [Wu 2008].

SVMs have their origin in the work of Vapnik *et al.* [Vapnik 1979] from the late seventies, but they have attracted wide attention since the 1990s. An excellent description of SVMs is provided in [Vapnik 1995] and [Cortes 1995].

The aim of the SVMs is to find the optimal hyperplane which separates two classes of data. Depending on the nature of the data, such a separation might be linear or non-linear.

SVMs belong to the supervised learning algorithms. It means that they use training samples, where each training sample is a pair of an input object (typically a vector) and a desired output value (class label). The SVMs analyze the training data and build an inferred function, that can be used to correctly determine the class label for an unseen input object.

3.2.3.1 Linear Support Vector Machines

Linear SVMs are one of the most popular SVMs for classification and regression. Linear SVMs have shown to provide very good and promising results with high-dimensional data (such as Fisher vectors) in solving many key Computer Vision problems, such as: face verification [Simonyan 2013], image retrieval [Perronnin 2010b], object detection [Cinbis 2013], and action and event recognition [Oneata 2013, Wang 2013b]. Moreover, linear SVMs have shown to be efficient both in the training and in the prediction step.

Let's denote the training data set by:

$$\mathcal{T} = \{(\mathbf{x}_i, y_i) | \mathbf{x}_i \in \mathbb{R}^p, y_i \in \{-1, +1\}\}_{i=1}^{n}, \tag{3.21}$$

where n is the number of instance-label pairs, $\mathbf{x}_i$ is a p-dimensional real vector, and y_i is either -1 or $+1$, indicating the class to which it belongs.

We use the C-SVM formulation introduced by Cortes and Vapnik [Cortes 1995]. The algorithm finds the optimal hyperplane, which best separates two classes of data by minimizing the following equation:

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n}\xi_i, \\ \text{subject to:} \quad & y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0, \\ & C > 0, \end{aligned} \tag{3.22}$$

where $\mathbf{w}$ is the normal vector to the optimal hyperplane, C is the regularization parameter, *i.e.* is the tradeoff between regularization and constraint violation, and ξ_i are the non-

negative slack variables which measure the degree of misclassification of the data $\mathbf{x}_i$.

One of the main advantage of the linear Support Vector Machines is that they are fast. Assuming that n is the number of training samples and p is the data dimensionality, training of the linear SVM can be done in $O(pn)$ and testing can be done in $O(p)$ (only a scalar product between the learnt weight vector $\mathbf{w}$ and the feature vector of a test sample $\mathbf{x}$ has to be computed) [Sreekanth 2010].

3.2.3.2 Non-Linear Support Vector Machines

Linear SVMs have shown to provide very good results in various linear problems. However, not all the problems can be solved by linear classifiers; *i.e.* data can be separated only by non-linear classifiers.

The SVMs task (Equation 3.22) can be also represented in the dual formulation form. We use a kernel function $K(\cdot,\cdot)$ and map the input space to a new space, called feature space, $K(\mathbf{x},\mathbf{z}) : \mathbb{R}^p \times \mathbb{R}^p \rightarrow \mathbb{R}$, that is the inner product, $\Phi(\mathbf{x}) \cdot \Phi(\mathbf{z})$, in unrealized, possibly high-dimensional feature space. The feature space has more dimensions than the input space, so we could separate the data in the new space linearly.

The SVM task in the dual formulation form is represented by the following equation:

$$\begin{aligned} \underset{\boldsymbol{\alpha}}{\text{maximize}} \quad & \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{ij} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j), \\ \text{subject to:} \quad & 0 \leq \alpha_i \leq C, \\ & \sum_{i=1}^{n} \alpha_i y_i = 0, \end{aligned} \tag{3.23}$$

The decision function is $\text{sign}(h(\mathbf{x}))$, where:

$$h(\mathbf{x}) = \sum_{l=1}^{m} \alpha_l y_l K(\mathbf{x}, \mathbf{x}_l) + b. \tag{3.24}$$

The dual formulation only requires access to the kernel function and not to the features $\Phi(\cdot)$, allowing to solve it in very high-dimensional feature spaces efficiently, what is often called the kernel trick. The features $\mathbf{x}_l : l \in \{1, 2, ..., m\}$ are called the support vectors, which lie on the margin and satisfy $y_i(\mathbf{w} \cdot \mathbf{x}_i + b) = 1$.

There are many kernel functions for SVMs, *e.g.*: Polynomial, Gaussian, Exponential, Laplacian, Sigmoid, Power, Exponential Chi-Squared, and Histogram Intersection.

The most popular kernels in Computer Vision applications, such as in action recognition in videos, are the generalized Radial-Basis Function (RBF) kernels, *e.g.* the Exponential χ^2 kernel [Laptev 2008, Marszalek 2009, Wang 2011a]. Such kernels

combine the advantages of the homogeneous additive kernels (*e.g.* the χ^2 kernel and the intersection kernel) and the RBF kernels (*e.g.* the exponential (Gaussian) kernel) [Sreekanth 2010]. The former kernels compare probability distributions between vectors and are designed to compare histograms. The latter kernels can represent local templates and can nonlinearly map samples into a higher dimensional space to handle nonlinear relations between class labels and attributes [Hsu 2003].

One of the main challenges of using the Support Vector Machines with the generalized RBF kernels is high complexity when a large number of support vectors is needed. Assuming that n is the number of training samples and p is the data dimensionality, training of SVMs with such kernels is typically between $O(pn^2)$ and $O(pn^3)$, and testing is done typically in $O(pn)$, as each novel sample has to be compared to the support vectors determined during the training process, and they are usually of order n [Sreekanth 2010].

3.2.3.3 Fast Exponential χ^2 Kernel

The Exponential χ^2 Kernel is the generalized Radial-Basis Function (RBF) kernel (see Section 3.2.3.2). It combines the homogeneous additive kernel, *i.e.* the χ^2 kernel, and the RBF kernel, *i.e.* the exponential (Gaussian) kernel.

Let's denote two m-elements feature vectors as $\mathbf{h}_i$ and $\mathbf{h}_j$. Then, the χ^2 distance between these two vectors is defined as:

$$\chi^2(\mathbf{h}_i, \mathbf{h}_j) = \frac{1}{2} \sum_{k=1}^{m} \left(\frac{(\mathbf{h}_i(k) - \mathbf{h}_j(k))^2}{\mathbf{h}_i(k) + \mathbf{h}_j(k)} \right), \tag{3.25}$$

where $\mathbf{h}_i$ and $\mathbf{h}_j$ have to be non-negative. Then χ^2 distance can be interpreted as a weighted difference between elements of the histograms.

The Exponential χ^2 Kernel between feature vectors $\mathbf{h}_i$ and $\mathbf{h}_j$ is defined as:

$$K(\mathbf{h}_i, \mathbf{h}_j) = \exp(-\frac{1}{A}\chi^2(\mathbf{h}_i, \mathbf{h}_j)), \tag{3.26}$$

where A is the scaling parameter, the parameter of the function, that can be determined through the cross-validation (Section 3.2.4.1). Zhang *et al.* [Zhang 2007] have shown that setting this parameter to the average distance between all the training samples reduces the computational cost and gives comparable results.

As we explained in the previous section (Section 3.2.3.2), one of the main challenges of using the Support Vector Machines with the generalized RBF kernels is high complexity when a large number of support vectors is needed. Therefore, in order to speed up the processing time, we applied several modifications to the LibSVM implementation [Chang 2011] of the SVMs. In particular:

- We use the fast implementation of the chi-squared distance between sets of histograms. The code uses compiler intrinsics and OpenMP parallelism [3].
- We set the scaling parameter A value to the mean of the χ^2 distances between all the training samples:

$$A = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \chi^2(\mathbf{h}_i, \mathbf{h}_j), \tag{3.27}$$

  where n is the number of the training samples. This reduces the computational cost and gives comparable results [Zhang 2007, Laptev 2008].
- We use the OpenMP parallelism both for the χ^2 distance calculations and to parallelize the LibSVM.
- LIBSVM stores instances as sparse vectors. In our case most of the data is dense; therefore we use a dense representation what can significantly save the computational time [4].

3.2.3.4 From Binary to Multi-Class Classifcation

Support Vector Machines are binary classifiers. The goal of the binary (binomial) classifier is to decide whether the input data, represented by real vectors, belongs to one or the other class; *i.e.* we want to divide the data points having $y_i = -1$ from those having $y_i = +1$.

In multi-class classification problem we want to assign labels to instances (*i.e.* points), but we have available a finite set of labels, with more than just 2 elements. There are two very popular strategies to use SVMs for multi-class classification: one-vs-one [Zhang 2007] and one-vs-all [Rifkin 2004]. Let's denote the number of classes to recognize as k. The former technique trains a classifier for each possible pair of classes, *i.e.* $\frac{k(k-1)}{2}$ classifiers. For each new test sample, all binary classifiers are evaluated, and the test sample is assigned to the class that is chosen by the majority of classifiers. The latter technique trains k binary classifiers, each to distinguish the samples in a single class from samples in all remaining classes. For each new test sample, all binary classifiers are evaluated, and the test sample is assigned to the class of which classifier outputs the largest (most positive) value (winner-takes-all strategy).

Yuan *et al.* [Yuan 2012] claim that the one-versus-one strategy is not practical for large-scale linear classification due to the huge memory space that is required to store all the $\frac{k(k-1)}{2}$ models of classifiers. Moreover, Rifkin and Klautau [Rifkin 2004] claim that the one-vs-all strategy is as accurate as any other approach, assuming that the underlying binary classifiers are well-tuned regularized classifiers such as the Support Vector Machines. This statement was confirmed by experimental comparisons, *e.g.* by the

[3] https://sites.google.com/site/christophlampert/software

[4] http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/#libsvm_for_dense_data

work of Zhang *et al*. [Zhang 2007], where they present that the one-vs-all strategy and the one-vs-one strategies give almost the same results for classification of texture and object categories.

Therefore, for multi-class classification we apply the one-vs-all strategy, also known as the one-vs-rest strategy.

3.2.4 Action Recognition Approach Assessment

In this section, we explain how we train and assess the action recognition approach. In particular, we clarify how we select the best parameters using the cross-validation (Section 3.2.4.1), how we use dataset splits and how we train a classifier on a dataset (Section 3.2.4.2), and what metric we use to report the results (Section 3.2.4.3).

3.2.4.1 Cross-Validation

Cross-validation (also known as rotation estimation) is a model validation technique for assessing how the model generalizes to independent data. The goal of this technique is to estimate how accurately the model performs in practice, and to limit problems like overfitting. Typically, a dataset is given with a training set and a test set. The cross-validation partitions a training set into complementary subsets, performing the training of a classifier on one subset (called the training subset), and validating the classifier on the other subset (called the validation subset). This is done several times using different partitions to reduce variability, and the final result is calculated as the average of all the results obtained on the validation subsets.

In k-fold cross-validation a training set is randomly partitioned into k equal size subsets, performing the training of a classifier on one subset, and validating the classifier on the remaining $k-1$ subsets. This process is repeated k times and each of the k folds is being used exactly once as the validation data.

Leave-one-person-out cross-validation is a particular case of cross-validation, where videos of one person are used as the validation subset, and the remaining videos as the training subset. This is done repeatedly so that videos of each person are used once as the validation data.

3.2.4.2 Dataset Splits

To properly train a classifier, a dataset should be divided into three independent subset:

- Training subset – it contains examples used for learning.
- Validation subset – it contains examples used to tune the parameters.
- Test subset – it contains examples used to assess the performance of a trained classifier.

We train a classifier on the training subset and we use the validation subset to optimize the parameters of a method and a classifier. Then, we use only the test set to report the results.

If a dataset is provided only with the training and testing sets, we apply the cross-validation (the leave-one-person-out cross-validation, if possible) on the training set to select the best parameters of the classifier.

If a dataset is not divided at all, we apply the rules of the cross-validation (the leave-one-person-out cross-validation, if possible) to split a dataset into the training and testing sets (with k folds we have k training and testing sets), and then we apply the cross-validation (the leave-one-person-out cross-validation, if possible) on the training sets to select the best parameters of the classifier.

3.2.4.3 Mean Class Accuracy Metric

Mean class accuracy (also known as average class accuracy) is the main metric in action recognition domain, which measures the difference between values predicted by a model (*i.e*. a classifier) and the values actually observed from the environment that is being modeled (*i.e*. the ground truth values). This statistical measure describes how correctly a classification test identifies action classes in videos. The mean class accuracy is defined as an average of all class-specific accuracies, *i.e*.:

$$MCA = \frac{1}{|\mathbb{C}|} \sum_{c \in \mathbb{C}} \text{accuracy}(c), \tag{3.28}$$

where $\mathbb{C}$ is the set of action classes, and the function $\text{accuracy}(\cdot)$ measures a class-specific recognition accuracy.

3.3 Datasets

In this section, we present five popular state-of-the-art action recognition datasets and we introduce a new action recognition dataset. The presented datasets are used throughout this thesis work to evaluate the proposed approaches.

We start with relatively simple datasets containing videos of one person at the same time, like the Weizmann dataset (Section 3.3.1) and the KTH dataset (Section 3.3.2). Although these datasets contain a small number of relatively simple actions, they have been widely used in recent years for evaluation. Therefore, they allow us to compare the proposed techniques with many existing state-of-the-art methods.

Then, we perform an evaluation on more realistic datasets. We present the URADL dataset (Section 3.3.3) and the MSR Daily Activity 3D dataset (Section 3.3.4), which contain more challenging videos, with activities of daily living.

Then, in Section 3.3.5, we present a big and very challenging dataset, the HMDB51 dataset, which contains a large set of videos, collected from digitized movies, public databases such as the Prelinger archive, videos available on the Internet, and from YouTube and Google videos.

Finally, in Section 3.3.6, we propose a new, locally collected action recognition dataset, the CHU Nice Hospital dataset.

3.3.1 Weizmann Dataset

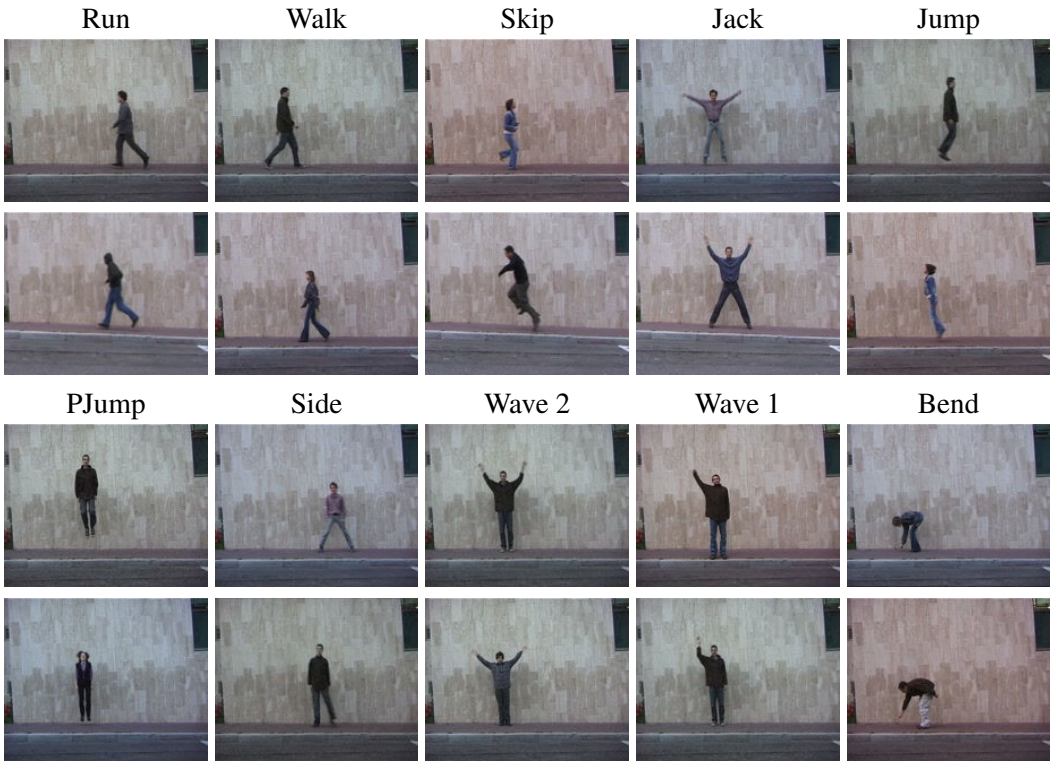


Figure 3.3 – Weizmann dataset: 2 sample video frames for each of the 10 action categories.

The Weizmann Action Recognition dataset (in short, the Weizmann dataset) has been introduced by Blank *et al.* [Blank 2005][5]. It contains videos of 10 types of human actions. The full list of actions is: run, walk, skip, jumping-jack (shortly "jack"), jump-forward-on-two-legs (shortly "jump"), jump-in-place-on-two-legs (shortly "pjump"), gallop-sideways (shortly "side"), wave-two-hands (shortly "wave2"), wave-one-hand (shortly "wave1"), and bend. Each action is performed by 9 people. Videos are recorded with 180×144 pixels spatial resolution and 50 frames per second frame rate. In total, the dataset contains 90 video sequences.

[5] http://www.wisdom.weizmann.ac.il/~vision/SpaceTimeActions.html

Id	Action Name	STIPs				Dense Trajectories			
		min	max	avg	std	min	max	avg	std
1	run	72	302	172	91	1108	1829	1414	211
2	walk	100	512	311	154	1157	2372	1995	426
3	skip	72	424	204	129	986	1922	1499	313
4	jack	87	486	221	140	1125	4596	2272	1156
5	jump	72	395	129	102	407	1200	723	267
6	pjump	72	197	116	41	940	1626	1256	277
7	side	83	176	133	32	963	1785	1461	283
8	wave2	72	72	72	0	283	893	495	203
9	wave1	72	72	72	0	88	630	258	158
10	bend	72	96	76	9	344	654	455	96
$\sum$	All Actions	72	512	153	115	88	4596	1197	768

Table 3.1 – Weizmann dataset: Statistical data (minimum, maximum, average, and standard deviation) about the extracted features (STIPs and Dense Trajectories) for each action category individually and all together. Average and standard deviation values are rounded to the nearest integers.

The main challenges of the Weizmann dataset are: low resolution videos, various people, and cloth variations. Sample video frames from the Weizmann dataset are presented in Figure 3.3. We have also extracted Spatio-Temporal Interest Points (see Section 3.2.1.1) and Dense Trajectories (see Section 3.2.1.2) from this dataset, and we present the statistical data about the extracted features in Table 3.1. On average, the Dense Trajetories extract 7.82 times more features than the Spatio-Temporal Interest Points. To evaluate an approach on this dataset, we use the leave-one-person-out cross-validation evaluation scheme (see Section 3.2.4.1), and we report results using the mean class accuracy metric (see Section 3.2.4.3).

3.3.2 KTH Dataset

Id	Action Name	STIPs				Dense Trajectories			
		min	max	avg	std	min	max	avg	std
1	boxing	202	2087	800	402	1780	35637	7474	6222
2	waving	292	2073	959	347	4837	32368	12336	5191
3	clapping	231	1380	667	239	945	33238	5404	5024
4	walking	382	1590	865	223	7310	22244	11702	2609
5	jogging	386	1972	793	264	5032	17420	8377	2184
6	running	309	1910	662	262	2961	20844	6940	2937
$\sum$	All Actions	202	2087	791	313	945	35637	8711	4965

Table 3.2 – KTH dataset: Statistical data (minimum, maximum, average, and standard deviation) about the extracted features (STIPs and Dense Trajectories) for each action category individually and all together. Average and standard deviation values are rounded to the nearest integers.

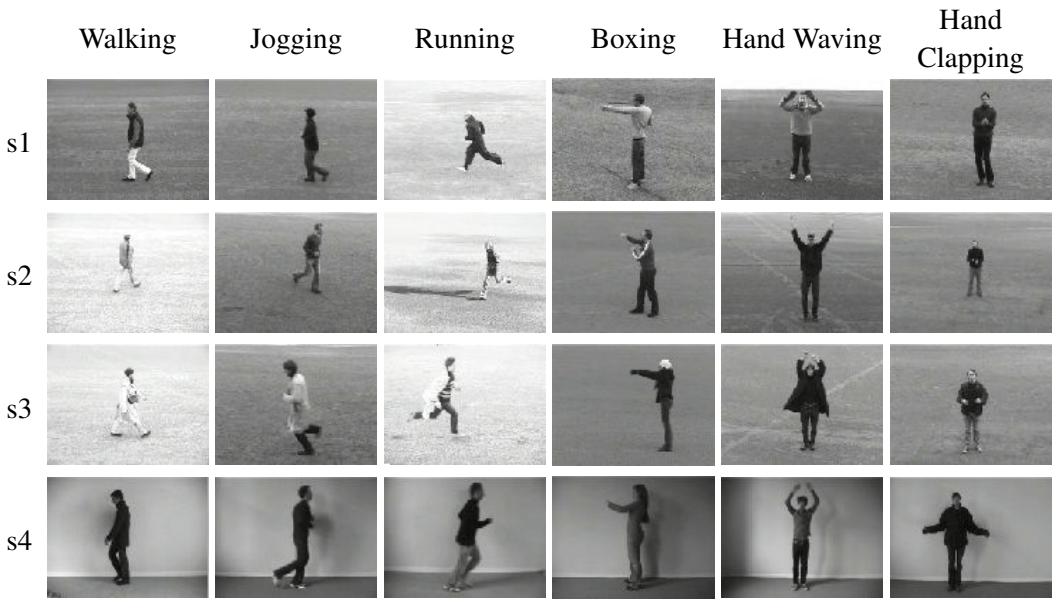


Figure 3.4 – KTH dataset: 4 sample video frames for each of the 6 action categories. Columns represent different types of human actions. Rows represent different scenarios.

The KTH Action dataset (in short, the KTH dataset) has been introduced by Schuldt *et al*. [Schuldt 2004][6]. It contains videos of 6 types of human actions. The full list of actions is: walking, jogging, running, boxing, hand waving, and hand clapping. Each action is performed several times by 25 different subjects. Each subject performs actions in four different scenarios: outdoors (s1), outdoors with scale variation (s2), outdoors with different clothes (s3), and indoors (s4). All videos are recorded over homogeneous backgrounds and are down-sampled by the authors to the spatial resolution of 160×120 pixels. The sequences are recorder using a static camera with 25 frames per second frame rate, and have a length of four seconds on average. In total, the dataset contains 599 video files.

The main challenges of the KTH dataset are: low resolution videos, scale changes, illumination variations, shadows, different people, different scenarios, cloth variations, inter and intra action class speed variations. Sample video frames from the KTH dataset are presented in Figure 3.4. We have also extracted Spatio-Temporal Interest Points (see Section 3.2.1.1) and Dense Trajectories (see Section 3.2.1.2) from this dataset, and we present the statistical data about the extracted features in Table 3.2. On average, the Dense Trajetories extract 11 times more features than the Spatio-Temporal Interest Points.

There are two commonly used experimental setups to evaluate an approach on this dataset: splitting-based evaluation scheme and leave-one-person-out cross-validation evaluation scheme. The former one is the original experimental setup proposed by the authors

[6] http://www.nada.kth.se/cvap/actions/

of this dataset. All sequences are divided with respect to the subjects into a test set (videos of 9 people, labeled as: 2, 3, 5, 6, 7, 8, 9, 10 and 22) and a training set (the remaining video sequences). The latter one, the leave-one-person-out cross-validation evaluation scheme, is the standard experimental setup, described in Section 3.2.4.1. To evaluate an approach on this dataset, we use both evaluation schemes, and we report results using the mean class accuracy metric (see Section 3.2.4.3).

3.3.3 URADL Dataset

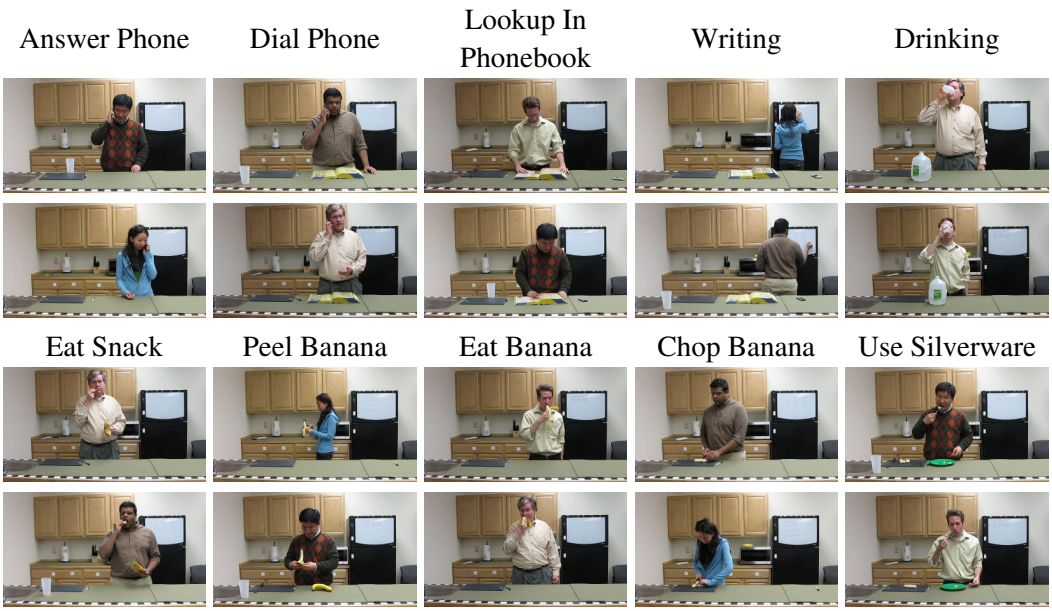


Figure 3.5 – URADL dataset: 2 sample video frames for each of the 10 action categories.

The University of Rochester Activities of Daily Living dataset (in short, the URADL dataset) has been introduced by Messing *et al.* [Messing 2009][7]. It contains videos of 10 types of human activities of daily living, selected to be useful for an assisted cognition task. The full list of activities is: answering a phone, dialling a phone, looking up a phone number in a telephone directory, writing a phone number on a whiteboard, drinking a glass of water, eating snack chips, peeling a banana, eating a banana, chopping a banana, and eating food with silverware. Each action is performed 3 times by 5 different people. Videos are recorded with 1280×720 pixels spatial resolution and are down-sampled to the 640×360 pixels spatial resolution. In total, the dataset contains 150 video sequences with 30 frames per second frame rate.

The main challenges of the URADL dataset are: different shapes, sizes, genders and ethnicity of people. Moreover, the actions were selected to be difficult to separate on the basis of any single source of information. For example, motion information might

[7] http://www.cs.rochester.edu/~rmessing/uradl/

Id	Action Name	STIPs				Dense Trajectories			
		min	max	avg	std	min	max	avg	std
1	answer phone	396	3388	1616	897	4830	22916	13656	6705
2	chop banana	763	5595	3657	1318	12057	35319	24059	5658
3	dial phone	629	2547	1487	550	8821	23843	14526	5052
4	drink water	4274	9040	6919	1777	35932	62422	48517	9096
5	eat banana	345	2854	1921	818	2944	30240	14744	7778
6	eat snack	7028	12390	9250	1569	32633	85690	61811	12888
7	lookup in phonebook	5792	14366	9645	2624	49201	101820	72237	16077
8	peel banana	2219	7430	4893	1594	15338	46023	34426	9645
9	use silverware	4191	8263	6360	1214	29990	55730	45294	7357
10	write on whiteboard	1270	3457	2148	607	10742	36896	18302	6417
$\sum$	All Actions	345	14366	4790	3282	2944	101820	34757	22162

Table 3.3 – URADL dataset: Statistical data (minimum, maximum, average, and standard deviation) about the extracted features (STIPs and Dense Trajectories) for each action category individually and all together. Average and standard deviation values are rounded to the nearest integers.

distinguish eating a banana from peeling a banana or answering a phone from dialling a phone, but motion information might have difficulty distinguishing eating snack chips from eating a banana, while appearance information could be more useful in the second than in the first task.

Sample video frames from the URADL dataset are presented in Figure 4.3. We have also extracted Spatio-Temporal Interest Points (see Section 3.2.1.1) and Dense Trajectories (see Section 3.2.1.2) from this dataset, and we present the statistical data about the extracted features in Table 3.3. On average, the Dense Trajetories extract 7.26 times more features than the Spatio-Temporal Interest Points. To evaluate an approach on this dataset, we use the leave-one-person-out cross-validation evaluation scheme (see Section 3.2.4.1), and we report results using the mean class accuracy metric (see Section 3.2.4.3).

3.3.4 MSR Daily Activity 3D Dataset

The MSR Daily Activity 3D dataset (the MSRDailyActivity3D dataset) has been introduced by Wang *et al*. [Wang 2012][8]. It contains videos of 16 types of human actions. The full list of actions is: drink, eat, read book, call cellphone, write on a paper, use laptop, use vacuum cleaner, cheer up, sit still, toss paper, play game, lie down on sofa, walk, play guitar, stand up, sit down. Each action is performed by 10 people. There is a sofa in the scene, and each subject performs each action twice, once in standing and once in sitting position. The dataset is captured using a Kinect device. Three channels are recorded: depth maps, skeleton joint positions, and RGB video. For consistency with evaluations on other datasets, we only consider the RBG videos. Videos are recorded with 640×360 pixels spatial resolution and 15 frames per second frame rate. In total, the dataset contains

[8] http://research.microsoft.com/en-us/um/people/zliu/ActionRecoRsrc/

Figure 3.6 – MSR Daily Activity 3D dataset: a sample video frame for each of the 16 action categories.

Id	Action Name	STIPs				Dense Trajectories			
		min	max	avg	std	min	max	avg	std
1	drink	91	2670	635	572	3105	26023	8145	5421
2	eat	117	3874	1185	978	4950	28755	12912	6485
3	read book	112	4916	1206	1430	4894	26774	13338	7686
4	call cellphone	134	12431	2300	3662	4638	104292	24528	29945
5	write on a paper	72	2096	412	500	3697	17111	8802	3923
6	use laptop	0	12694	1573	3139	3115	99955	20118	24582
7	use vacuum cleaner	709	8668	3753	1790	14160	77052	39226	13748
8	cheer up	1710	14865	5854	3637	12780	110535	42001	27294
9	sit still	0	2714	270	617	2730	21121	7446	5214
10	toss paper	235	3353	1214	823	3817	30425	11566	7316
11	play game	72	5442	1338	1669	5141	51408	14797	12716
12	lie down on sofa	1023	6926	3304	1686	10582	43193	22086	9731
13	walk	2425	20331	7790	5042	22054	127898	53897	28997
14	play guitar	145	14048	3965	4965	3403	95415	28206	31366
15	stand up	979	6839	2937	1506	9137	41435	19783	8356
16	sit down	809	6231	2788	1585	6924	43550	18615	10641
$\sum$	All Actions	0	20331	2533	3202	2730	127898	21592	21470

Table 3.4 – MSR Daily Activity 3D dataset: Statistical data (minimum, maximum, average, and standard deviation) about the extracted features (STIPs and Dense Trajectories) for each action category individually and all together. Average and standard deviation values are rounded to the nearest integers.

320 video sequences.

There are several challenges of the MSR Daily Activity 3D dataset. Each action is performed in standing and sitting position, which brings an additional intra-class variation. Motion in some actions is occluded, *e.g.* motion of hands in the action use laptop is occluded by the laptop, and thus low number of features is detected. Moreover, in such low movement actions, sometimes most of the features are detected *e.g.* on a head, which does not help in the action recognition. Another challenge is people passing and working in the background, what adds an additional noise to motion features.

Sample video frames from the MSR Daily Activity 3D dataset are presented in Figure 3.6. We have also extracted Spatio-Temporal Interest Points (see Section 3.2.1.1) and Dense Trajectories (see Section 3.2.1.2) from this dataset, and we present the statistical data about the extracted features in Table 3.4. On average, the Dense Trajetories extract 8.53 times more features than the Spatio-Temporal Interest Points. There are 3 video files across 2 actions, where we do not detect any Spatio-Temporal Interest Points (1 video from an action use laptop, and 2 videos from an action sit still). To evaluate an approach on this dataset, we use the leave-one-person-out cross-validation evaluation scheme (see Section 3.2.4.1), and we report results using the mean class accuracy metric (see Section 3.2.4.3).

3.3.5 HMDB51 Dataset

Id	Action Name	STIPs				Dense Trajectories			
		min	max	avg	std	min	max	avg	std
1	brush hair	136	22378	5056	3539	702	119744	40083	26245
2	cartwheel	72	3232	1381	896	604	42929	16738	9843
3	catch	72	1382	308	303	95	25068	4693	4662
4	chew	72	6043	1160	1217	334	37959	10922	9083
5	clap	72	6956	1095	1160	772	43377	9251	7512
6	climb stairs	86	4563	1487	1172	657	39069	16265	11150
7	climb	72	11631	1840	1903	350	104594	23500	20411
8	dive	72	6212	1336	1162	2312	59637	14620	10764
9	draw sword	72	4748	703	739	622	40275	6647	6073
10	dribble	117	7615	1455	1261	800	113169	16979	15961
11	drink	72	5482	1273	1179	1535	107960	16500	16091
12	eat	72	3983	1054	814	121	60716	12432	9761
13	fall floor	72	5407	1401	1040	1379	50745	19436	10695
14	fencing	189	15701	1781	2421	1407	129717	16320	19318
15	flic flac	72	3251	1215	792	4682	36861	17464	8053
16	golf	85	3475	650	474	915	27761	6492	5691
17	handstand	72	2824	1137	699	679	33720	12405	8453
18	hit	72	3012	495	491	875	33278	8137	6192
19	hug	218	6027	1993	1006	2954	52202	18766	7711
20	jump	72	3886	676	688	783	52509	9269	8128
21	kick ball	72	4929	898	769	767	46567	11374	8200
22	kick	72	3926	950	726	679	51637	12620	8278
23	kiss	72	6153	1455	1282	777	63958	20349	12873
24	laugh	72	16044	4566	3874	106	120686	27047	23674
25	pick	72	7169	1864	1374	878	63900	17617	12900
26	pour	72	9495	1058	1512	97	56696	7843	9426
27	pullup	92	3839	1202	844	1039	29247	9381	7074
28	punch	72	8565	1415	1134	1760	65486	15653	9743
29	push	213	5444	2231	1187	913	56739	21480	9810
30	pushup	340	3096	1578	695	2737	31092	12879	6717
31	ride bike	72	8253	2888	1945	654	54668	27937	11420
32	ride horse	72	16155	3959	2913	3104	106801	41243	23896
33	run	72	10280	1441	1471	24	77176	19502	15090
34	shake hands	137	4979	1665	1030	416	41980	14180	8076
35	shoot ball	72	4836	1366	1157	276	32497	10254	7720
36	shoot bow	283	4928	2042	1036	1455	53502	15743	9480
37	shoot gun	94	10263	1172	1436	349	85112	10590	13180
38	sit	72	4774	1301	1058	1275	62461	18604	12108
39	situp	72	2799	873	594	299	30758	7714	6250
40	smile	0	3209	570	593	24	39242	5748	6313

41	smoke	189	5993	1737	1298	2540	88828	16717	15668
42	somersault	0	4265	1016	882	1196	38377	11659	7404
43	stand	0	5398	1337	956	297	62861	16086	11119
44	swing baseball	116	6159	1644	918	540	50145	14372	8972
45	sword exercise	72	10484	1437	1906	663	69782	13028	16463
46	sword	77	13543	1554	1912	1027	122890	16932	17901
47	talk	72	9762	1261	1420	539	81214	14224	13842
48	throw	0	6472	1512	1206	29	70222	13818	10090
49	turn	72	4233	931	772	1747	48883	12797	9642
50	walk	72	11256	1726	1650	943	98040	21694	19011
51	wave	72	6557	1303	1201	806	43275	12579	9971
$\sum$	All Actions	0	22378	151	1675	24	129717	15463	14347

Table 3.5 – HMDB51 dataset: Statistical data (minimum, maximum, average, and standard deviation) about the extracted features (STIPs and Dense Trajectories) for each action category individually and all together. Average and standard deviation values are rounded to the nearest integers.

The HMDB: A Large Human Motion Database dataset (in short, the HMDB51 dataset) has been introduced by Kuehne *et al*. [Kuehne 2011][9]. It contains videos of 51 types of human actions. Actions can be divided into 5 categories:

1. General facial actions: smile, laugh, chew, talk.
2. Facial actions with object manipulation: smoke, eat, drink.
3. General body movements: cartwheel, clap hands, climb, climb stairs, dive, fall on the floor, backhand flip, hand-stand, jump, pull up, push up, run, sit down, sit up, somersault, stand up, turn, walk, wave.
4. Body movements with object interaction: brush hair, catch, draw sword, dribble, golf, hit something, kick ball, pick, pour, push something, ride bike, ride horse, shoot ball, shoot bow, shoot gun, swing baseball bat, sword exercise, throw.
5. Body movements for human interaction: fencing, hug, kick someone, kiss, punch, shake hands, sword fight.

Actions are performed by various numbers of people. Each action category contains a minimum of 101 video clips. Videos are collected from various sources, from: digitized movies, public databases such as the Prelinger archive, videos available on the Internet, and from YouTube and Google videos. In total, the dataset contains 6766 video sequences.

The main challenges of the HMDB51 dataset are: low resolution videos, presence of significant camera and background motion, huge variation of people and actions, multi person actions, videos collected from different sources, and big number of videos and actions.

[9] http://serre-lab.clps.brown.edu/resource/hmdb-a-large-human-motion-database/

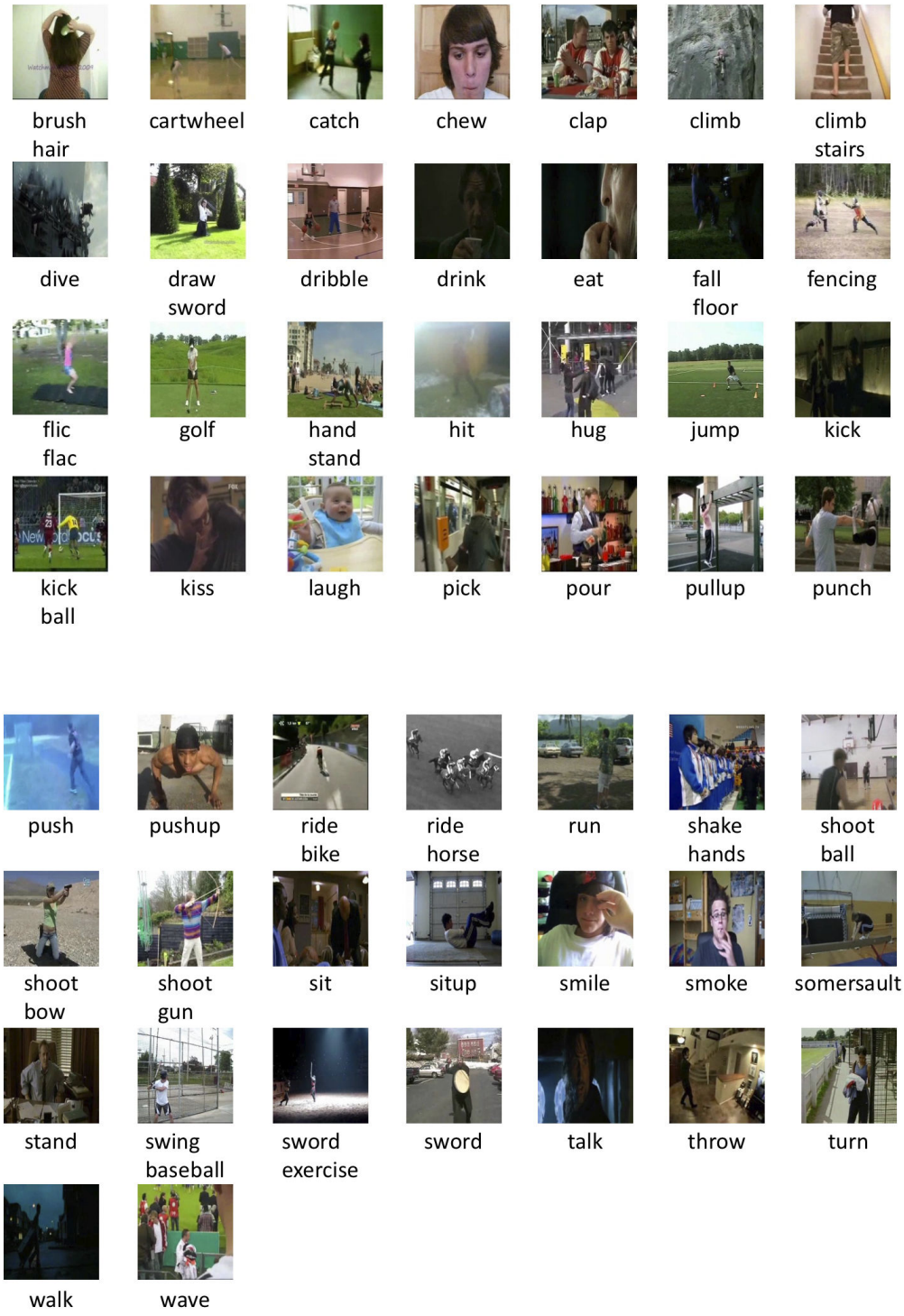


Figure 3.7 – HMDB51 dataset: a sample video frame for each of the 51 action categories.

Sample video frames from the HMDB51 dataset are presented in Figure 3.7. We have also extracted Spatio-Temporal Interest Points (see Section 3.2.1.1) and Dense Trajectories (see Section 3.2.1.2) from this dataset, and we present the statistical data about the extracted features in Table 3.5. On average, the Dense Trajetories extract 10.18 times more features than the Spatio-Temporal Interest Points. There are 5 video files across 4 actions, where we do not detect any Spatio-Temporal Interest Points (2 videos from an action smile, and 1 video from actions: somersault, stand, and throw). To evaluate an approach on this dataset, we use the 3 evaluation splits provided by the authors, we use 5 folds cross-validation for selection of parameters, and we report results using the mean class accuracy metric (see Section 3.2.4.3).

3.3.6 CHU Nice Hospital Dataset

Most of the existing public action recognition datasets could be divided into a few categories:

- low resolution videos of relatively simple actions (such as the Weizmann dataset and the KTH dataset),
- videos of actions of daily living (such as the URADL datset and the MSR Daily Activity 3D dataset), where the camera is set in front of the actor and background does not significantly change between videos,
- video sequences from broadcast television channels, movies, YouTube, and personal cameras (such as the HMDB51 dataset), where a person is often not fully visible, videos are recorded in a significant distance from people, videos are sometimes pixelated, blurred, and contain significant camera motion and background clutter.

Therefore, a new dataset is needed for the recognition of realistic human actions of daily living.

We propose a new dataset, the CHU Nice Hospital (in short, the CHU dataset), which was created with the help of medical scientists. The CHU dataset contains 8 types of real human actions of daily living. The full list of actions is: (a) playing cards, (b) matching ABCD sheets of paper, (c) reading, (d) sitting down and standing up, (e) turning back, (f) standing up and moving ahead, and (g) walking back and forth (2 activities). These actions were selected and annotated by medical doctors.

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

Figure 3.8 – CHU dataset: 3 sample video frames for actions: playing cards (top row), reading (second row), matching ABCD sheets of paper (third row), sitting down and standing up (fourth row), turning back (fifth row), and standing up and moving ahead (bottom row).

Figure 3.9 – CHU dataset: 3 sample video frames for walking1 (top row) and walking2 (bottom row) actions.

Id	Action Name	STIP				Dense Trajectories			
		min	max	avg	std	min	max	avg	std
1	playing cards	373	2741	1132	566	3138	17325	6060	3258
2	matching ABCD ...	4561	11468	6575	2065	21126	61573	31880	10159
3	reading	172	1967	867	544	1635	13651	6138	3380
4	sitting down ...	373	1345	691	244	2640	8089	4371	1471
5	turning back	17	2142	275	398	798	15593	2995	2773
6	standing up ...	930	2648	1725	492	7352	16807	11757	2965
7	walking back	235	2568	824	508	2974	12763	5849	2397
8	walking forth	478	2210	1155	379	3559	12286	7305	2086
$\sum$	All Actions	17	11468	1494	1954	798	61573	8780	9226

Table 3.6 – CHU dataset: Statistical data (minimum, maximum, average, and standard deviation) about the extracted features (STIPs and Dense Trajectories) for each action category individually and all together. Average and standard deviation values are rounded to the nearest integers.

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

Sample video frames from the CHU dataset are presented in Figure 3.8 and in Figure 3.9. More sample video frames are presented in Chapter B. We have also extracted Spatio-Temporal Interest Points (see Section 3.2.1.1) and Dense Trajectories (see Section 3.2.1.2) from this dataset, and we present the statistical data about the extracted features in Table 3.6. On average, the Dense Trajetories extract 5.77 times more features than the Spatio-Temporal Interest Points. To evaluate an approach on this dataset, we use the leave-five-people-out cross-validation evaluation scheme (similar to the leave-one-person-out cross-validation, see Section 3.2.4.1), and we report results using the mean class accuracy metric (see Section 3.2.4.3).

3.3.7 Datasets Summary

The presented datasets provide various scenarios and challenges. Some actions are very short and can last just a second like an action smile from the HMDB51 dataset; other actions can take up to 50 seconds like an action eat snack from the URADL dataset. For very short videos and actions the Spatio-Temporal Interest Points sometimes do not detect any features; there are 8 videos across 6 actions with 0 detected STIP features (the MSR Daily Activity 3D dataset: 1 video from an action use laptop, and 2 videos from an action sit still; the HMDB51 dataset: 2 videos from an action smile, and 1 video from actions: somersault, stand, and throw). The Dense Trajectories always detect features; in fact, they extract on average 8.43 times more features than the STIP, which can lead to the extraction of large amounts of features, *e.g.* up to 129717 features for action fencing from the HMDB51 dataset, whereas the STIP detects up to 15701 features for the same action.

3.4 Experiments, Comparison, and Analysis

In this section, we present an evaluation, comparison, and analysis of our action recognition framework. Firstly, we extract local spatio-temporal features. Then, we apply a local feature encoding technique for each descriptor. Finally, we fuse video representations and we apply a classifier.

We perform experiments using:

- **2 local spatio-temporal features:**
  - Spatio-Temporal Interest Points (Section 3.2.1.1),
  - Dense Trajectories (Section 3.2.1.2).
- **3 local features encoding techniques:**
  - Bag-of-features with L1 norm (Section 3.2.2.1),
  - Bag-of-features with L2 norm (Section 3.2.2.1),
  - Fisher vector encoding (Section 3.2.2.2).
- Local feature encoding techniques create **various codebook sizes:**
  - 10 various codebook sizes for the bag-of-features: $\{16, 32, 64, 128, 256, 512\}$ and $\{1000, 2000, 3000, 4000\}$,
  - 6 various codebook sizes for the Fisher vector encoding: $\{16, 32, 64, 128, 256, 512\}$.
- **2 classifiers:**
  - Support Vector Machines with the exponential χ^2 kernel for the bag-of-features approach,
  - Support Vector Machines with the linear kernel for the Fisher vector encoding.
- **4 state-of-the-art action recognition datasets:**
  - two bigger and more challenging datasets:
    - HMDB51 (Section 3.3.5),
    - MSRDailyActivity3D (Section 3.3.4).

    These datasets contain a greater number of actions and videos, *e.g.* 51 actions and 6766 videos for the HMDB51 dataset.
  - two smaller and less challenging datasets:
    - URADL (see Section 3.3.3),
    - Weizmann (see Section 3.3.1).

    These datasets contain a smaller number of actions and videos, *e.g.* 10 actions and 150 videos for the URADL dataset.

  The selected datasets vary in the number of actions, types of actions, number of videos, and challenges.

3.4.1 Spatio-Temporal Interest Points

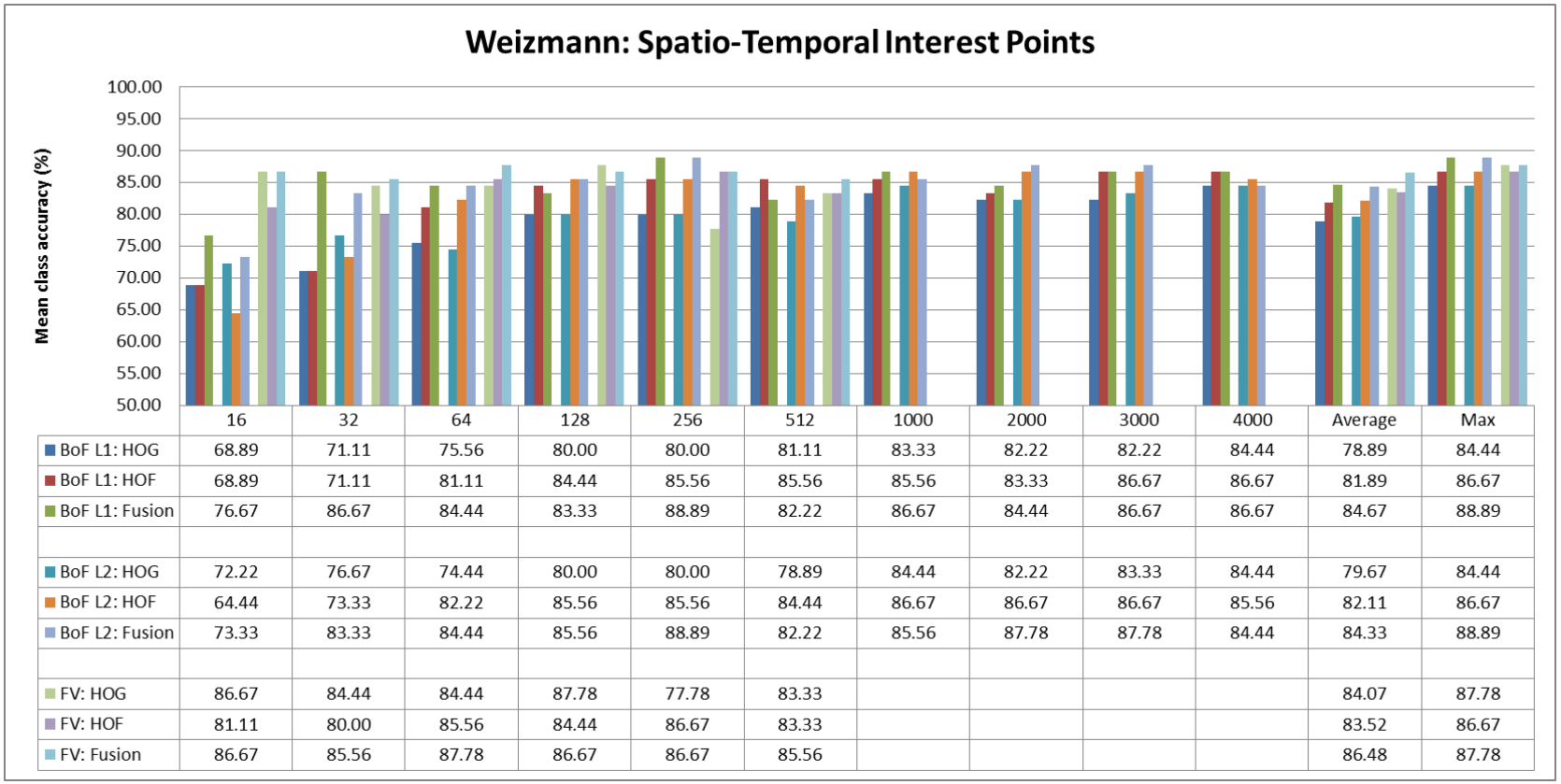
			HMDB51	MSR	URADL	Weizmann	Average
BoF L1	HOG	Mean	14.49%	41.41%	81.20%	78.89%	54.00%
		Max	17.52%	44.38%	89.33%	84.44%	58.92%
	HOF	Mean	16.42%	49.97%	82.67%	81.89%	57.74%
		Max	19.70%	54.06%	90.00%	86.67%	62.61%
	Fusion	Mean	22.49%	54.69%	89.67%	84.67%	62.88%
		Max	27.47%	60.00%	94.00%	88.89%	67.59%
BoF L2	HOG	Mean	14.81%	44.25%	82.33%	79.67%	55.27%
		Max	18.34%	52.19%	91.33%	84.44%	61.58%
	HOF	Mean	17.08%	52.41%	83.67%	82.11%	58.82%
		Max	20.72%	59.38%	91.33%	86.67%	64.52%
	Fusion	Mean	23.52%	57.72%	89.60%	84.33%	63.79%
		Max	27.91%	64.69%	93.33%	88.89%	68.70%
FV	HOG	Mean	14.46%	56.77%	82.22%	84.07%	59.38%
		Max	18.78%	58.75%	86.00%	87.78%	61.52%
	HOF	Mean	19.56%	65.26%	87.56%	83.52%	63.83%
		Max	23.83%	66.56%	90.00%	86.67%	66.36%
	Fusion	Mean	26.78%	70.57%	91.22%	86.48%	68.81%
		Max	31.53%	72.50%	94.00%	87.78%	70.44%

Table 3.7 – Evaluation results of the Spatio-Temporal Interest Points with 3 local feature encoding techniques (the Bag-of-Features (BoF) with L1 and L2 norms, and the Fisher vectors (FV)). The table presents the mean class accuracy of the Histogram of Oriented Gradients (HOG), Histogram of Optical Flow (HOF), and of the fusion of the HOG and the HOF descriptors, for HMDB51, MSR Daily Activity 3D (MSR), URADL, and Weizmann datasets, along with the average results over all these dataset. For each descriptor, fusion, and dataset we present the average and the maximum results over all the evaluated codebook sizes.

We present the general overview of the experimental results of the Spatio-Temporal Interest Points in Table 3.7. We also present the detail experimental results and the analysis of the results as follows.

3.4.1.1 Weizmann Dataset

- The detail evaluation results are presented in Figure 3.10.
- The bag-of-features with the L1 norm achieve the same results as the bag-of-features with the L2 norm.
- The HOF descriptor works better than the HOG descriptor for the bag-of-features approach, and slightly worse for the Fisher vectors.



	16	32	64	128	256	512	1000	2000	3000	4000	Average	Max
BoF L1: HOG	68.89	71.11	75.56	80.00	80.00	81.11	83.33	82.22	82.22	84.44	78.89	84.44
BoF L1: HOF	68.89	71.11	81.11	84.44	85.56	85.56	85.56	83.33	86.67	86.67	81.89	86.67
BoF L1: Fusion	76.67	86.67	84.44	83.33	88.89	82.22	86.67	84.44	86.67	86.67	84.67	88.89
BoF L2: HOG	72.22	76.67	74.44	80.00	80.00	78.89	84.44	82.22	83.33	84.44	79.67	84.44
BoF L2: HOF	64.44	73.33	82.22	85.56	85.56	84.44	86.67	86.67	86.67	85.56	82.11	86.67
BoF L2: Fusion	73.33	83.33	84.44	85.56	88.89	82.22	85.56	87.78	87.78	84.44	84.33	88.89
FV: HOG	86.67	84.44	84.44	87.78	77.78	83.33					84.07	87.78
FV: HOF	81.11	80.00	85.56	84.44	86.67	83.33					83.52	86.67
FV: Fusion	86.67	85.56	87.78	86.67	86.67	85.56					86.48	87.78

Figure 3.10 – Weizmann dataset: Evaluation results of the Spatio-Temporal Interest Points with 3 local feature encoding techniques (the Bag-of-Features (BoF) with L1 and L2 norms, and the Fisher vectors (FV)). The plot presents the mean class accuracy of the Histogram of Oriented Gradients (HOG), Histogram of Optical Flow (HOF), and the fusion of HOG and HOF descriptors, with respect to the codebook size (the "x" axis).

- The fusion of both descriptors improves the results for the bag-of-features.
- The best result, 88.89% of mean class accuracy, is achieved by the fusion of descriptors, bag-of-features, and the codebook size 256. Similar result, 87.78% of mean class accuracy, is achieved by the the fusion of descriptors, Fisher vectors, and the codebook size 64.

3.4.1.2 URADL Dataset

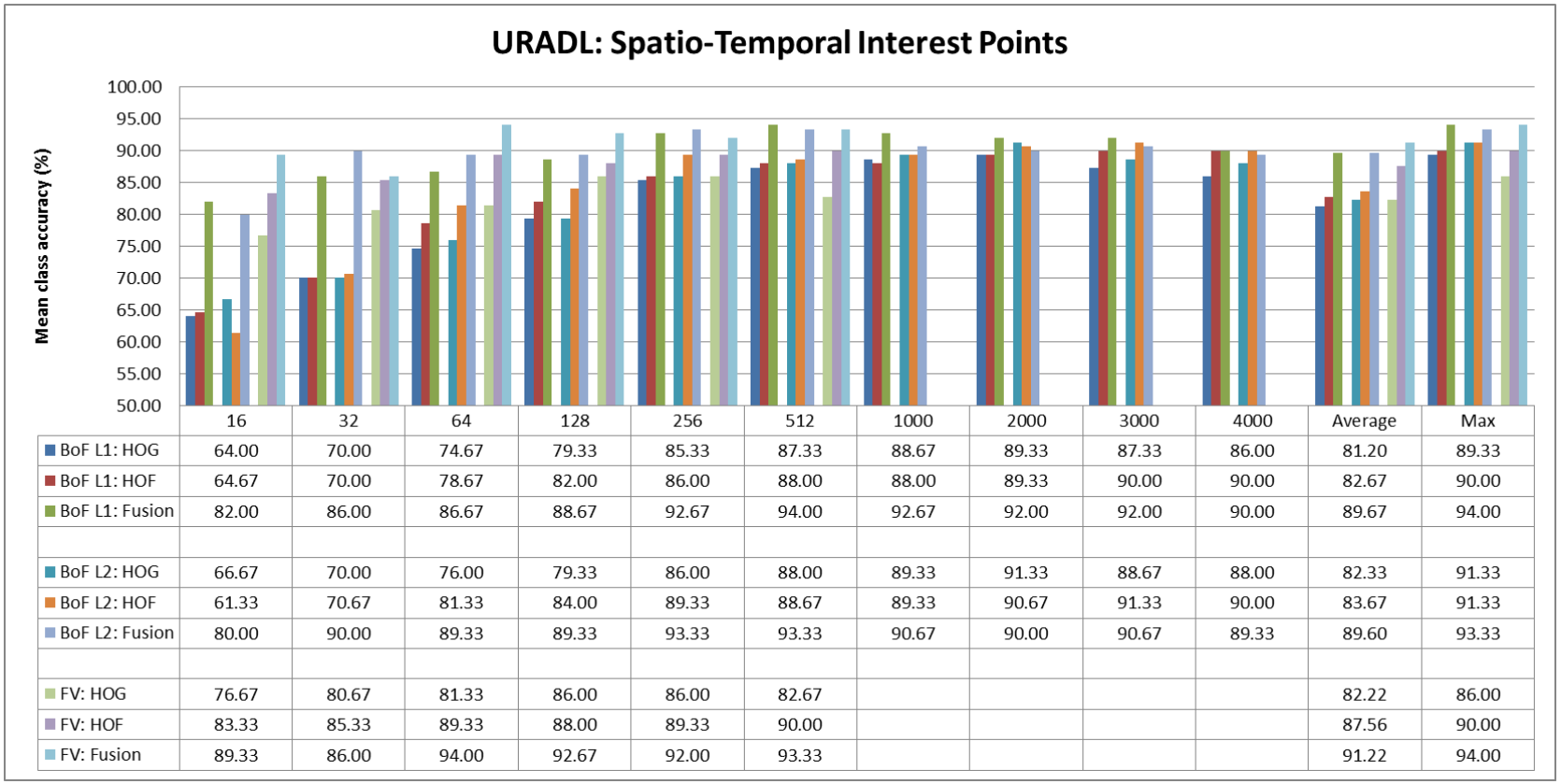
- The detail evaluation results are presented in Figure 3.11.
- For every evaluated local feature encoding technique: the HOF descriptor works better than the HOG descriptor, and the fusion of both descriptors always improves the results.
- The bag-of-features with L2 norm work the best for the HOG and the HOF descriptors alone, and the bag-of-features with L1 norm work the best for the fusion of both descriptors.
- The best result, 94% of mean class accuracy, is achieved by the fusion of descriptors and: (a) Fisher vectors and the codebook size 64, and (b) bag-of-features and the codebook size 512.

3.4.1.3 MSR Daily Activity 3D Dataset

- The detail evaluation results are presented in Figure 3.12.
- For every evaluated local feature encoding technique: the HOF descriptor works better than the HOG descriptor, and the fusion of both descriptors always improves the results.
- For every descriptor and their fusions: the bag-of-features with L2 norm always work much better than the bag-of-features with L1 norm, and the Fisher vector encoding always works much better than the bag-of-features approach.
- The best result, 72.50% of mean class accuracy, is achieved by the fusion of descriptors, Fisher vectors, and the codebook size 32.

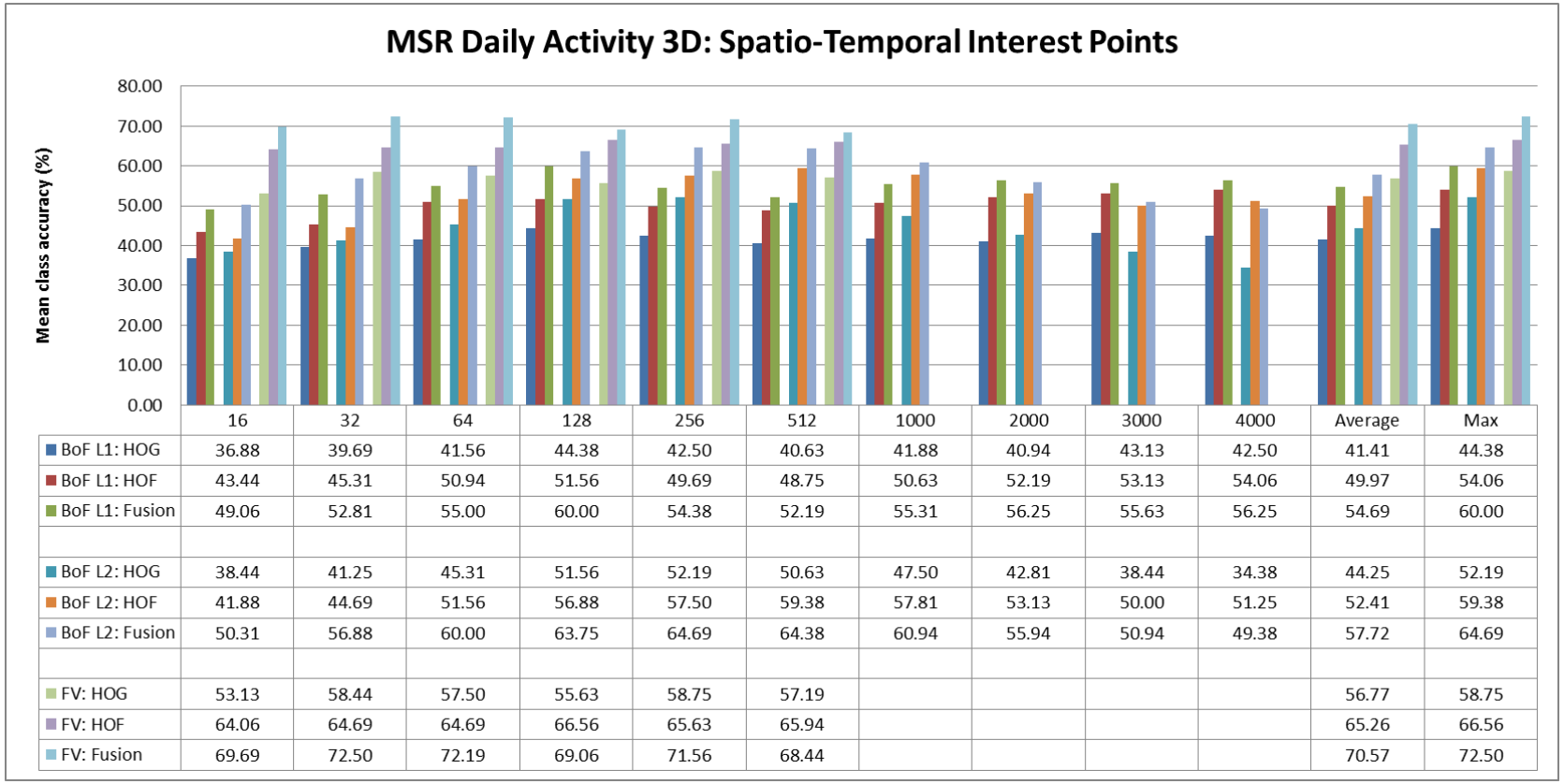
3.4.1.4 HMDB51 Dataset

- The detail evaluation results are presented in Figure 3.13.
- For every evaluated local feature encoding technique: the HOF descriptor works better than the HOG descriptor, and the fusion of both descriptors always improves the results.
- For every descriptor and their fusions: the bag-of-features with L2 norm always work better than the bag-of-features with L1 norm, and the Fisher vector encoding always works better than the bag-of-features approach.



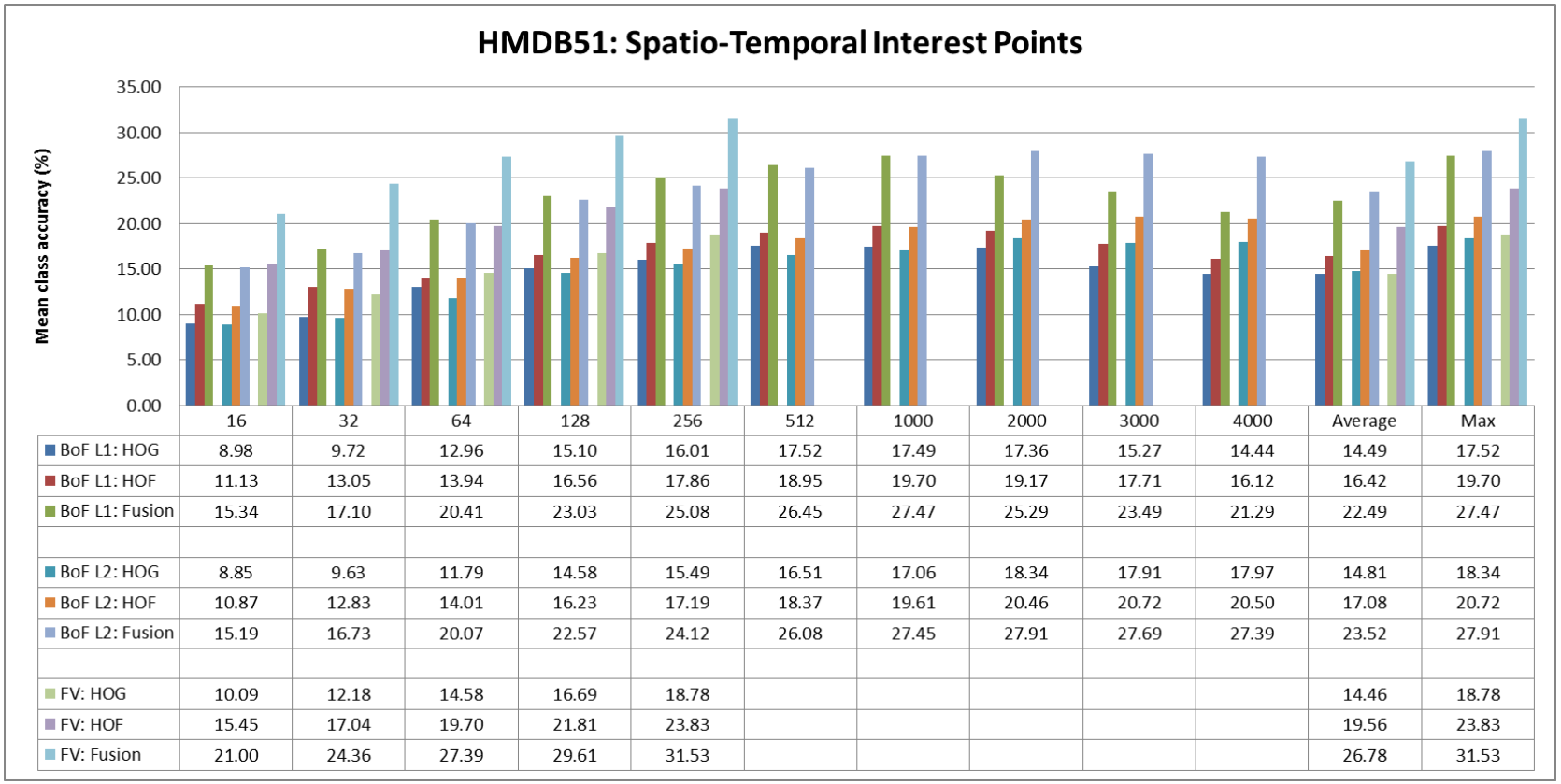
	16	32	64	128	256	512	1000	2000	3000	4000	Average	Max
BoF L1: HOG	64.00	70.00	74.67	79.33	85.33	87.33	88.67	89.33	87.33	86.00	81.20	89.33
BoF L1: HOF	64.67	70.00	78.67	82.00	86.00	88.00	88.00	89.33	90.00	90.00	82.67	90.00
BoF L1: Fusion	82.00	86.00	86.67	88.67	92.67	94.00	92.67	92.00	92.00	90.00	89.67	94.00
BoF L2: HOG	66.67	70.00	76.00	79.33	86.00	88.00	89.33	91.33	88.67	88.00	82.33	91.33
BoF L2: HOF	61.33	70.67	81.33	84.00	89.33	88.67	89.33	90.67	91.33	90.00	83.67	91.33
BoF L2: Fusion	80.00	90.00	89.33	89.33	93.33	93.33	90.67	90.00	90.67	89.33	89.60	93.33
FV: HOG	76.67	80.67	81.33	86.00	86.00	82.67					82.22	86.00
FV: HOF	83.33	85.33	89.33	88.00	89.33	90.00					87.56	90.00
FV: Fusion	89.33	86.00	94.00	92.67	92.00	93.33					91.22	94.00

Figure 3.11 – URADL dataset: Evaluation results of the Spatio-Temporal Interest Points with 3 local feature encoding techniques (the Bag-of-Features (BoF) with L1 and L2 norms, and the Fisher vectors (FV)). The plot presents the mean class accuracy of the Histogram of Oriented Gradients (HOG), Histogram of Optical Flow (HOF), and the fusion of HOG and HOF descriptors, with respect to the codebook size (the "x" axis).



	16	32	64	128	256	512	1000	2000	3000	4000	Average	Max
BoF L1: HOG	36.88	39.69	41.56	44.38	42.50	40.63	41.88	40.94	43.13	42.50	41.41	44.38
BoF L1: HOF	43.44	45.31	50.94	51.56	49.69	48.75	50.63	52.19	53.13	54.06	49.97	54.06
BoF L1: Fusion	49.06	52.81	55.00	60.00	54.38	52.19	55.31	56.25	55.63	56.25	54.69	60.00
BoF L2: HOG	38.44	41.25	45.31	51.56	52.19	50.63	47.50	42.81	38.44	34.38	44.25	52.19
BoF L2: HOF	41.88	44.69	51.56	56.88	57.50	59.38	57.81	53.13	50.00	51.25	52.41	59.38
BoF L2: Fusion	50.31	56.88	60.00	63.75	64.69	64.38	60.94	55.94	50.94	49.38	57.72	64.69
FV: HOG	53.13	58.44	57.50	55.63	58.75	57.19					56.77	58.75
FV: HOF	64.06	64.69	64.69	66.56	65.63	65.94					65.26	66.56
FV: Fusion	69.69	72.50	72.19	69.06	71.56	68.44					70.57	72.50

Figure 3.12 – MSR Daily Activity 3D dataset: Evaluation results of the Spatio-Temporal Interest Points with 3 local feature encoding techniques (the Bag-of-Features (BoF) with L1 and L2 norms, and the Fisher vectors (FV)). The plot presents the mean class accuracy of the Histogram of Oriented Gradients (HOG), Histogram of Optical Flow (HOF), and the fusion of HOG and HOF descriptors, with respect to the codebook size (the "x" axis).



	16	32	64	128	256	512	1000	2000	3000	4000	Average	Max
BoF L1: HOG	8.98	9.72	12.96	15.10	16.01	17.52	17.49	17.36	15.27	14.44	14.49	17.52
BoF L1: HOF	11.13	13.05	13.94	16.56	17.86	18.95	19.70	19.17	17.71	16.12	16.42	19.70
BoF L1: Fusion	15.34	17.10	20.41	23.03	25.08	26.45	27.47	25.29	23.49	21.29	22.49	27.47
BoF L2: HOG	8.85	9.63	11.79	14.58	15.49	16.51	17.06	18.34	17.91	17.97	14.81	18.34
BoF L2: HOF	10.87	12.83	14.01	16.23	17.19	18.37	19.61	20.46	20.72	20.50	17.08	20.72
BoF L2: Fusion	15.19	16.73	20.07	22.57	24.12	26.08	27.45	27.91	27.69	27.39	23.52	27.91
FV: HOG	10.09	12.18	14.58	16.69	18.78						14.46	18.78
FV: HOF	15.45	17.04	19.70	21.81	23.83						19.56	23.83
FV: Fusion	21.00	24.36	27.39	29.61	31.53						26.78	31.53

Figure 3.13 – HMDB51 dataset: Evaluation results of the Spatio-Temporal Interest Points with 3 local feature encoding techniques (the Bag-of-Features (BoF) with L1 and L2 norms, and the Fisher vectors (FV)). The plot presents the mean class accuracy of the Histogram of Oriented Gradients (HOG), Histogram of Optical Flow (HOF), and the fusion of HOG and HOF descriptors, with respect to the codebook size (the "x" axis).

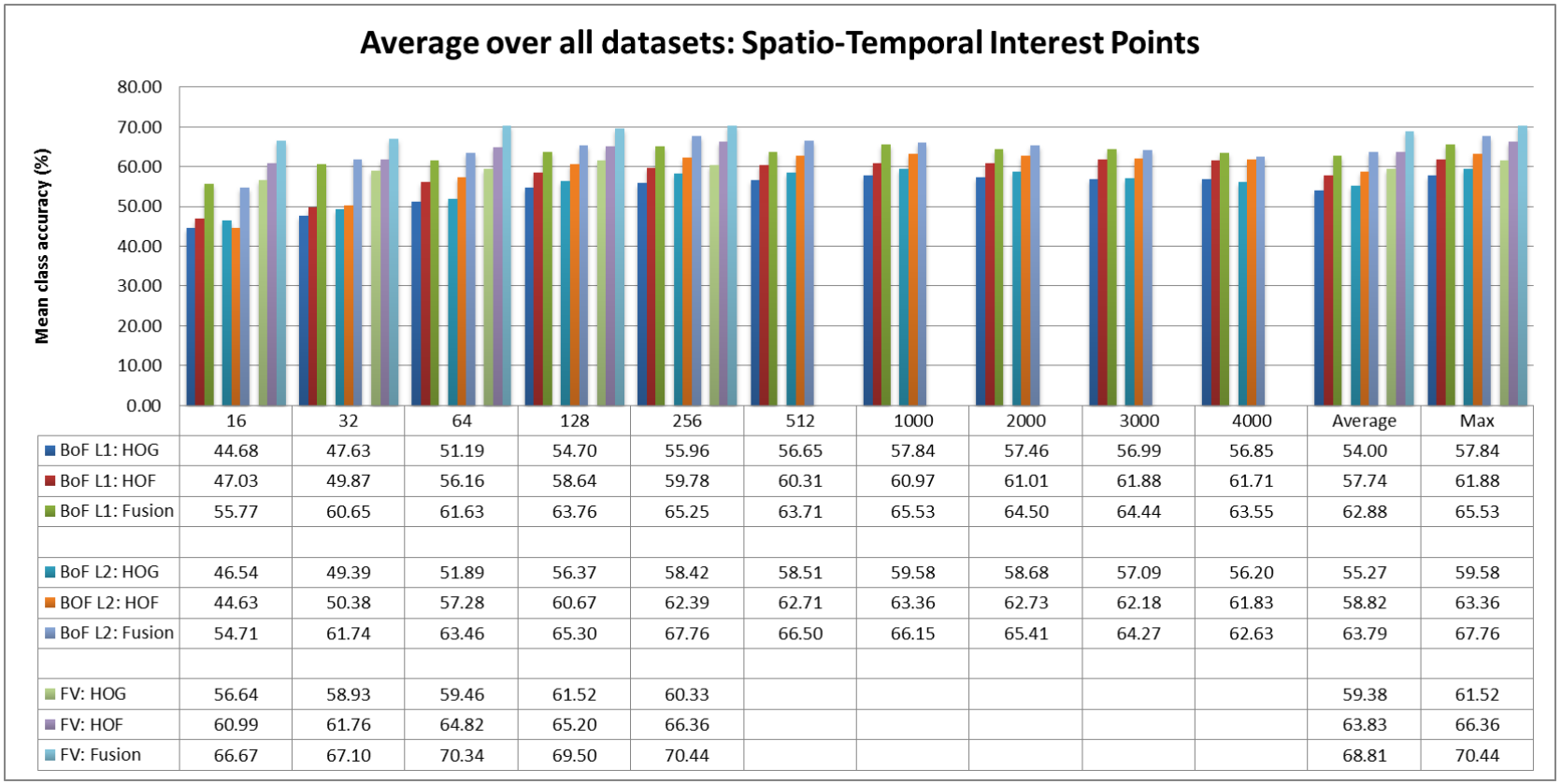
- The best result, 31.53% of mean class accuracy, is achieved by the fusion of descriptors, Fisher vectors, and the codebook size 256.

3.4.1.5 Overall

- The detail evaluation results are presented in Figure 3.14.
- For every evaluated local feature encoding technique: the HOF descriptor works better than the HOG descriptor, and the fusion of both descriptors always improves the results.
- For every descriptor and their fusions: the bag-of-features with the L2 norm work better than the bag-of-features with the L1 norm, and the Fisher vector encoding works better than the bag-of-features approach.

3.4.1.6 Results Summary and Analysis

- The HOF descriptor works better than the HOG descriptor.
- The motion information alone is not good enough to achieve good results. The fusion of motion and appearance information improves action recognition accuracy.
- Fisher vector encoding works better than the bag-of-features approach. Only for the Weizmann dataset the bag-of-features approach achieves better results, but the difference in the accuracy is small (~1.1%). For the HMDB51 dataset the Fisher vector encoding outperforms the bag-of-features by ~3.62% and for the MSR Daily Activity 3D dataset by ~7.8%.
- The bag-of-features with the L2 norm work better than the bag-of-features with the L1 norm (*e.g.* 64.69% vs. 60% for the MSR Daily Activity 3D dataset). Only for the URADL dataset the bag-of-features with L1 norm work better than the bag-of-features with L2 norm; however, the difference in the accuracy is small (94% vs. 93.33%).
- In general, there is no codebook size which always performs the best for the bag-of-features and the Fisher vectors.



	16	32	64	128	256	512	1000	2000	3000	4000	Average	Max
BoF L1: HOG	44.68	47.63	51.19	54.70	55.96	56.65	57.84	57.46	56.99	56.85	54.00	57.84
BoF L1: HOF	47.03	49.87	56.16	58.64	59.78	60.31	60.97	61.01	61.88	61.71	57.74	61.88
BoF L1: Fusion	55.77	60.65	61.63	63.76	65.25	63.71	65.53	64.50	64.44	63.55	62.88	65.53
BoF L2: HOG	46.54	49.39	51.89	56.37	58.42	58.51	59.58	58.68	57.09	56.20	55.27	59.58
BOF L2: HOF	44.63	50.38	57.28	60.67	62.39	62.71	63.36	62.73	62.18	61.83	58.82	63.36
BoF L2: Fusion	54.71	61.74	63.46	65.30	67.76	66.50	66.15	65.41	64.27	62.63	63.79	67.76
FV: HOG	56.64	58.93	59.46	61.52	60.33						59.38	61.52
FV: HOF	60.99	61.76	64.82	65.20	66.36						63.83	66.36
FV: Fusion	66.67	67.10	70.34	69.50	70.44						68.81	70.44

Figure 3.14 – Average over all dataset: Evaluation results of the Spatio-Temporal Interest Points with 3 local feature encoding techniques (the Bag-of-Features (BoF) with L1 and L2 norms, and the Fisher vectors (FV)). The plot presents the mean class accuracy of the Histogram of Oriented Gradients (HOG), Histogram of Optical Flow (HOF), and the fusion of HOG and HOF descriptors, with respect to the codebook size (the "x" axis).

3.4.2 Dense Trajectories

		HMDB51	MSR	URADL	Weizmann	Average
TRAJ	Mean	20.26%	64.53%	75.11%	91.11%	61.47%
	Max	22.29%	68.13%	80.67%	94.44%	64.63%
HOG	Mean	25.07%	57.03%	81.44%	89.81%	63.13%
	Max	27.36%	60.94%	83.33%	92.22%	64.63%
HOF	Mean	28.88%	66.15%	85.67%	85.56%	65.50%
	Max	31.53%	68.75%	92.67%	86.67%	66.90%
MBH-x	Mean	29.72%	63.59%	85.89%	92.22%	67.48%
	Max	32.24%	66.56%	87.33%	93.33%	69.70%
MBH-y	Mean	35.23%	65.83%	87.11%	94.63%	70.19%
	Max	38.58%	67.81%	89.33%	96.67%	72.34%
Fusion	Mean	44.67%	74.06%	92.56%	92.59%	75.93%
	Max	46.97%	76.25%	94.00%	93.33%	76.86%

Table 3.8 – The evaluation results of the Dense Trajectories with the Fisher vectors. The table presents the mean class accuracy of the Trajectory shape descriptor (TRAJ), Histogram of Oriented Gradients (HOG), Histogram of Optical Flow (HOF), Motion Boundary Histogram (MBH) for "x" and "y" components, and fusion of all these descriptors, for HMDB51, MSR Daily Activity 3D (MSR), URADL, and Weizmann datasets, along with the average results over all these dataset. For each descriptor, fusion, and dataset we present the average and the maximum results over all the evaluated codebook sizes.

One of the conclusions from the previous section is that the Fisher vector encoding works better than the bag-of-features approach. Therefore, most of our experiments with the Dense Trajectories are based on the Fisher vectors.

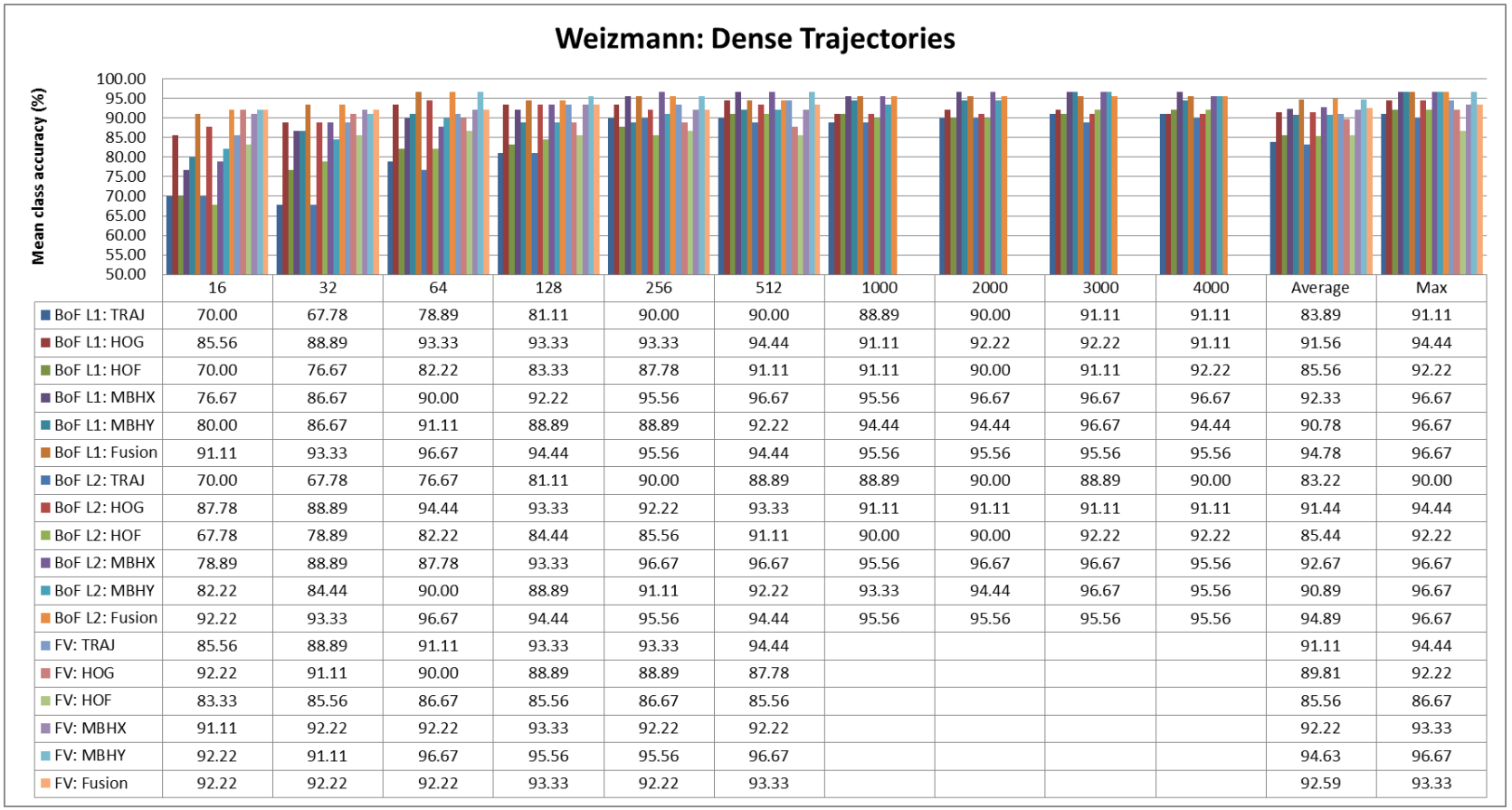
We present the general overview of the experimental results of the Dense Trajectories in Table 3.8 and in Table 3.9. We also present the detail experimental results and the analysis of the results as follows.

3.4.2.1 Weizmann Dataset

- The detail evaluation results are presented in Figure 3.15.
- Like with the Spatio-Temporal Interest Points (Section 3.4.1), the bag-of-features with the L1 norm achieves the same results as the bag-of-features with the L2 norm.
- For every evaluated local feature encoding technique: the HOG descriptor works better than the HOF descriptor, the MBH descriptor works better than the HOG descriptor.
- The fusion of all the descriptors improves the results over the individual descriptors for the bag-of-features, and decreases the results for the Fisher vectors.

			BoF L1	BoF L2	FV
URADL	TRAJ	Mean	71.40%	70.00%	75.11%
		Max	76.67%	74.00%	80.67%
	HOG	Mean	71.13%	71.13%	81.44%
		Max	87.33%	84.00%	83.33%
	HOF	Mean	77.87%	77.07%	85.67%
		Max	86.00%	84.67%	92.67%
	MBH-x	Mean	69.87%	70.47%	85.89%
		Max	79.33%	78.67%	87.33%
	MBH-y	Mean	81.33%	80.73%	87.11%
		Max	87.33%	86.67%	89.33%
	Fusion	Mean	85.27%	86.33%	92.56%
		Max	91.33%	92.00%	94.00%
Weizmann	TRAJ	Mean	83.89%	83.22%	91.11%
		Max	91.11%	90.00%	94.44%
	HOG	Mean	91.56%	91.44%	89.81%
		Max	94.44%	94.44%	92.22%
	HOF	Mean	85.56%	85.44%	85.56%
		Max	92.22%	92.22%	86.67%
	MBH-x	Mean	92.33%	92.67%	92.22%
		Max	96.67%	96.67%	93.33%
	MBH-y	Mean	90.78%	90.89%	94.63%
		Max	96.67%	96.67%	96.67%
	Fusion	Mean	94.78%	94.89%	92.59%
		Max	96.67%	96.67%	93.33%

Table 3.9 – The evaluation results of Dense Trajectories with 3 local feature encoding techniques (the Bag-of-Features (BoF) with L1 and L2 norms, and the Fisher vectors (FV)). The table presents the mean class accuracy of the Trajectory shape descriptor (TRAJ), Histogram of Oriented Gradients (HOG), Histogram of Optical Flow (HOF), Motion Boundary Histogram (MBH) for "x" and "y" components, and fusion of all these descriptors, for URADL and Weizmann datasets. For each descriptor, fusion, and dataset we present the average and the maximum results over all the evaluated codebook sizes.



	16	32	64	128	256	512	1000	2000	3000	4000	Average	Max
BoF L1: TRAJ	70.00	67.78	78.89	81.11	90.00	90.00	88.89	90.00	91.11	91.11	83.89	91.11
BoF L1: HOG	85.56	88.89	93.33	93.33	93.33	94.44	91.11	92.22	92.22	91.11	91.56	94.44
BoF L1: HOF	70.00	76.67	82.22	83.33	87.78	91.11	91.11	90.00	91.11	92.22	85.56	92.22
BoF L1: MBHX	76.67	86.67	90.00	92.22	95.56	96.67	95.56	96.67	96.67	96.67	92.33	96.67
BoF L1: MBHY	80.00	86.67	91.11	88.89	88.89	92.22	94.44	94.44	96.67	94.44	90.78	96.67
BoF L1: Fusion	91.11	93.33	96.67	94.44	95.56	94.44	95.56	95.56	95.56	95.56	94.78	96.67
BoF L2: TRAJ	70.00	67.78	76.67	81.11	90.00	88.89	88.89	90.00	88.89	90.00	83.22	90.00
BoF L2: HOG	87.78	88.89	94.44	93.33	92.22	93.33	91.11	91.11	91.11	91.11	91.44	94.44
BoF L2: HOF	67.78	78.89	82.22	84.44	85.56	91.11	90.00	90.00	92.22	92.22	85.44	92.22
BoF L2: MBHX	78.89	88.89	87.78	93.33	96.67	96.67	95.56	96.67	96.67	95.56	92.67	96.67
BoF L2: MBHY	82.22	84.44	90.00	88.89	91.11	92.22	93.33	94.44	96.67	95.56	90.89	96.67
BoF L2: Fusion	92.22	93.33	96.67	94.44	95.56	94.44	95.56	95.56	95.56	95.56	94.89	96.67
FV: TRAJ	85.56	88.89	91.11	93.33	93.33	94.44					91.11	94.44
FV: HOG	92.22	91.11	90.00	88.89	88.89	87.78					89.81	92.22
FV: HOF	83.33	85.56	86.67	85.56	86.67	85.56					85.56	86.67
FV: MBHX	91.11	92.22	92.22	93.33	92.22	92.22					92.22	93.33
FV: MBHY	92.22	91.11	96.67	95.56	95.56	96.67					94.63	96.67
FV: Fusion	92.22	92.22	92.22	93.33	92.22	93.33					92.59	93.33

Figure 3.15 – Weizmann dataset: Evaluation results of the Dense Trajectories with 3 local feature encoding techniques (the Bag-of-Features (BoF) with L1 and L2 norms, and the Fisher vectors (FV)). The plot presents the mean class accuracy of the Trajectory shape descriptor (TRAJ), Histogram of Oriented Gradients (HOG), Histogram of Optical Flow (HOF), Motion Boundary Histogram (MBH) for "x" and "y" components, and fusion of all these descriptors, with respect to the codebook size (the "x" axis).

- The MBH descriptor works particularly well.
- The MBH-y component works better than the MBH-x component.
- The best result, 96.67% of mean class accuracy, is achieved several times, by both the bag-of-features and the Fisher vectors.

3.4.2.2 URADL Dataset

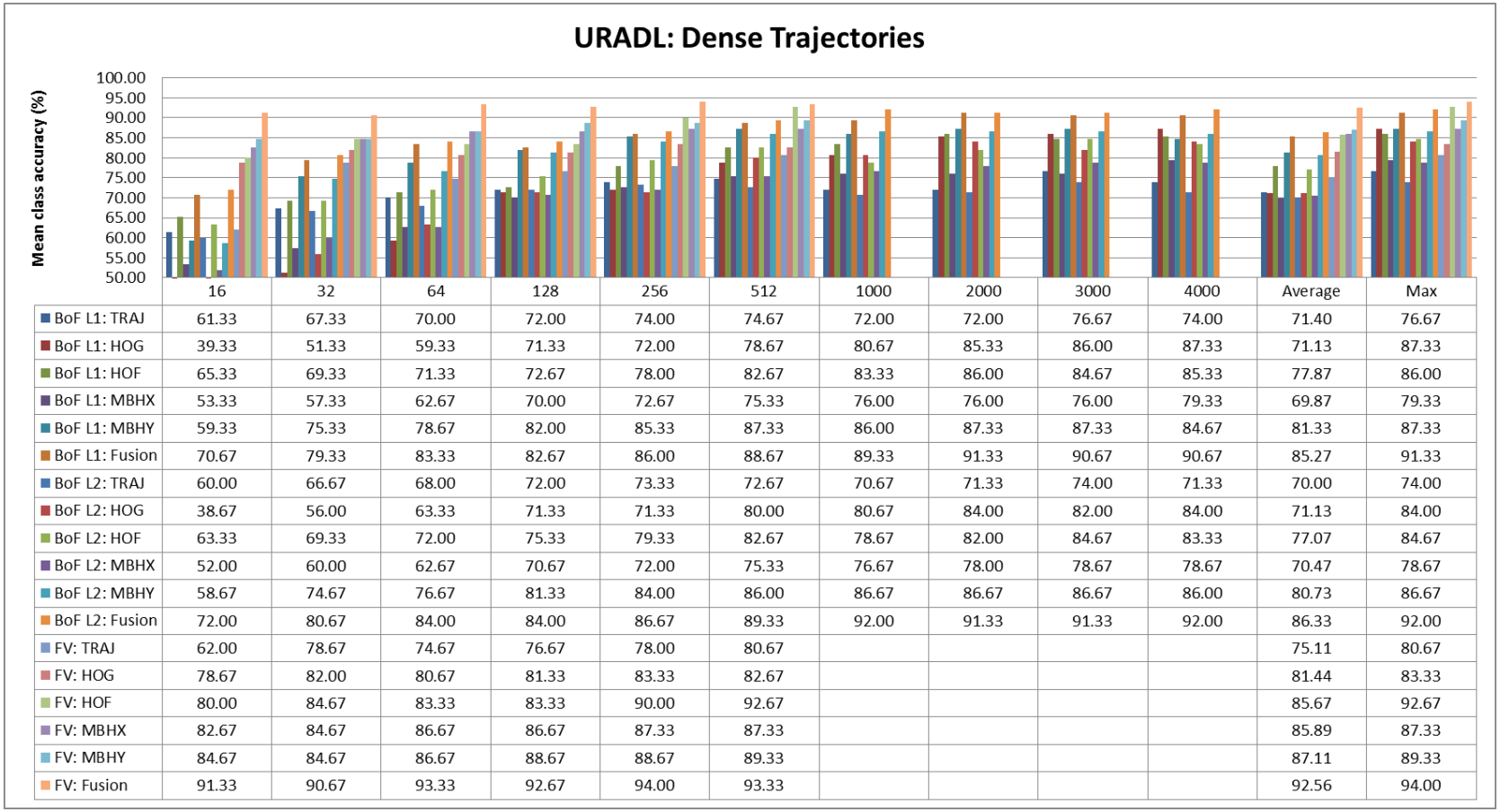
- The detail evaluation results are presented in Figure 3.16.
- The bag-of-features with the L2 norm achieve slightly better results than the bag-of-features with the L1 norm.
- For every evaluated local feature encoding technique: the HOG descriptor works better than the Trajectory shape descriptor, the HOF descriptor works better than the HOG descriptor, and the fusion of all the descriptors significantly improves the results over the individual descriptors.
- The MBH descriptor works better than the HOF descriptor for the bag-of-features, but worse for the Fisher vectors.
- The MBH-y component works better than the MBH-x component.
- The best result, 94% of mean class accuracy, is achieved by the fusion all the descriptors, Fisher vectors, and the codebook size 256.

3.4.2.3 MSR Daily Activity 3D Dataset

- The detail evaluation results are presented in Figure 3.17.
- The MBH descriptor works better than the HOG descriptor, the Trajectory shape descriptor works better than the MBH descriptor, the HOF works better than the Trajectory shape descriptor, and the fusion of all the descriptors significantly improves the results over the individual descriptors.
- The MBH-y component works better than the MBH-x component.
- The best result, 76.25% of mean class accuracy, is achieved by the fusion of all the descriptors and the codebook size 32.

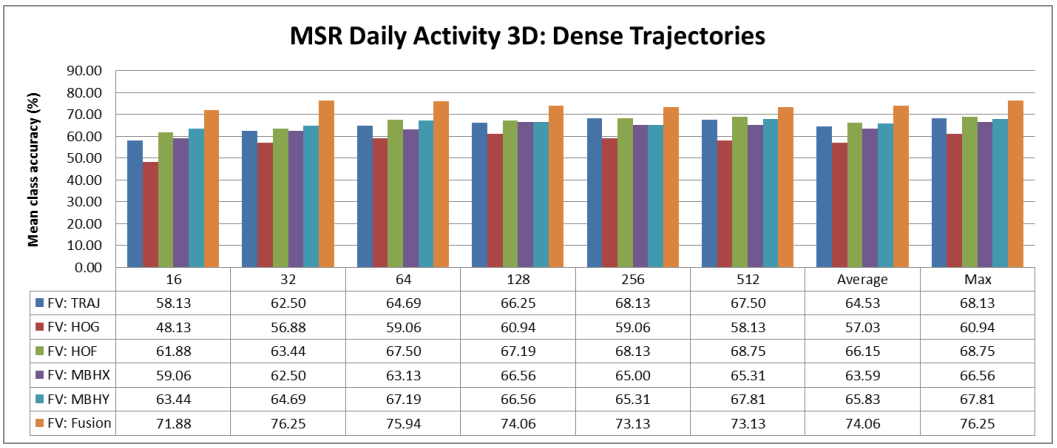
3.4.2.4 HMDB51 Dataset

- The detail evaluation results are presented in Figure 3.18.
- The HOG descriptor works better than the Trajectory shape descriptor, the HOF descriptor works better than the HOG descriptor, the MBH descriptor works better than the HOF descriptor, and the fusion of all the descriptors significantly improves the results over the individual descriptors.



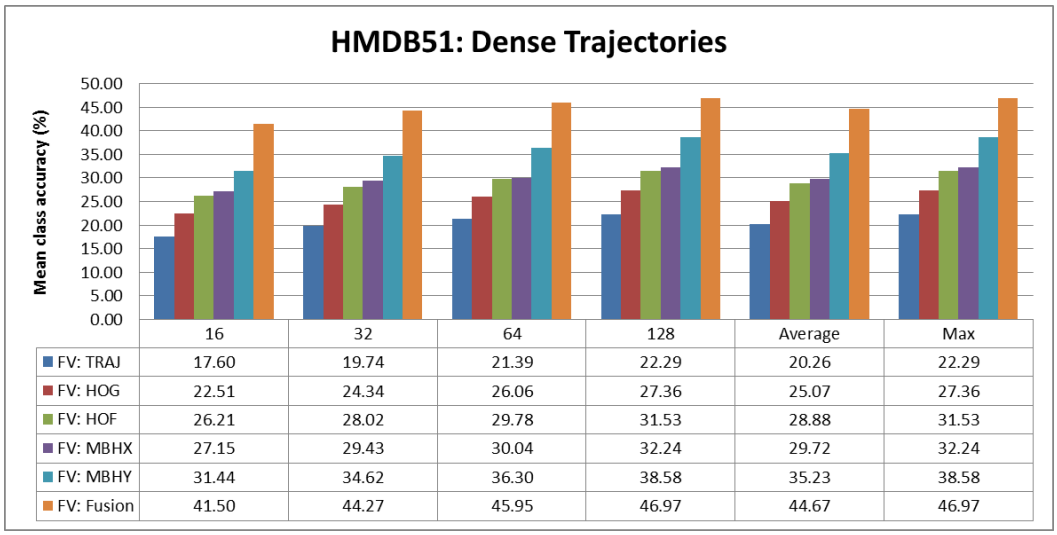
	16	32	64	128	256	512	1000	2000	3000	4000	Average	Max
BoF L1: TRAJ	61.33	67.33	70.00	72.00	74.00	74.67	72.00	72.00	76.67	74.00	71.40	76.67
BoF L1: HOG	39.33	51.33	59.33	71.33	72.00	78.67	80.67	85.33	86.00	87.33	71.13	87.33
BoF L1: HOF	65.33	69.33	71.33	72.67	78.00	82.67	83.33	86.00	84.67	85.33	77.87	86.00
BoF L1: MBHX	53.33	57.33	62.67	70.00	72.67	75.33	76.00	76.00	76.00	79.33	69.87	79.33
BoF L1: MBHY	59.33	75.33	78.67	82.00	85.33	87.33	86.00	87.33	87.33	84.67	81.33	87.33
BoF L1: Fusion	70.67	79.33	83.33	82.67	86.00	88.67	89.33	91.33	90.67	90.67	85.27	91.33
BoF L2: TRAJ	60.00	66.67	68.00	72.00	73.33	72.67	70.67	71.33	74.00	71.33	70.00	74.00
BoF L2: HOG	38.67	56.00	63.33	71.33	71.33	80.00	80.67	84.00	82.00	84.00	71.13	84.00
BoF L2: HOF	63.33	69.33	72.00	75.33	79.33	82.67	78.67	82.00	84.67	83.33	77.07	84.67
BoF L2: MBHX	52.00	60.00	62.67	70.67	72.00	75.33	76.67	78.00	78.67	78.67	70.47	78.67
BoF L2: MBHY	58.67	74.67	76.67	81.33	84.00	86.00	86.67	86.67	86.67	86.00	80.73	86.67
BoF L2: Fusion	72.00	80.67	84.00	84.00	86.67	89.33	92.00	91.33	91.33	92.00	86.33	92.00
FV: TRAJ	62.00	78.67	74.67	76.67	78.00	80.67					75.11	80.67
FV: HOG	78.67	82.00	80.67	81.33	83.33	82.67					81.44	83.33
FV: HOF	80.00	84.67	83.33	83.33	90.00	92.67					85.67	92.67
FV: MBHX	82.67	84.67	86.67	86.67	87.33	87.33					85.89	87.33
FV: MBHY	84.67	84.67	86.67	88.67	88.67	89.33					87.11	89.33
FV: Fusion	91.33	90.67	93.33	92.67	94.00	93.33					92.56	94.00

Figure 3.16 – URADL dataset: Evaluation results of the Dense Trajectories with 3 local feature encoding techniques (the Bag-of-Features (BoF) with L1 and L2 norms, and the Fisher vectors (FV)). The plot presents the mean class accuracy of the Trajectory shape descriptor (TRAJ), Histogram of Oriented Gradients (HOG), Histogram of Optical Flow (HOF), Motion Boundary Histogram (MBH) for "x" and "y" components, and fusion of all these descriptors, with respect to the codebook size (the "x" axis).



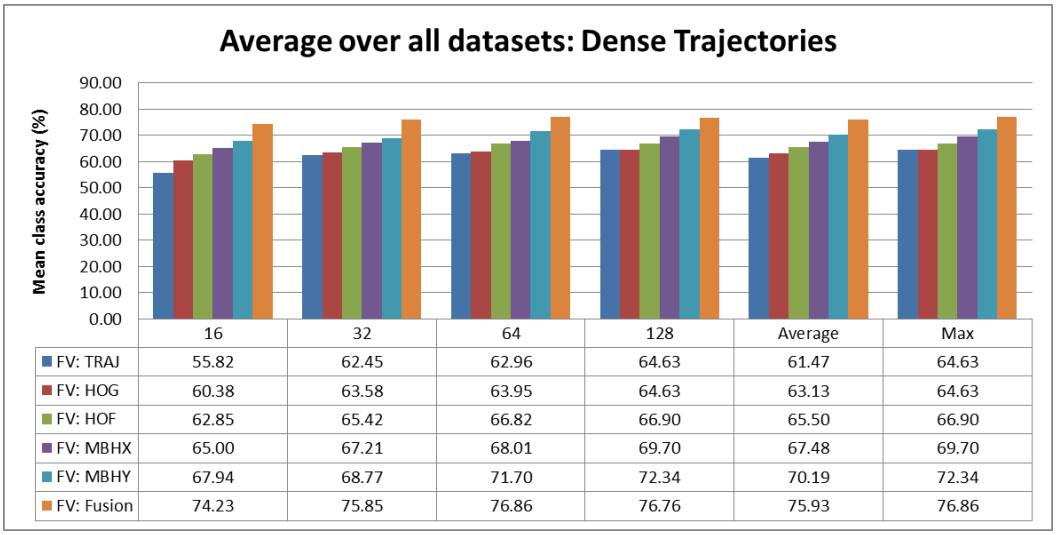
	16	32	64	128	256	512	Average	Max
FV: TRAJ	58.13	62.50	64.69	66.25	68.13	67.50	64.53	68.13
FV: HOG	48.13	56.88	59.06	60.94	59.06	58.13	57.03	60.94
FV: HOF	61.88	63.44	67.50	67.19	68.13	68.75	66.15	68.75
FV: MBHX	59.06	62.50	63.13	66.56	65.00	65.31	63.59	66.56
FV: MBHY	63.44	64.69	67.19	66.56	65.31	67.81	65.83	67.81
FV: Fusion	71.88	76.25	75.94	74.06	73.13	73.13	74.06	76.25

Figure 3.17 – MSR Daily Activity 3D dataset: Evaluation results of the Dense Trajectories with the Fisher vectors. The plot presents the mean class accuracy of the Trajectory shape descriptor (TRAJ), Histogram of Oriented Gradients (HOG), Histogram of Optical Flow (HOF), Motion Boundary Histogram (MBH) for "x" and "y" components, and fusion of all these descriptors, with respect to the codebook size (the "x" axis).



	16	32	64	128	Average	Max
FV: TRAJ	17.60	19.74	21.39	22.29	20.26	22.29
FV: HOG	22.51	24.34	26.06	27.36	25.07	27.36
FV: HOF	26.21	28.02	29.78	31.53	28.88	31.53
FV: MBHX	27.15	29.43	30.04	32.24	29.72	32.24
FV: MBHY	31.44	34.62	36.30	38.58	35.23	38.58
FV: Fusion	41.50	44.27	45.95	46.97	44.67	46.97

Figure 3.18 – HMDB51 dataset: Evaluation results of the Dense Trajectories with the Fisher vectors. The plot presents the mean class accuracy of the Trajectory shape descriptor (TRAJ), Histogram of Oriented Gradients (HOG), Histogram of Optical Flow (HOF), Motion Boundary Histogram (MBH) for "x" and "y" components, and fusion of all these descriptors, with respect to the codebook size (the "x" axis).



	16	32	64	128	Average	Max
FV: TRAJ	55.82	62.45	62.96	64.63	61.47	64.63
FV: HOG	60.38	63.58	63.95	64.63	63.13	64.63
FV: HOF	62.85	65.42	66.82	66.90	65.50	66.90
FV: MBHX	65.00	67.21	68.01	69.70	67.48	69.70
FV: MBHY	67.94	68.77	71.70	72.34	70.19	72.34
FV: Fusion	74.23	75.85	76.86	76.76	75.93	76.86

Figure 3.19 – Average over all dataset: Evaluation results of the Dense Trajectories with the Fisher vectors. The plot presents the mean class accuracy of the Trajectory shape descriptor (TRAJ), Histogram of Oriented Gradients (HOG), Histogram of Optical Flow (HOF), Motion Boundary Histogram (MBH) for "x" and "y" components, and fusion of all these descriptors, with respect to the codebook size (the "x" axis).

- The MBH descriptor works particularly well.
- The MBH-y component works better than the MBH-x component.
- The best result, 46.97% of mean class accuracy, is achieved by the fusion of all the descriptors and the codebook size 128.

3.4.2.5 Overall

- The detail evaluation results are presented in Figure 3.19.
- The HOF descriptor works better than the Trajectory shape descriptor and the HOG descriptor, the MBH descriptor works better than the HOF descriptor, and the fusion of all the descriptors improves the results over the individual descriptors (except for the Weizmann dataset, for which the MBH descriptors work particularly good).

3.4.2.6 Results Summary and Analysis

- The motion based descriptors work better than the appearance descriptors.
- The motion information alone is not good enough to achieve good results. The fusion of motion and appearance information improves action recognition accuracy.
- The MBH descriptor works particularly well for the HMDB51 dataset, it works better than the HOF descriptor. However, the MBH descriptor works worse than the HOF

descriptor for the MSR Daily Activity 3D dataset and the URADL dataset. The HMDB51 dataset contains a lot of camera motion, whereas the MSR Daily Activity 3D dataset and the URADL dataset are recorded with static cameras.

- Fisher vector encoding works better than the bag-of-features approach.
- The bag-of-features with the L2 norm work slightly better than the bag-of-features with the L1 norm (*e.g.* 92% vs. 91.33% for the URADL dataset).
- In general, there is no codebook size which always performs the best for the bag-of-features and the Fisher vectors.

3.4.3 Summary and Conclusion

	HMDB51	MSR	URADL	Weizmann	Average
Spatio-Temporal Interest Points	31.53%	72.50%	94.00%	87.78%	70.44%
Dense Trajectories	46.97%	76.25%	94.00%	93.33%	76.68%

Table 3.10 – The table presents the mean class accuracy of the Spatio-Temporal Interest Points and the Dense Trajectories, using fusion of their descriptors and Fisher vectors. The comparison is presented for the HMDB51, MSR Daily Activity 3D (MSR), URADL, and Weizmann datasets, along with the average results over all these dataset.

The general summary and conclusion are as follows:

- The motion based descriptors work better than the appearance descriptors. It seems natural that the motion information is very important for action recognition, and often more important than the appearance information.
- The motion information alone is not good enough to achieve good results. The fusion of motion and appearance information improves action recognition accuracy. Both motion and appearance are necessary for action recognition. The motion information is very important for action recognition, however, in some cases it might have difficulty distinguishing similar actions, *e.g.* eating a banana from eating snack chips, whereas the appearance information might easily distinguish these actions from each other.
- The MBH descriptor is particularly useful and works well when the camera motion occurs (*e.g.* the HMDB51 dataset). This is because the MBH descriptor computes motion boundaries by a derivative operation on the optical flow. Thus, motion due to locally translational camera movement is canceled out and relative motion is captured.

- Fisher vector encoding works better than the bag-of-features approach. The main cause why the Fisher vector encoding works better than the bag-of-features approach may be the loss of information by the bag-of-features approach when doing the hard assignment of local features to visual words.
- The Dense Trajectories with the Fisher vector encoding work better than the Spatio-Temporal Interest Points with the Fisher vector encoding (see Table 3.10).
- The bag-of-features with the L2 norm work better than the bag-of-features with the L1 norm, and sometimes the difference in the accuracy is large (*e.g.* 64.69% vs. 60% for the Spatio-Temporal Interest Points and the MSR Daily Activity 3D dataset).
- In general, there is no codebook size which always performs the best for the bag-of-features and the Fisher vectors.

3.5 Conclusion

In this chapter, we have introduced the action recognition framework, which consists of three steps: local spatio-temporal video features extraction, video-action representation, and video-action recognition. We have reviewed the popular techniques for each of the steps. Then, we have presented five popular state-of-the-art action recognition datasets and we have proposed a new action recognition dataset. The datasets are presented with their statistical analysis. Finally, we have performed the extensive evaluation, comparison, and analysis of the presented techniques.

CHAPTER 4

Video Covariance Matrix Logarithm

Contents

4.1 Introduction 100
4.2 Video Covariance Matrix Logarithm Descriptor 103
4.2.1 Video Frame Descriptor 103
4.2.2 Pixel-Level Features 105
4.2.3 Video Volume Descriptor 108
4.2.4 Riemannian Geometry 109
4.2.5 Fast Covariance Matrix Calculation 112
4.3 Approach Overview 114
4.4 Experiments 116
4.4.1 Weizmann Dataset 117
4.4.2 URADL Dataset 119
4.4.3 MSR Daily Activity 3D Dataset 123
4.4.4 HMDB51 Dataset 123
4.4.5 Results Summary and Analysis 126
4.5 Conclusion 126

In this chapter, we propose a new local spatio-temporal descriptor for videos, and we propose a new approach for action recognition in videos based on the introduced descriptor. The new descriptor is called the Video Covariance Matrix Logarithm (VCML). The VCML descriptor is based on a covariance matrix representation, and it models linear relationships between different low-level features, such as intensity and gradient. We apply the VCML descriptor to encode appearance information of local spatio-temporal video volumes, which are extracted by the Dense Trajectories. Then, we present an extensive evaluation of the proposed VCML descriptor with the Fisher vector encoding and the Support Vector Machines on four various action recognition datasets. We show that the VCML descriptor always achieves better results than the HOG descriptor. Moreover, we present that the VCML descriptor carries complementary information to the HOG descriptor, as their fusion always gives a significant improvement in action recognition accuracy. Finally, we show that the VCML descriptor improves action recognition accuracy in comparison to the state-of-the-art Dense Trajectories.

4.1 Introduction

In Chapter 2, we present the most popular local spatio-temporal feature descriptors for action recognition in videos, which are:

- Histogram of Oriented Gradients (HOG) descriptor [Laptev 2005], which encodes a visual appearance and shape information.
- Histogram of Optical Flow (HOF) descriptor [Laptev 2005] and Motion Boundary Histogram (MBH) descriptor [Wang 2011a], which encode visual motion information.

Then, in Chapter 3, we present the evaluation of the above local spatio-temporal descriptors on various action recognition datasets. We observe that typically the motion based descriptors work better than the appearance descriptor. However, the motion information alone is not good enough to achieve good results, and both motion and appearance are necessary for action recognition.

There are two possible reasons why the motion based descriptors work better than the appearance based descriptors; simply, the motion information is more important for action recognition, or the existing appearance based descriptors are not discriminative enough. As still image based human action recognition techniques [Guo 2014] have shown to achieve good results and they do not use temporal information, we believe that more discriminative descriptors for modeling appearance information could be proposed. Therefore, in this chapter we primary focus on modeling the appearance information.

All the above descriptors, *i.e*. HOG, HOF, and MBH, are based on a 1-dimensional histogram representation of individual features, and they directly model values of given features. However, the joint statistics between individual features are ignored by these descriptors, whereas such information may be informative. Therefore, the above descriptors might not be discriminative enough to recognize some actions.

In image processing, a novel trend has emerged that ignores explicit values of given features, focusing on their pairwise relations instead.

A relation between features is well explained in the covariance, which is a measure of how much random variables change together. Covariance provides a measure of the strength of the correlation between features.

Covariance based features have been introduced by Tuzel *et al*. for object detection and texture classification [Tuzel 2006]. They have been successfully applied also for object tracking [Porikli 2006], shape modeling [Wang 2007], face recognition [Pang 2008], and person re-identification [Bak 2012b].

Moreover, covariance based features have also been applied for action recognition. In [Guo 2010a, Guo 2010b, Guo 2013], Guo *et al.* have modeled a whole video sequence using a covariance based representation, and they have applied a sparse linear representation framework to recognize actions. As input features for covariance calculation, they have applied feature vectors from segments of silhouette tunnels of moving objects [Guo 2010a, Guo 2013], and they have applied optical flow features from moving pixels approximated by thresholding the smoothed temporal gradients [Guo 2010b, Guo 2013]. One of the main drawbacks of the proposed approaches is that they required precise segmentation, which is very difficult to obtain in real world videos.

Instead of modeling a whole video sequence using a covariance based representation [Guo 2010a, Guo 2010b, Guo 2013], Yuan *et al.* [Yuan 2009a] have applied covariance based features for local spatio-temporal interest points proposed by Dollar *et al.* [Dollar 2005]. As input features for covariance calculation, they have applied the position of interest points, a gradient, and an optical flow. Then, the authors have represented each video sequence by an occurrence histogram of covariance based features, and they have applied the Earth Mover's Distance (with the L2 norm as the ground distance) to match pairs of video sequences. Finally, the authors have used the Nearest Neighbor algorithm as a classifier. One of the main limitations of this approach (and also of the approaches proposed by Guo *et al.*) is the lack of any structural information in a descriptor; the authors have modeled a given spatio-temporal video volume using a single covariance based representation. Moreover, the authors have computed video representations with different sizes of histograms, and as the result they have not taken the advantage of powerful metrics developed to match histograms (*e.g.* χ^2 distance and histogram intersection distance).

Different from the existing techniques, we introduce a new local spatio-temporal descriptor for videos and we propose a new approach for action recognition based on the introduced descriptor:

- Descriptor: We introduce a new local spatio-temporal descriptor for videos, called the Video Covariance Matrix Logarithm (VCML). The VCML descriptor is based on a covariance matrix representation, and it models linear relationships between different pixel-level features. The VCML descriptor can be used to represent any low-level features, such as visual appearance and motion features. In order to encode structural information of the video volume, we use the spatio-temporal grid, and we compute a covariance representation for each cell of the grid. Although the covariance matrix does not lie on the Euclidean space, the VCML descriptor is mapped to the Euclidean space, and it can be used with any action recognition framework using the Euclidean space.

- Approach: We compute the Dense Trajectories in a video sequence, and we propose to extract local spatio-temporal video volumes around the trajectories. Then, we propose to represent the appearance information of each local spatio-temporal video volume by our VCML descriptor. Moreover, we extract the Trajectory shape,

Histogram of Oriented Gradients, Histogram of Optical Flow, and Motion Boundary Histogram descriptors for each local spatio-temporal video volume. Then, we apply the Fisher vector encoding to represent videos, we fuse the obtained video representations, and we use the Support Vector Machines for action classification.

- Experiments: We present an extensive evaluation of our descriptor and our approach on four various state-of-the-art datasets. We show that the VCML descriptor always achieves better results than the HOG descriptor. Moreover, we present that the VCML descriptor carries complementary information to the HOG descriptor, as their fusion always gives a significant improvement in action recognition accuracy. Finally, we show that the VCML descriptor improves the action recognition in comparison to the state-of-the-art Dense Trajectories.

The main differences between the proposed VCML approach and the state-of-the-art action recognition approaches based on covariance features are:

- VCML vs. [Guo 2010a, Guo 2010b, Guo 2013]: The VCML approach does not require segmentation, which is very difficult to obtain in real world videos.

- VCML vs. [Yuan 2009a]: The VCML approach encodes structural information of a video volume, and for each cell of the grid a smaller size descriptor is calculated ($\frac{d(d+1)}{2}$ vs. d^2). The VCML approach creates video representations with the same size of distributions, which facilitates the matching of video representations. Moreover, the VCML approach extracts local spatio-temporal video volumes using the Dense Trajectories (which have shown superior results over the interest points), it is based on the Fisher vector encoding (which has shown to outperform the histogram encoding), and it uses the Support Vector Machines classifier (which has shown results superior to the Nearest Neighbor algorithm).

The remainder of the chapter is organized as follows. In Section 4.2, we propose the Video Covariance Matrix Logarithm descriptor. Section 4.3 presents our action recognition framework. In Section 4.4, we present experimental results, comparison, and analysis. Finally, we conclude in Section 4.5.

4.2 Video Covariance Matrix Logarithm Descriptor

In this section, we propose a new descriptor to encode a local spatio-temporal video volume. The new descriptor is called the Video Covariance Matrix Logarithm (VCML). It is based on a covariance matrix representation, and it models linear relationships between different pixel-level features.

Similarly to the most popular and powerful action recognition local spatio-temporal descriptors, *i.e*. HOG, HOF, and MBH descriptors, we base our descriptor on the representation of individual frames.

In Section 4.2.1, we propose a video frame descriptor, and in Section 4.2.2, we present the low-level, *i.e*. pixel-level, features that we use to compute the video frame descriptor. Then, in Section 4.2.3, we propose a video volume descriptor, which is an extension of the video frame descriptor to the spatio-temporal domain. Section 4.2.4 presents a brief introduction to the Riemannian geometry and distance metrics for covariance matrices. Finally, we present the method for fast covariance matrix calculation in Section 4.2.5.

4.2.1 Video Frame Descriptor

We are given a single video frame t of spatial size $n_x \times n_y$ pixels, and our goal is to create its discriminative and compact representation.

The overview of the calculation process of the proposed video frame descriptor for a sample image is presented in Figure 4.1.

Firstly, we calculate low-level (*i.e*. pixel-level) features, *e.g*. intensities in red, green, and blue channels (see Section 4.2.2). For each pixel of a given video frame, we extract d low-level features. Therefore, we represent a video frame t by a set $\{f_{(x,y,t)}\}_{1 \leq x \leq n_x, 1 \leq y \leq n_y}$ of d-dimensional feature vectors ($f_{(x,y,t)} \in \mathcal{R}^d$). Such a frame representation is typically of high dimension ($n_x \times n_y \times d$), and thus it is necessary to transform it into a more compact representation.

For simplicity, we denote the set $\{f_{(x,y,t)}\}_{1 \leq x \leq n_x, 1 \leq y \leq n_y}$ as $\{f_{(k,t)}\}_{k=1 \ldots n}$, where n is the number of pixels in each video frame ($n = n_x \times n_y$).

We propose to represent each video frame t via covariance matrix (also known as dispersion matrix or variance-covariance matrix). The covariance matrix encodes the variance within each feature and the covariance between different features. The covariance matrix is defined as:

$$C_t = \frac{1}{n-1} \sum_{k=1}^{n} (f_{(k,t)} - \mu_t)(f_{(k,t)} - \mu_t)^T, \tag{4.1}$$

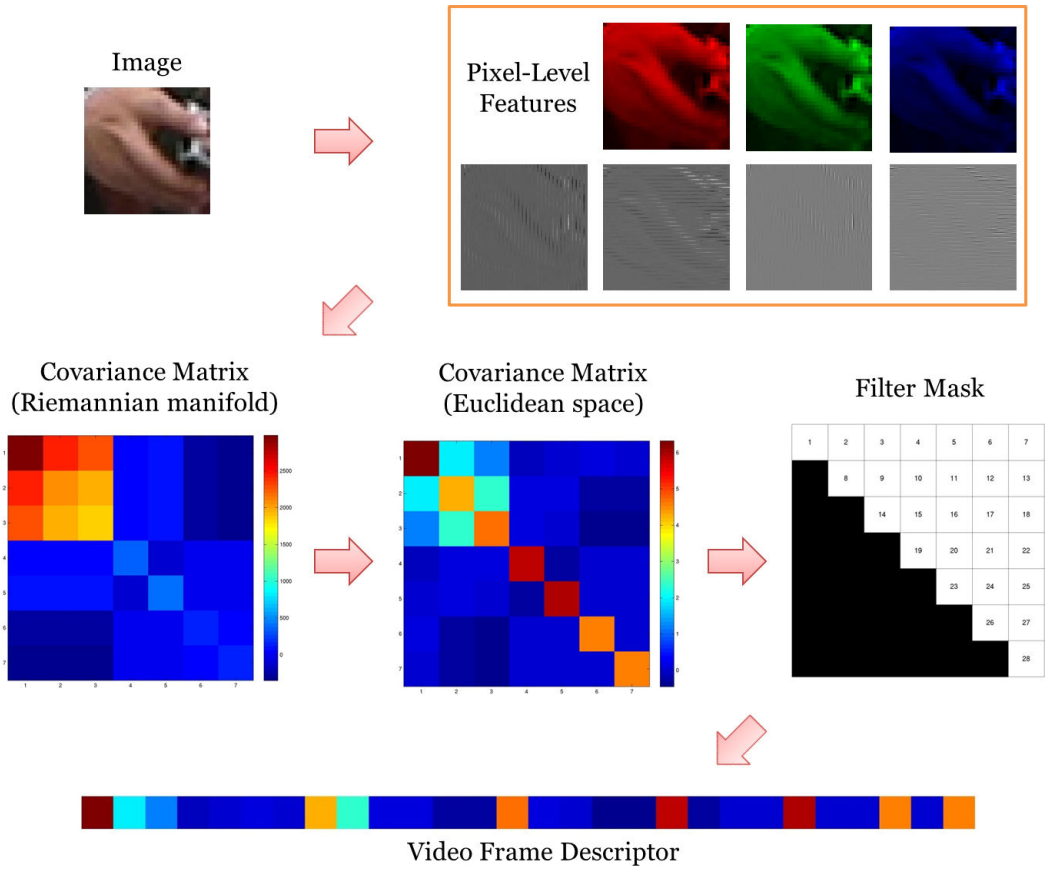


Figure 4.1 – Overview of the calculation process of the video frame descriptor for a sample image. Firstly, we extract pixel-level features of an input image. Then, we represent this image using the covariance matrix and the extracted pixel-level features. We map the covariance matrix from the Riemannian manifold to the Euclidean space. Then, we use the symmetric property of the covariance matrix and we apply a filter mask extracting all the entries of the upper triangular part of the covariance matrix. We represent these entries in a form of a vector, called the video frame descriptor.

where μ_t is the mean of the feature vectors:

$$\mu_t = \frac{1}{n} \sum_{k=1}^{n} f_{(k,t)}. \tag{4.2}$$

Therefore, we transform a video frame representation of size $n_x \times n_y \times d$ into a tensor C_t of size $d \times d$.

Typically, local feature encoding techniques, such as the bag-of-features approach and the Fisher vector encoding, use the Euclidean space to compare features. The main problem is that covariance matrices do not lie on the Euclidean space, but instead they lie on the Riemannian manifold (see Section 4.2.4). Therefore, we need a metric to compare two covariance matrices. To solve this issue we use the log-Euclidean Riemannian metric [Arsigny 2006] (see Section 4.2.4), based on which we can map covariance matrices to the Euclidean space using the matrix logarithm operation (see Section 4.2.4.4). Using the log-Euclidean Riemannian metric, the new representation of the covariance matrix is:

$$C_t^{(log)} = \log(C_t), \tag{4.3}$$

where $\log(C_t)$ is the matrix logarithm operation applied for the covariance matrix C_t.

The covariance matrix is a symmetric matrix, and thus it is determined by $\frac{d(d+1)}{2}$ values, forming the upper or lower triangular part of the covariance matrix. To represent a single video frame and create its compact representation, we apply a filter mask extracting all the entries on and above (below) the diagonal of the covariance matrix. We represent these values in a form of a vector V_t (see Figure 4.1):

$$V_t = \mathrm{triu}(\log(C_t)), \tag{4.4}$$

where $\mathrm{triu}(\cdot)$ is the filter mask operation.

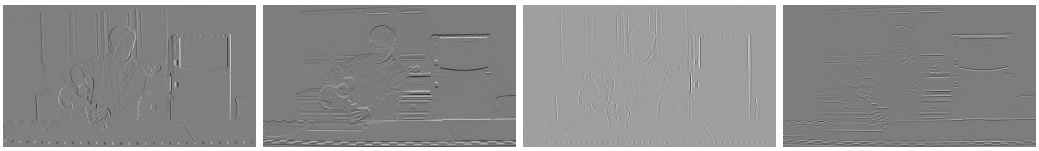
Therefore, we transform a video frame representation of size $n_x \times n_y \times d$ into a compact vector V_t of size $\frac{d(d+1)}{2}$. The obtained feature vector V_t is called the video frame descriptor.

4.2.2 Pixel-Level Features

In this section, we present the extraction of low-level, *i.e.* pixel-level, features in a single video frame. As mentioned before, we focus on the representation of the appearance information.

For every pixel in each frame of the given video volume, we extract seven low-level appearance features. We extract normalized intensities in red, green, and blue channels,

Original image | Red channel | Green channel | Blue channel



1^{st} deriv. axis x | 1^{st} deriv. axis y | 2^{nd} deriv. axis x | 2^{nd} deriv. axis y

Figure 4.2 – Seven low-level appearance features extracted in a sample video frame from the URADL dataset.

and first and second order derivatives of gray scale intensity image along "x" and "y" axes. Thus, every pixel is represented in the following form:

$$f = \left[R,\ G,\ B,\ \frac{\partial I}{\partial x},\ \frac{\partial I}{\partial y},\ \frac{\partial^2 I}{\partial x^2},\ \frac{\partial^2 I}{\partial y^2} \right], \tag{4.5}$$

where R, G, and B are the red, green, and blue intensity channels, and I is the corresponding gray scale intensity image. An example of the extracted seven low-level appearance features is presented in Figure 4.2.

We also visualize sample covariance matrices in order to validate if the selected low-level features can be discriminative. We calculate covariance matrices using defined low-level visual appearance features, and we map the calculated covariance matrices to the Euclidean space using the matrix logarithm operation. The sample covariance matrices for 10 different action categories are presented in Figure 4.3. We notice some similarities of covariance matrices belonging to the same action categories and some differences of covariance matrices belonging to different action categories.

The covariance representation based on the above seven low-level features provides a rotation invariant representation of a video frame. However, the relationships between these low-level features and the spatial positions of these features may be informative and useful for action recognition. Therefore, we also use the extended set of low-level features, where every pixel is represented in the following form:

$$f' = \left[X,\ Y,\ R,\ G,\ B,\ \frac{\partial I}{\partial x},\ \frac{\partial I}{\partial y},\ \frac{\partial^2 I}{\partial x^2},\ \frac{\partial^2 I}{\partial y^2} \right], \tag{4.6}$$

where X and Y represent the spatial position of a pixel in a video frame, and the remaining pixel-level features are presented in Equation 4.5.

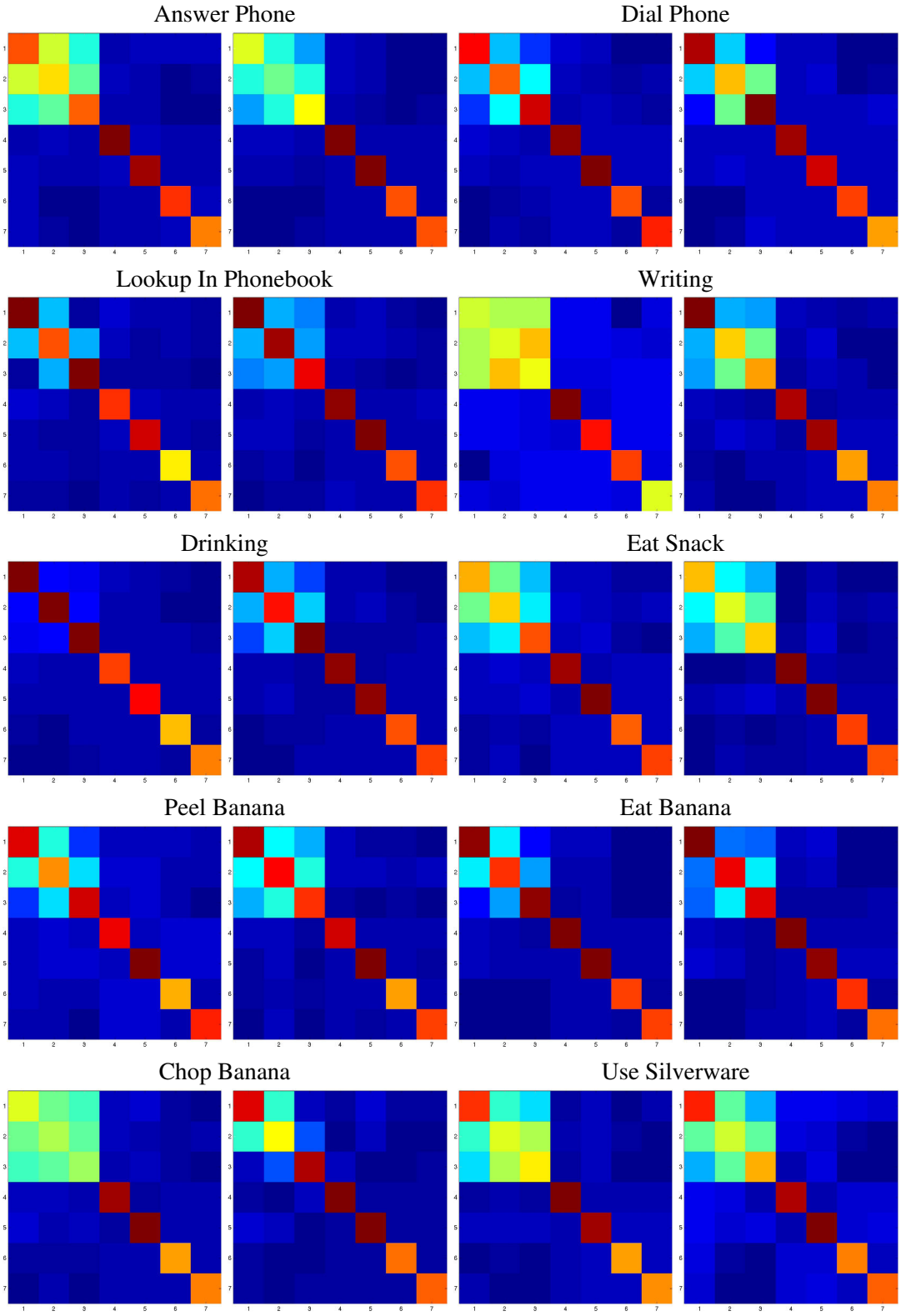


Figure 4.3 – URADL dataset: 2 sample normalized covariance matrices mapped to the Euclidean space are presented for 10 different action categories. The covariance matrices present relations between seven pixel-level features, which are defined in Equation 4.5.

4.2.3 Video Volume Descriptor

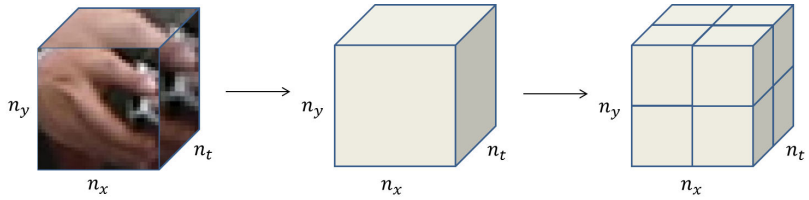


Figure 4.4 – An input spatio-temporal video volume is treated as a cuboid, and the cuboid is divided into a spatio-temporal grid with cells of equal dimensions.

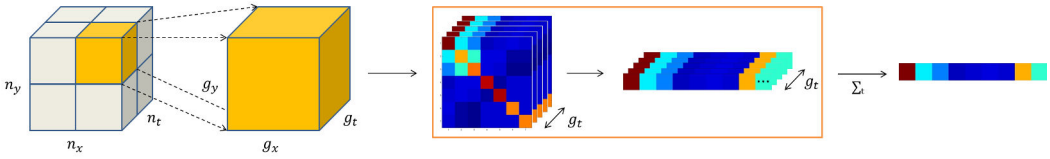


Figure 4.5 – Overview of the calculation process of the VCML descriptor for a sample spatio-temporal video volume. Firstly, a video volume is divided into a spatio-temporal grid. We encode each video frame in each cell of the grid by the video frame descriptor. Then, we describe each cell of the grid as a mean of all video frame representations calculated inside this cell. Finally, we define the VCML descriptor as a concatenation of all the descriptors from all cells of the grid.

We are given a spatio-temporal video volume of size $n_x \times n_y \times n_t$, of spatial size $n_x \times n_y$ pixels, and of temporal size n_t video frames. Our goal is to create its discriminative and compact representation.

Firstly, we use the spatio-temporal grid to encode structural information of the video volume. Thus, we treat an input spatio-temporal video volume as a cuboid, and we divide it into a spatio-temporal grid (see Figure 4.4), where each cell of the grid is of size $g_x \times g_y \times g_t$, of spatial size $g_x \times g_y$ pixels, and of temporal size g_t video frames.

The overview of the calculation process of the Video Covariance Matrix Logarithm (VCML) descriptor for a sample spatio-temporal video volume, which is divided into a spatio-temporal grid, is presented in Figure 4.5.

For each video frame in each cell of the grid, we compute a separate video frame descriptor V_t, as explained in Section 4.2.1. Then, to create a compact cell representation, we describe each cell of the grid as a mean of all video frame representations calculated inside this cell:

$$V_{cell} = \frac{1}{g_t} \sum_{t=1}^{g_t} \text{triu}(\log(C_t)). \tag{4.7}$$

Finally, we define the Video Covariance Matrix Logarithm (VCML) descriptor D as the concatenation of all the descriptors from all cells of the grid:

$$D = [V_{cell_1}, V_{cell_2}, ..., V_{cell_m}]^T, \tag{4.8}$$

where m is the number of cells of the spatio-temporal grid.

4.2.4 Riemannian Geometry

Covariance matrices are symmetric and positive semidefinite (nonnegative definite) matrices. The main problem with the covariance matrices is that the tensor space of the covariance matrices is a manifold, that is not a vector space with the usual additive structure.

A manifold [Tuzel 2008] is a topological space that is locally similar to an Euclidean space. Every point on the d-dimensional manifold has a neighborhood, for which there exists a homeomorphism (one-to-one, onto and continuous mapping in both directions), mapping the neighborhood to the d-dimensional space $\mathbb{R}^d$.

A Riemannian manifold $\mathcal{M}$ [Tuzel 2008] is a differentiable manifold, in which each tangent space has an inner product $<,>_{\mathbf{X}\in\mathcal{M}}$, which varies smoothly from point to point. The inner product induces a norm for the tangent vectors in the tangent space such that $||y||^2_{\mathbf{X}} =< y, y >_{\mathbf{X}}$.

Covariance matrices can be represented as a connected Riemannian manifold. Since the Euclidean norm does not correctly capture the distance between two covariance matrices, we need to apply a Riemannian metric in order to use the covariance matrix based descriptors with a local feature encoding technique.

There are two popular distance metrics for covariance matrices, which are defined on the Riemannian manifold:

- Affine-Invariant Riemannian Metric (Section 4.2.4.1),
- Log-Euclidean Riemannian Metric (Section 4.2.4.2).

Our choice of the Riemannian metric is presented in Section 4.2.4.3.

4.2.4.1 Affine-Invariant Riemannian Metric

The affine-invariant Riemannian metric was proposed by Forstner and Moonen [Förstner 1999]. This metric defines a distance between two covariance matrices C_i and C_j of size $d \times d$ as:

$$dist(C_i, C_j) = \|\log(C_j^{-1}C_i)\|_F = \sqrt{\sum_{k=1}^{d} \log^2 \lambda_k(C_i, C_j)}, \tag{4.9}$$

where $\log(\cdot)$ is the matrix logarithm (see Section 4.2.4.4), $\|\cdot\|_F$ is the Frobenius norm of a matrix, and $\{\lambda_k(C_i, C_j)\}_{1 \leq k \leq d}$ are the generalized eigenvalues of C_i and C_j, *i.e.*:

$$\forall_{k=1..d} : \lambda_k C_i \mathbf{u}_k = C_j \mathbf{u}_k, \tag{4.10}$$

where $\mathbf{u}_k \neq 0$ is the k-th generalized eigenvector.

4.2.4.2 Log-Euclidean Riemannian Metric

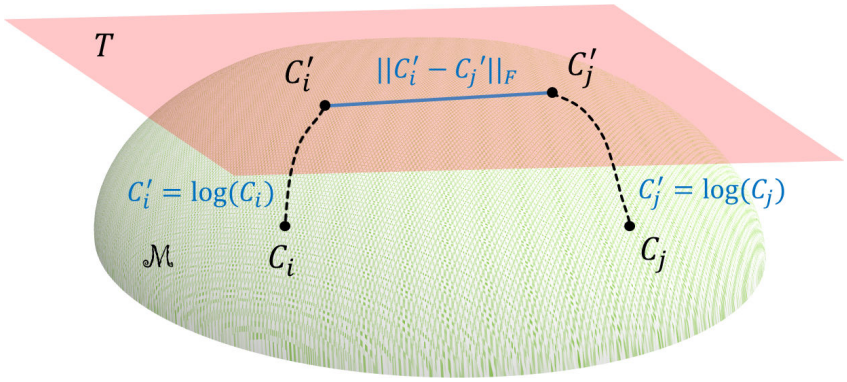


Figure 4.6 – Overview of the Log-Euclidean Riemannian Metric. Two sample covariance matrices C_i and C_j are projected from a two-dimensional manifold $\mathcal{M}$ to the tangent space $\mathcal{T}$ via matrix logarithm operation $\log(\cdot)$. Then, the difference between the two projected covariance matrices is calculated, and the Frobenius norm $\|\cdot\|_F$ is applied to the result.

The log-Euclidean Riemannian metric was proposed by Arsigny *et al.* [Arsigny 2006]. This metric defines a distance between two covariance matrices C_i and C_j as:

$$dist(C_i, C_j) = \|\log(C_i) - \log(C_j)\|_F, \tag{4.11}$$

where $\log(\cdot)$ is the matrix logarithm (see Section 4.2.4.4), and $\|\cdot\|_2$ is the Frobenius norm of a matrix. The overview of the metric calculation process is presented in Figure 4.6.

4.2.4.3 Riemannian Metric Selection

There are two popular Riemannian metrics that could be applied for covariance matrices: the affine-invariant Riemannian metric and the log-Euclidean Riemannian metric. Both Riemannian metrics provide results very similar to each other [Arsigny 2006], and they have been successfully applied for many Computer Vision topics.

The affine-invariant Riemannian metric has been applied *e.g.* for: pedestrian detection [Tuzel 2008], object detection and texture classification [Tuzel 2006], object tracking [Porikli 2006], and person re-identification [Bak 2012a].

The log-Euclidean Riemannian metric has been applied *e.g.* for: visual tracking [Li 2008], human detection, texture classification, and object tracking [Li 2012].

Our goal is to use the covariance matrix based features with a local feature encoding technique. Therefore, we need to create a codebook, and the codebook is typically created using a clustering algorithm. Since covariance matrices do not form a Euclidean vector space, standard clustering algorithms cannot be used effectively. Clustering on the Riemannian manifold is still an open research problem.

Therefore, we use the log-Euclidean Riemannian metric. According to it, we can map covariance matrices from the Riemannian manifold to the Euclidean space using the matrix logarithm operation.

4.2.4.4 Matrix Logarithm

Given a covariance matrix C of size $n \times n$, we apply the Singular Value Decomposition (SVD) [Strang 2009]. The SVD decomposes the covariance matrix into three matrices:

$$C = U\Sigma U^T, \tag{4.12}$$

where U is the orthonormal matrix of size $n \times n$, and Σ is the square diagonal matrix with nonnegative real numbers, eigenvalues, on the diagonal, and it is of size $n \times n$:

$$\Sigma = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_n \end{bmatrix}. \tag{4.13}$$

Then, the matrix logarithm can be defined as:

$$\log(C) = U\Sigma' U^T, \tag{4.14}$$

where Σ' is the square diagonal matrix of size $n \times n$ with logarithm values of eigenvalues on the diagonal, *i.e.*:

$$\Sigma' = \begin{bmatrix} \log(\lambda_1) & 0 & \dots & 0 \\ 0 & \log(\lambda_2) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \log(\lambda_n) \end{bmatrix}. \tag{4.15}$$

4.2.5 Fast Covariance Matrix Calculation

In Section 3.2.1.2, we present the Dense Trajectories approach, which can be used to extract local spatio-temporal video volumes. The Dense Trajectories approach samples feature points in each video frame on a dense grid, with the distance between the points (*i.e.* the step size) of W pixels. The parameter W is usually set to 5 pixels and the dense sampling is applied on multiple spatial scales (at most 8 spatial scales). It means that applying the dense sampling on just a single spatial scale on a video sequence with 640×480 pixels spatial resolution, we can potentially calculate up to $\frac{640-5}{5} \times \frac{480-5}{5} = 127 \times 95 = 12065$ covariance matrices in a single video frame. The default spatial size of a local spatio-temporal video volume is 32×32 and the default temporal length of a local spatio-temporal video volume is 15. It means that a great many mathematical operations might be required to calculate covariance matrices for all the spatio-temporal video volumes extracted in a video sequence. Therefore, in this section we present a technique for fast covariance matrix calculation, which is based on the integral images [Tuzel 2008].

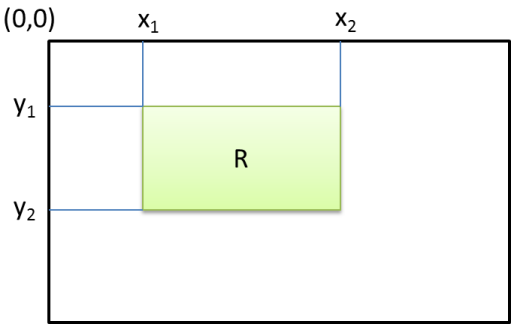


Figure 4.7 – A rectangular subset of a grid, an image.

A summed area table is a data structure and an algorithm, which allows for fast calculation of the sum of values in a rectangular subset of a grid (see Figure 4.7). The summed area table was first introduced to computer graphics in 1984 by Frank Crow [Crow 1984]. In the Image Processing domain, the summed area table is known as the integral image, and it was first introduced for object detection in 2001 by Viola and Jones [Viola 2001].

An integral image S (a summed area table) of an image I is defined as:

$$S(x', y') = \sum_{\substack{x \leq x' \\ y \leq y'}} I(x, y), \tag{4.16}$$

and it can be efficiently calculated using a single pass over the image I using the following property:

$$S(x, y) = I(x, y) + S(x-1, y) + S(x, y-1) - S(x-1, y-1), \tag{4.17}$$

with the assumptions: $S(0,\cdot) = 0$ and $S(\cdot,0) = 0$.

Then, the sum of values in a rectangular subset R of an image I can be calculated using 4 accesses to the structure S and 3 simple mathematical operations:

$$\sum_{\substack{x_1 \leq x' \leq x_2 \\ y_1 \leq y' \leq y_2}} I(x', y') = S(x_2, y_2) - S(x_1, y_2) - S(x_2, y_1) + S(x_1, y_1). \tag{4.18}$$

Integral images can be used to speed-up the covariance matrix calculation. We can write the (i, j)-th element of the covariance matrix $C(i, j)$ (see Eq. 4.1) as:

$$C(i,j) = \frac{1}{n-1} \sum_{k=1}^{n} (f_k(i) - \mu(i))(f_k(j) - \mu(j))^T, \quad i, j = 1 \dots d, \tag{4.19}$$

where μ is the mean of the feature vectors (see Eq. 4.2), and d is the number of input features of the covariance matrix.

This equation can be rewritten expanding the mean and rearranging the terms:

$$C(i,j) = \frac{1}{n-1} \left[\sum_{k=1}^{n} f_k(i) f_k(j) - \frac{1}{n} \sum_{k=1}^{n} f_k(i) \sum_{k=1}^{n} f_k(j) \right], \quad i, j = 1 \dots d. \tag{4.20}$$

Then, define an integral image P such as:

$$P_{x',y'} = P(x', y') = \sum_{\substack{x \leq x' \\ y \leq y'}} f(x, y), \tag{4.21}$$

where $f(x, y)$ is a vector of size $d \times 1$ with the pixel-level features extracted for a pixel position (x, y) of an image I.

Moreover, define an integral image Q such as:

$$Q_{x',y'} = Q(x', y') = \sum_{\substack{x \leq x' \\ y \leq y'}} f(x, y) f(x, y)^T. \tag{4.22}$$

The integral image Q can be also written as:

$$Q_{x',y'} = Q(x', y', i, j) = \sum_{\substack{x \leq x' \\ y \leq y'}} f(x, y, i) f(x, y, j), \tag{4.23}$$

where $f(x, y, i)$ is the i-th element of the vector $f(x, y)$. The integral image Q is a symmetric matrix, and thus only $\frac{d(d+1)}{2}$ passes over the image are enough to calculate it.

Then, the covariance of a rectangular region R (see Figure 4.7) of an image I can be calculated as:

$$\begin{aligned} C_{x_1,y_1,x_2,y_2} = \frac{1}{m-1}\Big[& Q_{x_2,y_2} - Q_{x_2,y_1-1} - Q_{x_1-1,y_2} + Q_{x_1-1,y_1-1} \\ & -\frac{1}{m}(P_{x_2,y_2} - P_{x_2,y_1-1} - P_{x_1-1,y_2} + P_{x_1-1,y_1-1}) \\ & (P_{x_2,y_2} - P_{x_2,y_1-1} - P_{x_1-1,y_2} + P_{x_1-1,y_1-1})^T\Big], \end{aligned} \tag{4.24}$$

where:

$$m = (x_2 - x_1 + 1)(y_2 - y_1 + 1). \tag{4.25}$$

Therefore, the covariance of any rectangular region can be computed in $O(d^2)$ time.

4.3 Approach Overview

In this section, we present our action recognition framework based on the introduced VCML descriptor.

In the first step of our approach, we extract local spatio-temporal video volumes (see Figure 4.8). In order to do that, we compute the Dense Trajectories in a video sequence; we apply a dense sampling to extract interest points and we track these interest points using a dense optical flow field (see Section 3.2.1.2). Then, we extract local spatio-temporal video volumes around the detected trajectories. By extracting dense trajectories, we provide a good coverage of a video sequence and we ensure extraction of meaningful features. The Dense Trajectories were selected based on their use in the recent literature. However, the VCML descriptor can be used to represent local spatio-temporal video volumes extracted by any other algorithm, *e.g.* by the Spatio-Temporal Interest Points [Laptev 2005] (see Section 3.2.1.1).

Then, in the second step of our approach, we use the proposed Video Covariance Matrix Logarithm descriptor to represent appearance information of local spatio-temporal video volumes. Moreover, we extract the Trajectory shape, Histogram of Oriented Gradients, Histogram of Optical Flow, and Motion Boundary Histogram descriptors for each local spatio-temporal video volume, as these descriptors carry complementary information about the visual appearance and visual motion.

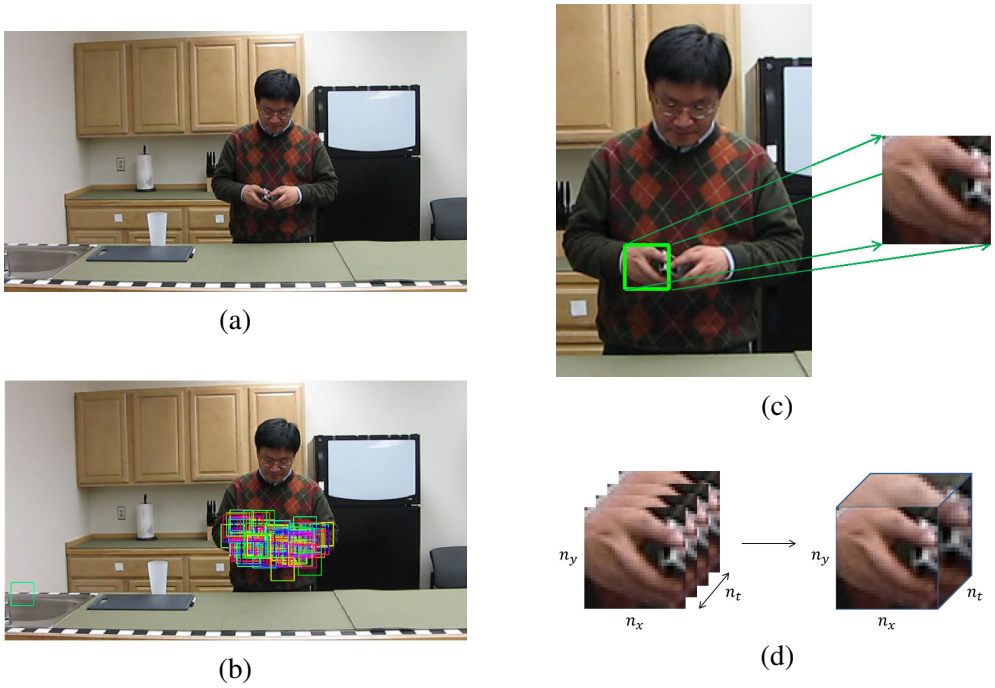


Figure 4.8 – (a) Sample video frame. (b) Sample spatial neighborhoods of the extracted local video volumes. (c) Selected sample neighborhood. (d) Spatial neighbourhoods around the consecutive positions of a trajectory form a spatio-temporal video volume.

Once the descriptors are calculated in a video sequence, we use them to represent this video sequence. We apply the Fisher vector encoding, which was introduced in Section 3.2.2.2. We compute a separate video representation for each descriptor, and we concatenate the calculated Fisher vector based representations into a single feature vector.

Finally, we apply the Support Vector Machines to classify video representations into action categories (see Section 3.2.3). We use the Support Vector Machines with the linear kernel. Linear classifiers have shown to be efficient and have shown to provide good accuracy with high dimensional video representations (see Section 3.2.2.2). For multi-class classification, we use the one-vs-all approach (see Section 3.2.3).

4.4 Experiments

In this section, we present an evaluation, comparison, and analysis of the proposed VCML descriptor and the proposed action recognition approach.

The following experiments are based on the Fisher vector encoding and 6 codebook sizes: $\{16, 32, 64, 128, 256, 512\}$. Moreover, we investigate the influence of using the Principal Component Analysis (PCA) technique with descriptors on action recognition accuracy.

The experiments are performed on 4 state-of-the-art action recognition datasets.

- 2 smaller and less challenging datasets:
  - Weizmann dataset,
  - URADL dataset.

  These datasets contain a smaller number of actions and videos, *e.g.* 10 actions and 150 videos for the URADL dataset.
- 2 bigger and more challenging datasets:
  - MSRDailyActivity3D dataset,
  - HMDB51 dataset.

  These datasets contain a greater number of actions and videos, *e.g.* 51 actions and 6766 videos for the HMDB51 dataset.

The selected datasets vary in the number of actions, types of actions, number of videos, and challenges.

In this section, we use the following abbreviations:

- VCML7 – Video Covariance Matrix Logarithm descriptor using 7 pixel-level features (see Section 4.2.2),

- VCML9 – Video Covariance Matrix Logarithm descriptor using 9 pixel-level features (see Section 4.2.2).
- #PCA – # descriptor with PCA,
- #k@ – # descriptor applied with Fisher vector encoding using the codebook size @,
- DT – Dense Trajectories, *i.e*. Trajectory Shape, HOG, HOF, and MBH descriptors.

The remainder of the section is organized as follows. In Section 4.4.1, we present experiments on the Weizmann dataset. Section 4.4.2 presents experiments on the URADL dataset. In Section 4.4.3, we present experiments on the MSR Daily Activity 3D dataset. Then, in Section 4.4.4, we present experiments on the HMDB51 dataset. Finally, we present the summary and analysis of the results in Section 4.4.5.

4.4.1 Weizmann Dataset

The Weizmann Action Recognition dataset (in short, the Weizmann dataset) is presented in Section 3.3.3.

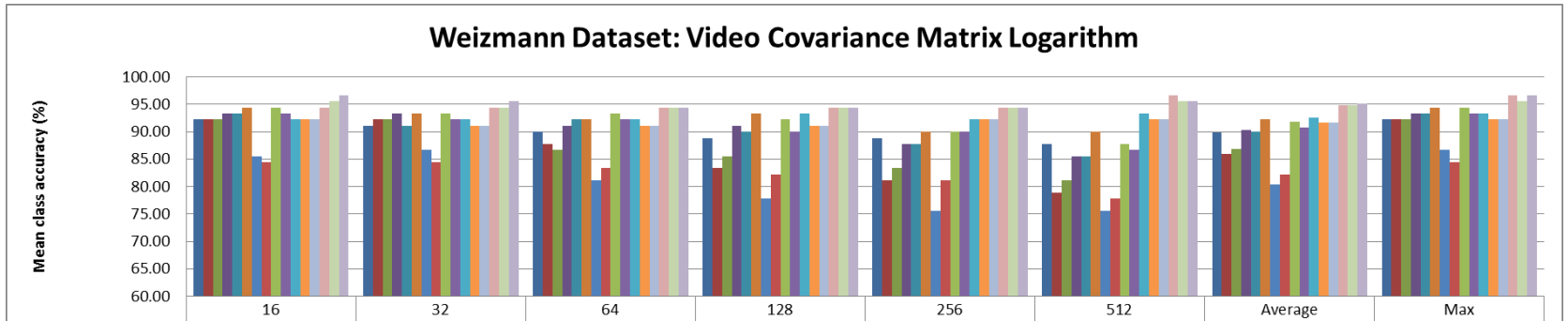
The detail evaluation results are presented in Figure 4.9.

Individually, HOG, VCML7, and VCML9 descriptors achieve 92.22% of mean class accuracy. Then, we fuse the HOG and VCML descriptors, and we obtain the following results: HOG + VCML7 representation achieves 93.33%, and HOG + VCML9 also achieves 93.33%. The fusion HOG + VCML improves action recognition accuracy in comparison to the accuracy of these descriptors alone.

Then, we evaluate the above descriptors with PCA. The HOG with PCA achieves 94.44%, the VCML7 with PCA achieves 86.67%, and the VCML9 with PCA achieves 84.44%. The PCA increases the accuracy for the HOG descriptor, but significantly decreases the accuracy for the VCML descriptors. Then, we fuse the HOG and VCML descriptors, and we obtain the following results: HOGPCA + VCML7 achieves 94.44%, and HOGPCA + VCML9 achieves 93.33%. The fusion HOGPCA + VCML7 achieves the same accuracy as the HOGPCA alone, but the fusion HOGPCA + VCML9 slightly decreases the accuracy in comparison to the accuracy of the HOGPCA alone.

Then, we evaluate the action recognition accuracy of the Dense Trajectories (*i.e*. Trajectory Shape, HOG, HOF, and MBH descriptors fused together). The Dense Trajectories representation achieves 93.33% of mean class accuracy. Moreover, we evaluate the DT with PCA, and we achieve 96.67% of accuracy.

Finally, we fuse the Dense Trajectories representation and the proposed VCML descriptors, *i.e*. we evaluate the accuracy of: DT + VCML7, DT + VCML9, DTPCA + VCML7, and DTPCA + VCML9. The best result, 96.67% of mean class accuracy, is



	16	32	64	128	256	512	Average	Max
HOG	92.22	91.11	90.00	88.89	88.89	87.78	89.81	92.22
VCML7	92.22	92.22	87.78	83.33	81.11	78.89	85.93	92.22
VCML9	92.22	92.22	86.67	85.56	83.33	81.11	86.85	92.22
HOG + VCML7	93.33	93.33	91.11	91.11	87.78	85.56	90.37	93.33
HOG + VCML9	93.33	91.11	92.22	90.00	87.78	85.56	90.00	93.33
HOGPCA	94.44	93.33	92.22	93.33	90.00	90.00	92.22	94.44
VCML7PCA	85.56	86.67	81.11	77.78	75.56	75.56	80.37	86.67
VCML9PCA	84.44	84.44	83.33	82.22	81.11	77.78	82.22	84.44
HOGPCA + VCML7	94.44	93.33	93.33	92.22	90.00	87.78	91.85	94.44
HOGPCA + VCML9	93.33	92.22	92.22	90.00	90.00	86.67	90.74	93.33
DT	92.22	92.22	92.22	93.33	92.22	93.33	92.59	93.33
DT + VCML7	92.22	91.11	91.11	91.11	92.22	92.22	91.67	92.22
DT + VCML9	92.22	91.11	91.11	91.11	92.22	92.22	91.67	92.22
DTPCA	94.44	94.44	94.44	94.44	94.44	96.67	94.81	96.67
DTPCA + VCML7	95.56	94.44	94.44	94.44	94.44	95.56	94.81	95.56
DTPCA + VCML9	96.67	95.56	94.44	94.44	94.44	95.56	95.19	96.67

Figure 4.9 – Weizmann dataset: Evaluation results of Video Covariance Matrix Logarithm descriptors. The plot presents the mean class accuracy of descriptors with respect to the codebook size (the “x” axis).

achieved both by the DTPCA, and by the DTPCA + VCML9.

Although the VCML representation do not increase (and do not decrease) maximum action recognition accuracy across codebooks, which is already very high, on average, the VCML representation improves action recognition accuracy (average accuracy: 95.19% of DTPCA + VCML vs. 94.81% of DTPCA, see Figure 4.9).

4.4.2 URADL Dataset

The University of Rochester Activities of Daily Living dataset (in short, the URADL dataset) is presented in Section 3.3.3.

The detail evaluation results are presented in Figure 4.10.

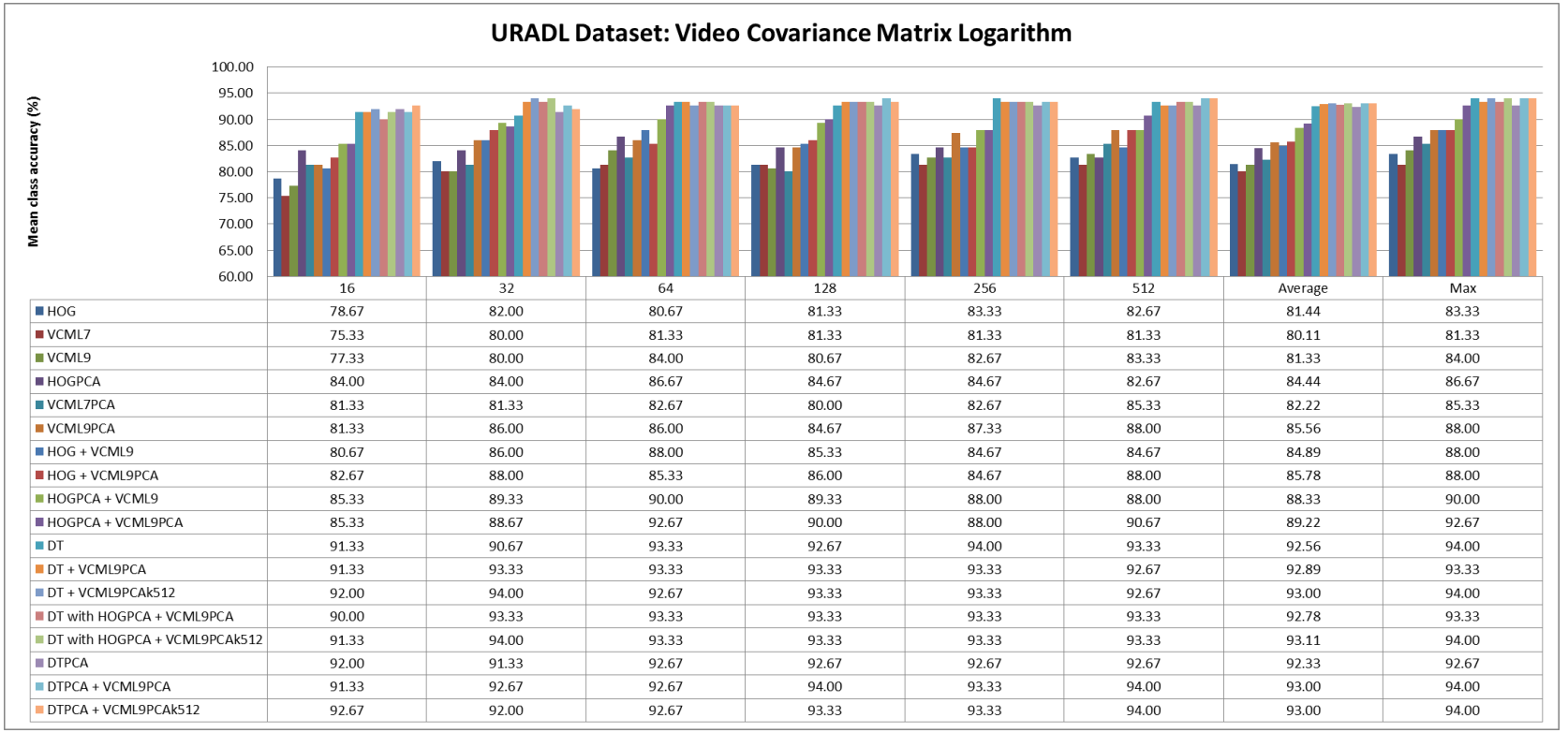
Individually, HOG, VCML7, and VCML9 descriptors achieve 83.33%, 81.33%, and 84% of mean class accuracy, respectively. The VCML9 descriptor outperforms the HOG descriptor.

Then, we evaluate the above descriptors with PCA. The HOG with PCA achieves 86.67%, the VCML7 with PCA achieves 85.33%, and the VCML9 with PCA achieves 88%. The PCA significantly increases the accuracy of each descriptor, and the VCML9 descriptor outperforms the HOG descriptor again.

Then, we fuse the best HOG and VCML descriptors, with and without the use of PCA, and we obtain the following results: HOG + VCML9 representation achieves 88%, HOG + VCML9PCA achieves 88%, HOGPCA + VCML9 achieves 90%, and HOGPCA + VCML9PCA achieves 92.67%. The fusion HOGPCA + VCML9PCA significantly improves action recognition accuracy in comparison to the accuracy of the HOG descriptor (92.67% vs. 83.33%).

Then, we evaluate the action recognition accuracy of the Dense Trajectories. The Dense Trajectories representation achieves 94% without PCA and 92.67% with PCA. Note that the HOGPCA + VCML9PCA representation (2 descriptors) achieves 92.67% of mean class accuracy, the same accuracy as the DT with PCA (using 5 descriptors, *i.e.* Trajectory Shape, HOG, HOF, MBHX, and MBHY), and very close to the accuracy of the DT (also using 5 descriptors).

Finally, we fuse the Dense Trajectories representation and the best VCML descriptors, *i.e.* we evaluate the accuracy of: DT + VCML9PCA, DT + VCML9PCAk512, DT with HOGPCA + VCML9PCA, DT with HOGPCA + VCML9PCA, DTPCA + VCML9PCA, and DTPCA + VCML9PCAk512. The VCML9PCA representation achieves very good results with the codebook size 512 (VCML9PCAk512), and thus the fusion is done using various codebook sizes (see Figure 4.10). The best result, 94% of mean class accuracy, is achieved by several representations, *i.e.* DT, DT + VCML9PCAk512, DTPCA +



	16	32	64	128	256	512	Average	Max
HOG	78.67	82.00	80.67	81.33	83.33	82.67	81.44	83.33
VCML7	75.33	80.00	81.33	81.33	81.33	81.33	80.11	81.33
VCML9	77.33	80.00	84.00	80.67	82.67	83.33	81.33	84.00
HOGPCA	84.00	84.00	86.67	84.67	84.67	82.67	84.44	86.67
VCML7PCA	81.33	81.33	82.67	80.00	82.67	85.33	82.22	85.33
VCML9PCA	81.33	86.00	86.00	84.67	87.33	88.00	85.56	88.00
HOG + VCML9	80.67	86.00	88.00	85.33	84.67	84.67	84.89	88.00
HOG + VCML9PCA	82.67	88.00	85.33	86.00	84.67	88.00	85.78	88.00
HOGPCA + VCML9	85.33	89.33	90.00	89.33	88.00	88.00	88.33	90.00
HOGPCA + VCML9PCA	85.33	88.67	92.67	90.00	88.00	90.67	89.22	92.67
DT	91.33	90.67	93.33	92.67	94.00	93.33	92.56	94.00
DT + VCML9PCA	91.33	93.33	93.33	93.33	93.33	92.67	92.89	93.33
DT + VCML9PCAk512	92.00	94.00	92.67	93.33	93.33	92.67	93.00	94.00
DT with HOGPCA + VCML9PCA	90.00	93.33	93.33	93.33	93.33	93.33	92.78	93.33
DT with HOGPCA + VCML9PCAk512	91.33	94.00	93.33	93.33	93.33	93.33	93.11	94.00
DTPCA	92.00	91.33	92.67	92.67	92.67	92.67	92.33	92.67
DTPCA + VCML9PCA	91.33	92.67	92.67	94.00	93.33	94.00	93.00	94.00
DTPCA + VCML9PCAk512	92.67	92.00	92.67	93.33	93.33	94.00	93.00	94.00

Figure 4.10 – URADL dataset: Evaluation results of Video Covariance Matrix Logarithm descriptors. The plot presents the mean class accuracy of descriptors with respect to the codebook size (the “x” axis).

VCML9PCA, and DTPCA + VCML9PCAk512.

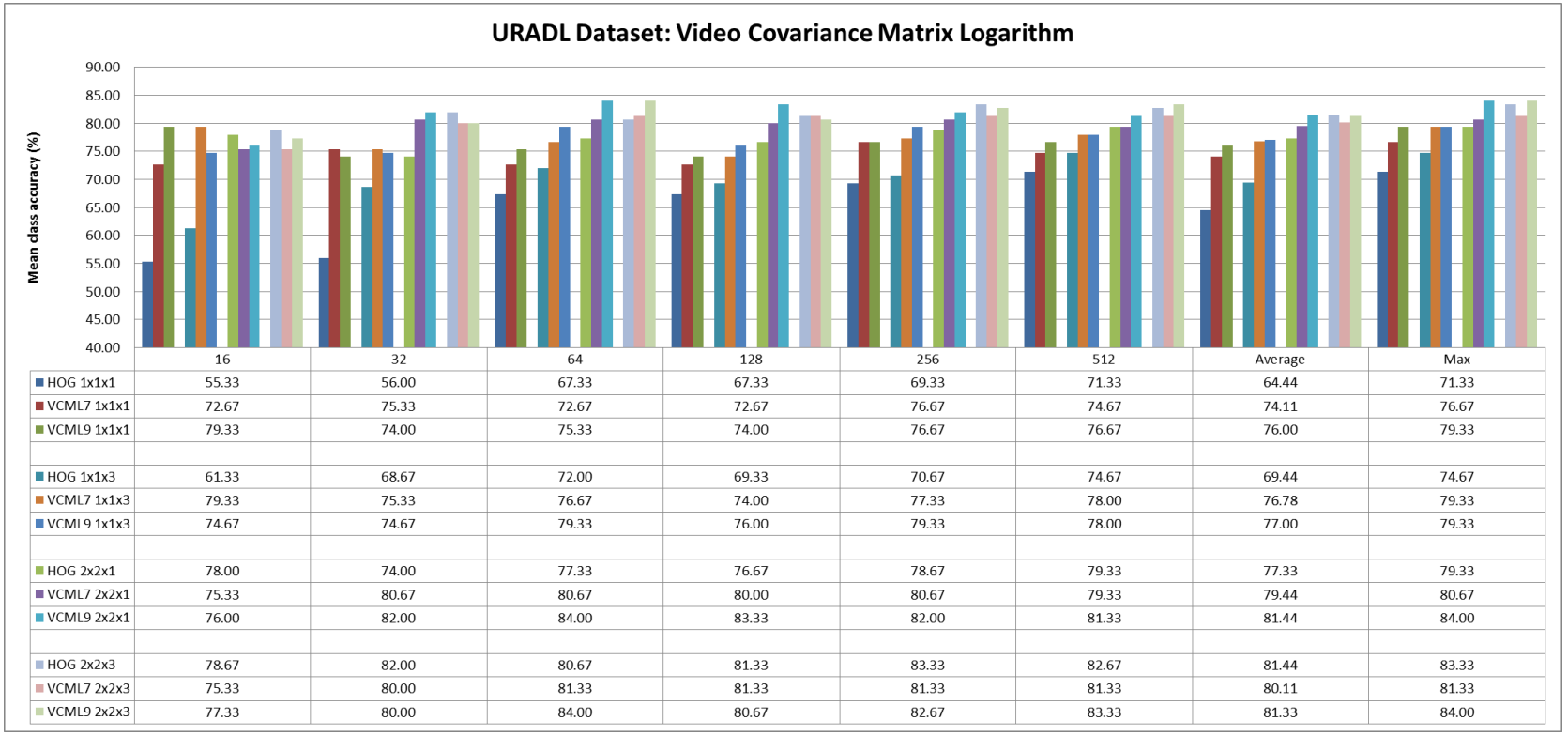
Although the VCML representation do not increase (and do not decrease) maximum action recognition accuracy across codebooks, which is already very high, on average, the VCML representation again improves action recognition accuracy (*e.g.* average accuracy: 93% of DTPCA + VCML9PCA vs. 92.56% of DTPCA, see Figure 4.10).

Moreover, we evaluate HOG, VCML7, and VCML9 descriptors with various spatio-temporal grids:

- Without the use of spatio-temporal grid: HOG, VCML7, and VCML9 achieve 71.33%, 76.67%, and 79.33% of mean class accuracy, respectively.
- Using spatio-temporal grid $1 \times 1 \times 3$: HOG, VCML7, and VCML9 achieve 74.67%, 79.33%, and 79.33%, respectively.
- Using spatio-temporal grid $2 \times 2 \times 1$: HOG, VCML7, and VCML9 achieve 79.33%, 80.67%, and 84%, respectively.
- Using spatio-temporal grid $2 \times 2 \times 3$: HOG, VCML7, and VCML9 achieve 83.33%, 81.33%, and 84%, respectively.

We observe that the spatio-temporal grid significantly improves action recognition accuracy. The spatial grid increases the accuracy more than the temporal grid, but the best results are achieved by the spatio-temporal grid. Moreover, typically, the VCML9 works better than the HOG descriptor, and it works better than the VCML7 descriptor.

The detail evaluation results are presented in Figure 4.11.



	16	32	64	128	256	512	Average	Max
HOG 1x1x1	55.33	56.00	67.33	67.33	69.33	71.33	64.44	71.33
VCML7 1x1x1	72.67	75.33	72.67	72.67	76.67	74.67	74.11	76.67
VCML9 1x1x1	79.33	74.00	75.33	74.00	76.67	76.67	76.00	79.33
HOG 1x1x3	61.33	68.67	72.00	69.33	70.67	74.67	69.44	74.67
VCML7 1x1x3	79.33	75.33	76.67	74.00	77.33	78.00	76.78	79.33
VCML9 1x1x3	74.67	74.67	79.33	76.00	79.33	78.00	77.00	79.33
HOG 2x2x1	78.00	74.00	77.33	76.67	78.67	79.33	77.33	79.33
VCML7 2x2x1	75.33	80.67	80.67	80.00	80.67	79.33	79.44	80.67
VCML9 2x2x1	76.00	82.00	84.00	83.33	82.00	81.33	81.44	84.00
HOG 2x2x3	78.67	82.00	80.67	81.33	83.33	82.67	81.44	83.33
VCML7 2x2x3	75.33	80.00	81.33	81.33	81.33	81.33	80.11	81.33
VCML9 2x2x3	77.33	80.00	84.00	80.67	82.67	83.33	81.33	84.00

Figure 4.11 – URADL dataset: Evaluation results of Video Covariance Matrix Logarithm descriptors. The plot presents the mean class accuracy of descriptors with respect to the codebook size (the “x” axis).

4.4.3 MSR Daily Activity 3D Dataset

The MSR Daily Activity 3D dataset is presented in Section 3.3.4.

The detail evaluation results are presented in Figure 4.12.

Individually, HOG, VCML7, and VCML9 descriptors achieve 60.94%, 56.88%, and 59.38% of mean class accuracy, respectively.

Then, we evaluate the above descriptors with PCA. The HOG with PCA achieves 59.69%, the VCML7 with PCA achieves 55.31%, and the VCML9 with PCA achieves 54.38%. Therefore, the PCA decreases the accuracy of the descriptors on this dataset.

Then, we fuse the best HOG and VCML descriptors, with and without the use of PCA, and we obtain the following results: HOG + VCML9 representation achieves 63.44%, and HOGPCA + VCML9 achieves 63.13%. The fusion of the descriptors improves action recognition accuracy.

Then, we evaluate the action recognition accuracy of the Dense Trajectories, and we obtain 76.25% of mean class accuracy without PCA and 75.31% with PCA.

Finally, we fuse the Dense Trajectories with VCML descriptors, and we achieve 76.25% of DTPCA + VCML9, and 78.13% of DT + VCML9. In both cases, the VCML descriptors improve action recognition accuracy.

The obtained results confirm that the VCML representation improves action recognition accuracy.

4.4.4 HMDB51 Dataset

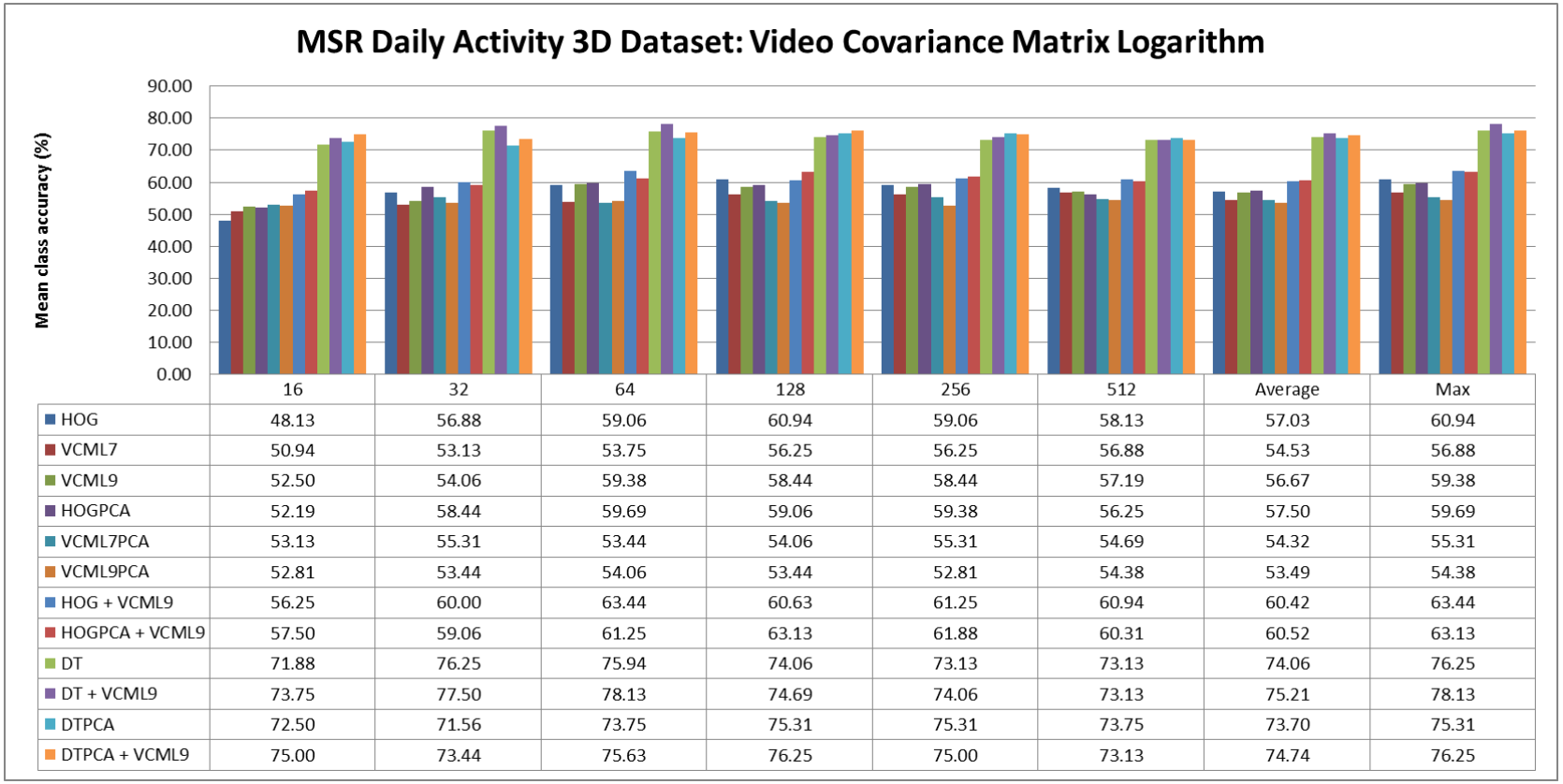
The HMDB: A Large Human Motion Database dataset (in short, the HMDB51 dataset) is presented in Section 3.3.5.

The detail evaluation results are presented in Figure 4.13.

Individually, HOG, VCML7, and VCML9 descriptors achieve 25.64%, 24.68%, and 27.10% of mean class accuracy, respectively.

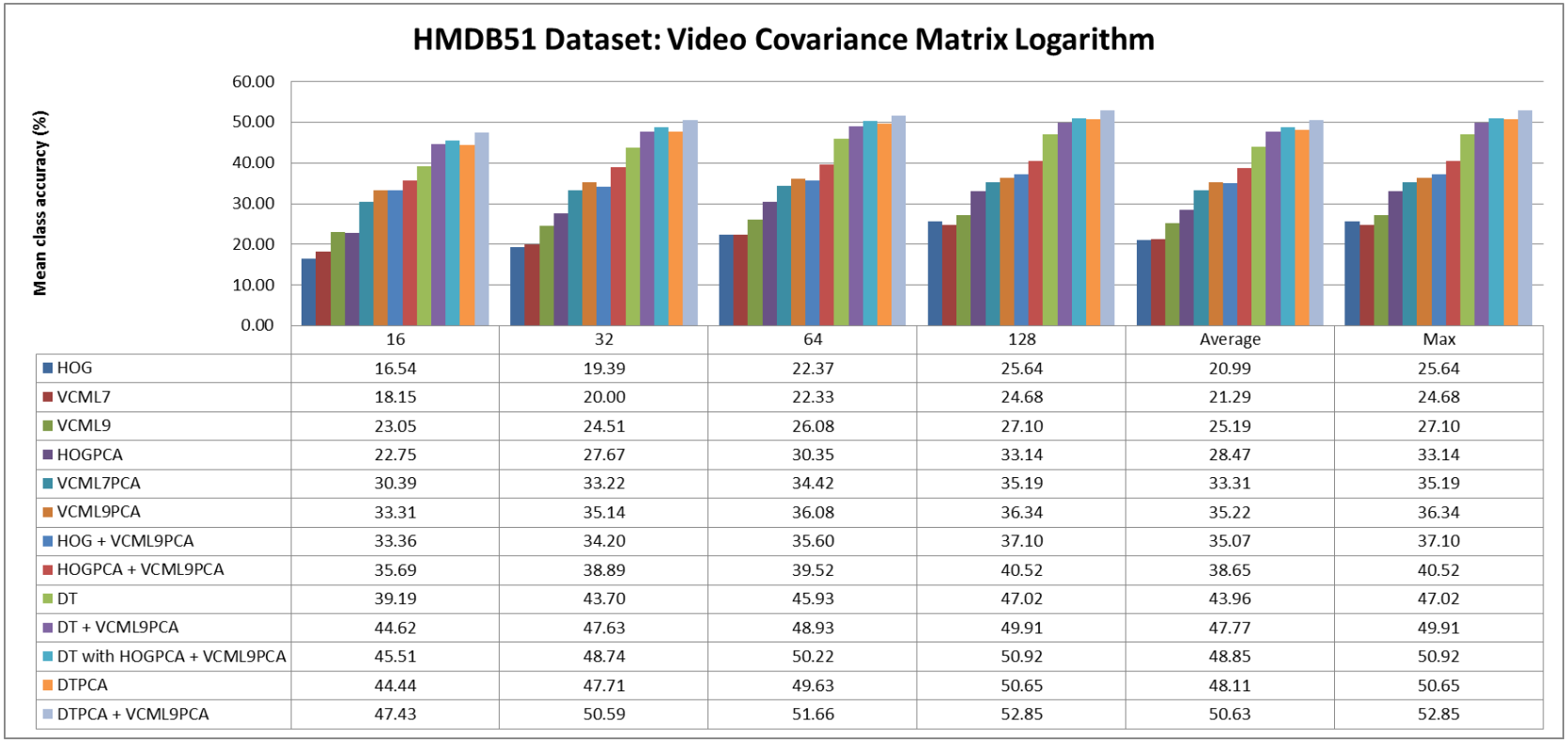
Then, we evaluate the above descriptors with PCA. The HOG with PCA achieves 33.14%, the VCML7 with PCA achieves 35.19%, and the VCML9 with PCA achieves 36.34%. The PCA significantly improves action recognition accuracy of these descriptors.

Then, we fuse the best HOG and VCML descriptors, with and without the use of PCA, and we obtain the following results: HOG + VCML9PCA representation achieves



	16	32	64	128	256	512	Average	Max
HOG	48.13	56.88	59.06	60.94	59.06	58.13	57.03	60.94
VCML7	50.94	53.13	53.75	56.25	56.25	56.88	54.53	56.88
VCML9	52.50	54.06	59.38	58.44	58.44	57.19	56.67	59.38
HOGPCA	52.19	58.44	59.69	59.06	59.38	56.25	57.50	59.69
VCML7PCA	53.13	55.31	53.44	54.06	55.31	54.69	54.32	55.31
VCML9PCA	52.81	53.44	54.06	53.44	52.81	54.38	53.49	54.38
HOG + VCML9	56.25	60.00	63.44	60.63	61.25	60.94	60.42	63.44
HOGPCA + VCML9	57.50	59.06	61.25	63.13	61.88	60.31	60.52	63.13
DT	71.88	76.25	75.94	74.06	73.13	73.13	74.06	76.25
DT + VCML9	73.75	77.50	78.13	74.69	74.06	73.13	75.21	78.13
DTPCA	72.50	71.56	73.75	75.31	75.31	73.75	73.70	75.31
DTPCA + VCML9	75.00	73.44	75.63	76.25	75.00	73.13	74.74	76.25

Figure 4.12 – MSR Daily Activity 3D dataset: Evaluation results of Video Covariance Matrix Logarithm descriptors. The plot presents the mean class accuracy of descriptors with respect to the codebook size (the "x" axis).



	16	32	64	128	Average	Max
HOG	16.54	19.39	22.37	25.64	20.99	25.64
VCML7	18.15	20.00	22.33	24.68	21.29	24.68
VCML9	23.05	24.51	26.08	27.10	25.19	27.10
HOGPCA	22.75	27.67	30.35	33.14	28.47	33.14
VCML7PCA	30.39	33.22	34.42	35.19	33.31	35.19
VCML9PCA	33.31	35.14	36.08	36.34	35.22	36.34
HOG + VCML9PCA	33.36	34.20	35.60	37.10	35.07	37.10
HOGPCA + VCML9PCA	35.69	38.89	39.52	40.52	38.65	40.52
DT	39.19	43.70	45.93	47.02	43.96	47.02
DT + VCML9PCA	44.62	47.63	48.93	49.91	47.77	49.91
DT with HOGPCA + VCML9PCA	45.51	48.74	50.22	50.92	48.85	50.92
DTPCA	44.44	47.71	49.63	50.65	48.11	50.65
DTPCA + VCML9PCA	47.43	50.59	51.66	52.85	50.63	52.85

Figure 4.13 – HMDB51 dataset: Evaluation results of Video Covariance Matrix Logarithm descriptors. The plot presents the mean class accuracy of descriptors with respect to the codebook size (the “x” axis).

37.10%, and HOGPCA + VCML9PCA achieves 40.52%. The fusion of the descriptors improves action recognition accuracy.

Then, we evaluate the action recognition accuracy of the Dense Trajectories and we obtain 47.02% of mean class accuracy. When we fuse the DT with VCML9PCA we achieve 49.91%, and DT with HOGPCA + VCML9PCA achieves 50.92%.

Finally, we evaluate the DT with PCA and we achieve 50.65%. This result is further improved by the fusion of DTPCA and VCML9PCA, which achieves 52.85% of mean class accuracy.

The obtained results confirm that the VCML representation improves action recognition accuracy.

4.4.5 Results Summary and Analysis

Based on the above experimental results we observe that:

- The VCML9 descriptor typically works better than the VCML7 descriptor. This confirms that relations between pixel-level appearance features and positions of these features are informative and useful for action recognition.
- The fusion of HOG and VCML descriptors improves action recognition accuracy. This confirms that HOG and VCML descriptors are complementary to each other, as the former descriptor directly models pixel-level features and the latter descriptor models relations between pixel-level features.
- The fusion of VCML and DT (*i.e*. Trajectory Shape, HOG, HOF, and MBH) representations improves action recognition accuracy in comparison to DT representation alone. The is natural as the VCML and HOG representations capture information about the appearance and the Trajectory Shape, HOF, and MBH representations capture information about the motion.
- The use of the spatio-temporal grid is very important for the accuracy of the VCML representation, and we recommend to use it, as it adds the structural information to the video volume representation.

4.5 Conclusion

We have proposed a new local spatio-temporal descriptor for videos to encode local spatio-temporal video volumes. The new descriptor is called the Video Covariance Matrix Logarithm (VCML). The VCML descriptor is based on a covariance matrix representation, and it models linear relationships between different pixel-level features, such as intensity and gradient.

We have applied the VCML descriptor to encode appearance information of local spatio-temporal video volumes. Using the Fisher vector encoding and Support Vector Machines, we have presented an extensive evaluation of the VCML descriptor on four various action recognition datasets. In comparison with the most popular visual appearance descriptor, *i.e.* the HOG descriptor, the VCML descriptor always achieves superior results. The experiments have shown that the additional accuracy increase can be achieved by the fusion of these two descriptors. This is not surprising as the HOG descriptor and the VCML descriptor are complementary to each other. The former descriptor directly models pixel-level features and the latter descriptor models relations between pixel-level features. Finally, we have presented that the VCML descriptor improves action recognition accuracy in comparison to the state-of-the-art Dense Trajectories.

In future work, we intend to examine the VCML descriptors with motion features, such as optical flow and temporal gradient.

CHAPTER 5

Video Brownian Covariance

Contents

5.1 Introduction 130
5.2 Video Brownian Covariance Descriptor 131
5.2.1 Brownian Covariance 132
5.2.2 Video Frame Descriptor 135
5.2.3 Pixel-Level Features 136
5.2.4 Video Brownian Covariance 137
5.2.5 Normalization 137
5.2.6 Dimension Reduction 138
5.3 Approach Overview 138
5.4 Experiments 139
5.4.1 Descriptor Evaluation 140
5.4.2 Approach Evaluation 144
5.5 Conclusion 149

In this chapter, we propose a new local spatio-temporal descriptor for videos, and we propose a new approach for action recognition based on the introduced descriptor. The new descriptor is called the Video Brownian Covariance (VBC). The VBC descriptor is based on a Brownian covariance, and it models relations between different low-level features, such as intensity and gradient. While the classical covariance can model only linear relationships between features, the Brownian covariance measures all kinds of possible relations between features. We apply the VBC descriptor to encode appearance information of local spatio-temporal video volumes, which are extracted by the Dense Trajectories. Then, we present an extensive evaluation of the proposed VBC descriptor with the Fisher vector encoding and the Support Vector Machines on four various action recognition datasets. We show that the VBC descriptor carries complementary information to the HOG descriptor, as their fusion gives an improvement in action recognition accuracy. Finally, we present that the fusion of the VBC descriptors with the state-of-the-art Dense Trajectories also improves the action recognition accuracy.

5.1 Introduction

In the previous chapter, Chapter 4, we present the motivation behind using descriptors modeling pairwise relations between pixel-level features for action recognition in videos. We also propose a new local spatio-temporal descriptor, the Video Covariance Matrix Logarithm (VCML) descriptor, which is based on the covariance matrix representation. Moreover, we present that the VCML descriptor outperforms the HOG descriptor. An additional increase in action recognition accuracy can be achieved by the fusion of these two descriptors, as they carry complementary information to each other.

The VCML descriptor is based on the covariance matrix representation. The classical covariance measures the strength of the correlation between two variables. However, the covariance can model only linear relationships between variables, and it is not able to measure nonlinear or nonmonotone dependencies, whereas such dependencies may exist and may be useful for action recognition in videos. This indicates some information loss, so covariance based descriptors may not be enough to capture sufficient information in a complex environment such as action recognition.

Brownian covariance can be seen as an extension of the classical covariance, as it measures all kinds of possible relationships between two random variables in arbitrary dimension [Székely 2009]. The Brownian covariance relates to the Brownian motion.

Brownian motion (also known as Brownian movement) is the random motion of microscopic particles suspended in a fluid or a gas. It is the continuous-time stochastic (or probabilistic) process, and it was first observed in 1827 by the Scottish botanist Robert Brown, who noticed a "rapid oscillatory motion" of microscopic particles within pollen grains suspended in water. The Brownian motion term also refers to the mathematical model used to describe random movements of such particles, and a Brownian covariance can be used to express the interactions between these particles.

In Image Processing, a Brownian covariance has been recently proposed as an image descriptor for person Re-identification [Bak 2013]. Driven by recent achievements in the mathematical statistics related to Brownian motion, and promising results of Brownian covariance in Image Processing, we extend the idea of Brownian covariance for images to the spatio-temporal domain of videos. We introduce a new local spatio-temporal descriptor for videos, and we propose a new approach for action recognition based on the introduced descriptor:

- Descriptor: We introduce a new local spatio-temporal descriptor for videos, called the Video Brownian Covariance (VBC). The VBC descriptor is based on a Brownian covariance, and in particular on a sample distance covariance, which measures dependence between two random vectors in arbitrary dimension. The VBC descriptor can be used to represent any low-level features, such as visual appearance and motion features. In order to encode structural information of the video volume, we use

the spatio-temporal grid, and we compute a covariance representation for each cell of the grid.

- Approach: Similarly to the VCML approach, we compute the Dense Trajectories in a video sequence, and we propose to extract local spatio-temporal video volumes around the trajectories. Then, we propose to represent the appearance information of each local spatio-temporal video volume by our VBC descriptor. Moreover, we extract the Trajectory shape, Histogram of Oriented Gradients, Histogram of Optical Flow, and Motion Boundary Histogram descriptors for each local spatio-temporal video volume. Then, we apply the Fisher vector encoding to represent videos. Finally, we fuse the obtained video representations, and we use the Support Vector Machines for action classification.

- Experiments: We present an extensive evaluation of our descriptor and our approach on four various state-of-the-art datasets. We present that the VBC descriptor carries complementary information to the HOG descriptor, as their fusion always gives an improvement in action recognition accuracy. Moreover, we present that the fusion of the VBC descriptors with the state-of-the-art Dense Trajectories also improves the action recognition accuracy.

The remainder of the chapter is organized as follows. In Section 5.2, we propose the Video Brownian Covariance descriptor. Section 5.3 presents our action recognition framework. In Section 5.4, we present experimental results, comparison, and analysis. Finally, we conclude in Section 5.5.

5.2 Video Brownian Covariance Descriptor

In this section, we propose a new descriptor to encode local spatio-temporal video volumes. The new descriptor is called the Video Brownian Covariance (VBC). It is based on the theory in mathematical statistics related to the Brownian motion. The descriptor is based on *distance covariance* statistics that measure the dependence between random vectors in arbitrary dimension.

Similarly to the VCML descriptor, presented in Chapter 4, and the most popular and powerful action recognition local spatio-temporal descriptors, *i.e*. HOG, HOF, and MBH descriptors, we base our descriptor on the representation of individual frames.

Section 5.2.1 presents the theory in mathematical statistics related to the Brownian motion. In Section 5.2.2, we propose a video frame descriptor, and in Section 5.2.3, we present the pixel-level features that we use to compute the video frame descriptor. In Section 5.2.4, we propose a video volume descriptor, which is an extension of the video frame descriptor to the spatio-temporal domain. Then, in Section 5.2.5, we present a normalization technique, which is applied to the extracted descriptors. Finally, in Section 5.2.6, we apply the Principal Component Analysis to reduce the dimensionality of the descriptor.

5.2.1 Brownian Covariance

While the classical covariance can model only linear relationships between variables, a Brownian covariance measures the degree of all kinds of possible relationships between features [Székely 2009].

In this section, we present the theory in mathematical statistics related to the Brownian motion. The mathematical notations and formulas provided here are in accordance with [Székely 2009]. In particular, we present the definition of a Brownian covariance (Section 5.2.1.1), a sample distance covariance (Section 5.2.1.2), and a distance correlation (Section 5.2.1.3).

5.2.1.1 Distance Covariance $\mathcal{V}^2$

Let $X \in \mathbb{R}^p$ and $Y \in \mathbb{R}^q$ be random vectors, where p and q are positive integers. f_X and f_Y denote the characteristic functions of X and Y, respectively, and their joint characteristic function is denoted as $f_{X,Y}$. In terms of characteristic functions, X and Y are independent if and only if $f_{X,Y} = f_X f_Y$. Thus, a natural way of measuring the dependence between X and Y is to find a suitable norm to measure the distance between $f_{X,Y}$ and $f_X f_Y$.

The *distance covariance* $\mathcal{V}^2$ [Székely 2009] is a new measure of dependence between two random vectors X and Y, and it can be defined as:

$$\mathcal{V}^2(X,Y) = ||f_{X,Y}(t,s) - f_X(t)f_Y(s)||^2 \tag{5.1}$$

$$= \frac{1}{c_p c_q} \int_{\mathbb{R}^{p+q}} \frac{|f_{X,Y}(t,s) - f_X(t)f_Y(s)|^2}{|t|_p^{1+p}|s|_q^{1+q}} dtds, \tag{5.2}$$

where c_p and c_q are constants determining norm function in $\mathbb{R}^p \times \mathbb{R}^q$, $t \in X$, $s \in Y$. This measure is analogous to classical covariance, but with the important property that $\mathcal{V}^2(X,Y) = 0$ if and only if X and Y are independent. In [Székely 2009] the *distance covariance* is seen as a natural extension and a generalization of the classical covariance measure; it can be computed between any random vectors in arbitrary dimension, and it has the ability to measure linear association to all types of dependence relations.

5.2.1.2 Sample Distance Covariance $\mathcal{V}_n^2$

We are interested in finding relations between low-level features. Working with images, these features are limited by the amount of pixels. Thus, we use a sample counterpart of a distance covariance. The paper [Székely 2009] provides us the following definition of a sample distance covariance $\mathcal{V}_n^2$.

For a random sample $(\mathbf{X}, \mathbf{Y}) = \{(X_k, Y_k) : k = 1 \dots n\}$ of n i.i.d random vectors (X, Y) from their joint distribution, compute the Euclidean distance matrices $(a_{kl}) = (|X_k - X_l|_p)$ and $(b_{kl}) = (|Y_k - Y_l|_q)$. Define:

$$A_{kl} = a_{kl} - \bar{a}_{k\cdot} - \bar{a}_{\cdot l} + \bar{a}_{\cdot\cdot}, \quad k, l = 1, \dots, n, \tag{5.3}$$

where:

$$\bar{a}_{k\cdot} = \frac{1}{n}\sum_{l=1}^{n} a_{kl}, \quad \bar{a}_{\cdot l} = \frac{1}{n}\sum_{k=1}^{n} a_{kl}, \quad \bar{a}_{\cdot\cdot} = \frac{1}{n^2}\sum_{k,l=1}^{n} a_{kl}. \tag{5.4}$$

Similarly, we define $B_{kl} = b_{kl} - \bar{b}_{k\cdot} - \bar{b}_{\cdot l} + \bar{b}_{\cdot\cdot}$.

The A_{kl} and B_{kl} are simple linear functions of the pairwise distances between n sample elements of X and Y distributions.

Then, the sample distance covariance $\mathcal{V}_n^2$ between two random vector X and Y is defined as:

$$\mathcal{V}_n^2(X, Y) = \frac{1}{n^2}\sum_{k,l=1}^{n} A_{kl} B_{kl}. \tag{5.5}$$

Although the relation of equations (5.1) and (5.5) is not straightforward, THEOREM 2 from [Székely 2009] justifies it:
If $E|X|_p < \infty$ and $E|Y|_q < \infty$, then almost surely

$$\lim_{n\to\infty} \mathcal{V}_n(X, Y) = \mathcal{V}(X, Y). \tag{5.6}$$

5.2.1.3 Distance Correlation $\mathcal{R}_n^2$

A sample distance covariance $\mathcal{V}_n^2$ has its standardized version referred to as *distance correlation* $\mathcal{R}_n^2$, defined as:

$$\mathcal{R}_n^2(X, Y) = \begin{cases} \frac{\mathcal{V}_n^2(X,Y)}{\sqrt{\mathcal{V}_n^2(X)\mathcal{V}_n^2(Y)}}, & \mathcal{V}_n^2(X)\mathcal{V}_n^2(Y) > 0; \\ 0, & \mathcal{V}_n^2(X)\mathcal{V}_n^2(Y) = 0, \end{cases} \tag{5.7}$$

where:

$$\mathcal{V}_n^2(X) = \mathcal{V}_n^2(X, X) = \frac{1}{n^2}\sum_{k,l=1}^{n} A_{kl}^2. \tag{5.8}$$

Figure 5.1 presents the advantage of using the Distance correlation to the correlation [1]

[1] The correlation is the ratio of the covariance of the two variables to the product of their standard deviations.

when nonlinear relationships between variables exist.

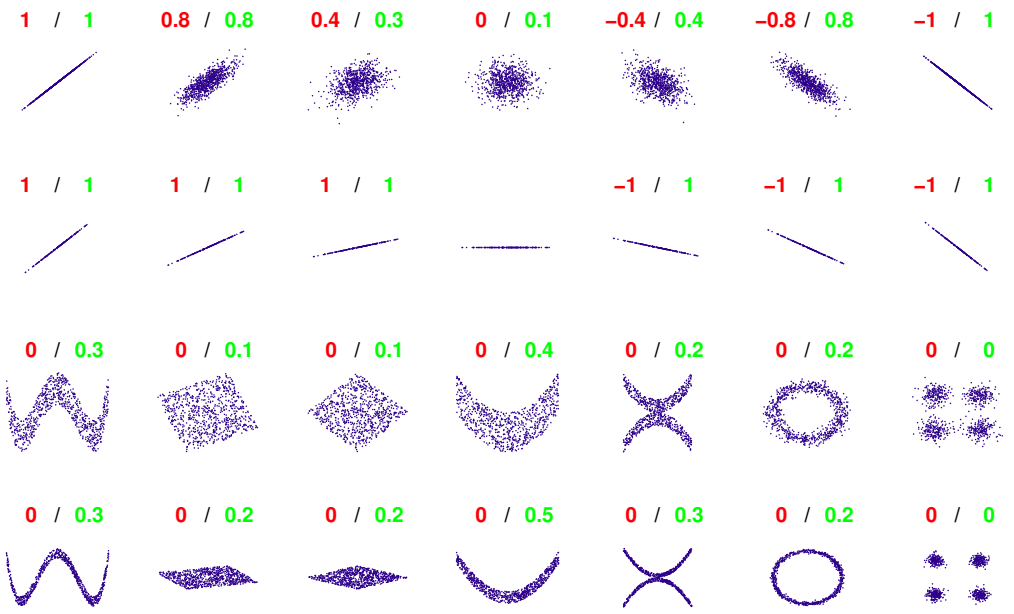


Figure 5.1 – Several sample sets of 2-dimensional points with the correlation (left number, red) and the Distance correlation (right number, green) coefficients calculated between the dimensions of the points for each set. The correlation reflects the noisiness and direction of a linear relationship (first row), but not the slope of that relationship (second row), nor nonlinear relationships (third and fourth rows). The Distance correlation reflects the noisiness of a linear relationship (first row), and nonlinear relationships (third and fourth rows), but not the direction of a linear relationship (first and second rows), nor the slope of that relationship (second row).

5.2.2 Video Frame Descriptor

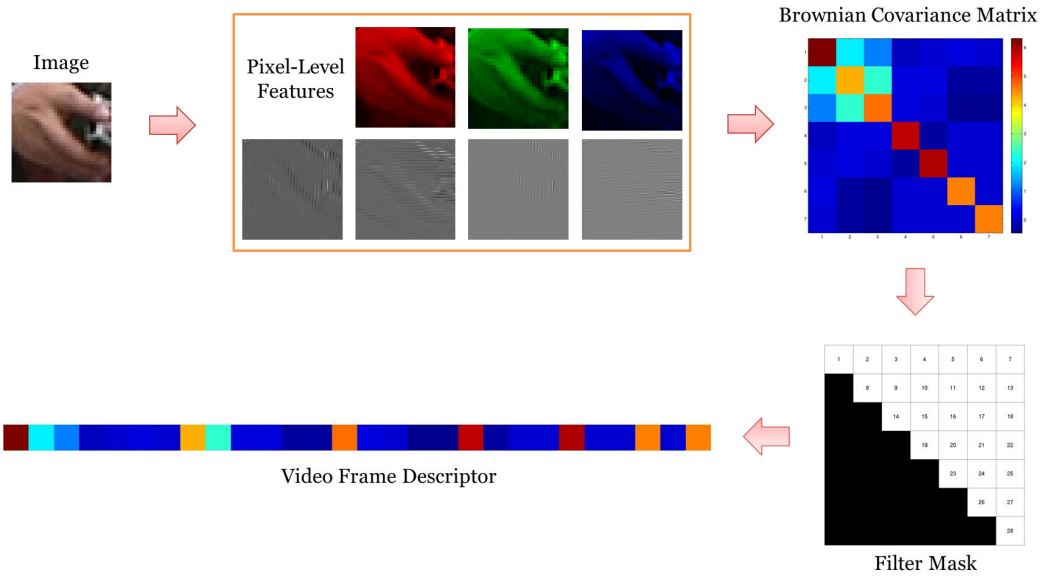


Figure 5.2 – Overview of the calculation process of the video frame descriptor for a sample image. Firstly, we extract pixel-level features of an input image. We represent this image using the Brownian covariance matrix and the extracted pixel-level features. Then, we apply a filter mask extracting all the entries of the upper triangular part of the Brownian covariance matrix. We represent these entries in a form of a vector, called the video frame descriptor.

We are given a single video frame t of spatial size $n_x \times n_y$, and our goal is to create its discriminative and compact representation.

The overview of the calculation process of the proposed video frame descriptor for a sample image is presented in Figure 5.2

Firstly, we calculate low-level (*i.e.* pixel-level) features, *e.g.* intensities in red, green, and blue channels (see Section 5.2.3). For each pixel of a given video frame, we extract d low-level features. Therefore, we represent a video frame t by a set $\{f_{(x,y,t)}\}_{1 \leq x \leq n_x, 1 \leq y \leq n_y}$ of d-dimensional feature vectors ($f_{(x,y,t)} \in \mathcal{R}^d$). Such a frame representation is typically of high dimension ($n_x \times n_y \times d$), and thus it is necessary to transform it into a more compact representation.

In Chapter 4, we use the covariance matrix to create a compact representation of a video frame using the extracted low-level features. The classical covariance measures the strength of the correlation between two variables. However, it can model only linear relationships between variables, and it is not able to measure nonlinear or nonmonotone dependencies, whereas such dependencies may exist and may be useful for action

recognition in videos.

Therefore, we use the Brownian covariance, which measures the degree of all kinds of possible relationships between features [Székely 2009]. The video frame descriptor is based on *distance covariance* statistics that measure the dependence between random vectors in arbitrary dimension.

The calculation of the video frame descriptor is similar to the Video Covariance Matrix Logarithm descriptor, see Chapter 4. The main difference between these descriptors is that we use a Brownian covariance, in particular distance correlation, instead of a classical covariance measure.

5.2.3 Pixel-Level Features

In this section, we present the extraction of low-level, *i.e.* pixel-level, features in a single image. As mentioned before, we focus on the representation of the appearance information.

For fair comparison with the Video Covariance Matrix Logarithm descriptor, we use exactly the same pixel-level features as before. In order to make the chapter self contained, we briefly describe the low-level appearance features.

For every pixel in each frame of the given video volume, we extract seven low-level appearance features. We extract normalized intensities in red, green, and blue channels, and first and second order derivatives of gray scale intensity image along "x" and "y" axes. Thus, every pixel is represented in the following form:

$$f = \left[R,\ G,\ B,\ \frac{\partial I}{\partial x},\ \frac{\partial I}{\partial y},\ \frac{\partial^2 I}{\partial x^2},\ \frac{\partial^2 I}{\partial y^2} \right], \tag{5.9}$$

where R, G, and B are the red, green, and blue intensity channels, and I is the corresponding gray scale intensity image. An example of the extracted seven low-level appearance features is presented in Figure 4.2 in Chapter 4.

The representation based on the above seven low-level features provides a rotation invariant representation of an image. However, the relationships between these low-level features and the spatial positions of these features may be informative and useful for action recognition. Therefore, we also use the extended set of low-level features, where every pixel is represented in the following form:

$$f' = \left[X,\ Y,\ R,\ G,\ B,\ \frac{\partial I}{\partial x},\ \frac{\partial I}{\partial y},\ \frac{\partial^2 I}{\partial x^2},\ \frac{\partial^2 I}{\partial y^2} \right], \tag{5.10}$$

where X and Y represent the spatial position of a pixel in a video frame, and the remaining pixel-level features are presented in Equation 5.9.

5.2.4 Video Brownian Covariance

We are given a spatio-temporal video volume of size $n_x \times n_y \times n_t$, of spatial size $n_x \times n_y$ pixels, and of temporal size n_t video frames. Our goal is to create its discriminative and compact representation.

Firstly, similarly to the VCML descriptor, we use the spatio-temporal grid to encode structural information of the video volume. Thus, we treat an input spatio-temporal video volume as a cuboid and we divide it into a spatio-temporal grid (see Figure 4.4 in Chapter 4), where each cell of the grid is of size $g_x \times g_y \times g_t$, of spatial size $g_x \times g_y$ pixels, and of temporal size g_t video frames.

For each video frame in each cell of the grid, we compute a separate video frame descriptor V_t, as explained in Section 5.2.3. Then, to create a compact cell representation, we describe each cell of the grid as an average of all video frame representations calculated inside this cell:

$$V_{cell} = \frac{1}{g_t} \sum_{t=1}^{g_t} V_t. \tag{5.11}$$

Finally, we define the Video Brownian Covariance (VBC) descriptor D as the concatenation of all the descriptors from all cells of the grid:

$$D = [V_{cell_1}, V_{cell_2}, ..., V_{cell_m}]^T, \tag{5.12}$$

where m is the number of cells of the spatio-temporal grid.

5.2.5 Normalization

Each element of the VBC descriptor has a different meaning and it encodes a different relationship between two features. Therefore, the ranges of the elements of the extracted descriptors may vary. If one of the elements has a broad range of values and another element has a small range of values, the distance between the descriptors may be governed by the former element. The distance between the descriptors is important for action recognition, and thus we apply a data normalization technique to the extracted VBC descriptors to make the elements of the descriptors uniform, so that each element contributes proportionately to the final distance.

We propose to use the unity based normalization. Firstly, we extract the descriptors from the training videos, and we compute a minimum D_i^{min} and a maximum D_i^{max} of each element D_i of the VBC descriptor D. The unity based normalization (also known as feature scaling and rescaling) is a normalization technique, which treats each element of the descriptor separately and scales it to the range $[0, 1]$:

$$D'_i = \frac{D_i - D_i^{min}}{D_i^{max} - D_i^{min}}. \tag{5.13}$$

The importance of using the normalization with the VBC descriptors is presented in Section 5.4.1.

5.2.6 Dimension Reduction

In order to reduce the size of the VBC descriptors, we use the Principal Component Analysis (PCA) [Jolliffe 2002]. The PCA is a feature extraction and dimension reduction technique.

The main idea of the Principal Component Analysis is to reduce the dimensionality of a large number of interrelated variables, while retaining the variation present in the data as much as possible. This is achieved by transforming the data to a new set of uncorrelated variables.

The importance of using the PCA with the VBC descriptors is presented in Section 5.4.1.

5.3 Approach Overview

The VBC action recognition framework is similar to the one proposed in Chapter 4, but instead of the VCML descriptor we use the VBC descriptor.

In the first step of our approach, we extract local spatio-temporal video volumes (see Figure 4.8). In order to do that, we compute the Dense Trajectories in a video sequence; we apply a dense sampling to extract interest points and we track these interest points using a dense optical flow field (see Section 3.2.1.2). Then, we extract local spatio-temporal video volumes around the detected trajectories. By extracting dense trajectories, we provide a good coverage of a video sequence and we ensure extraction of meaningful features. The Dense Trajectories were selected based on their use in the recent literature. However, our VBC descriptor can be used together with any other algorithm extracting local spatio-temporal video volumes, *e.g.* with the Spatio-Temporal Interest Points proposed by Laptev [Laptev 2005] (see Section 3.2.1.1).

Then, in the second step of our approach, we use the proposed Video Brownian Covariance descriptor to represent appearance information of local spatio-temporal video volumes. Moreover, we extract the Trajectory shape, Histogram of Oriented Gradients, Histogram of Optical Flow, and Motion Boundary Histogram descriptors for each local spatio-temporal video volume, as these descriptors carry complementary information about the visual appearance and visual motion.

Once the descriptors are calculated in a video sequence, we use them to represent this video sequence. We apply the Fisher vector encoding, which was introduced in Section 3.2.2.2. We compute a separate video representation for each descriptor, and we

concatenate the calculated Fisher vector based representations into a single feature vector.

Finally, we apply the Support Vector Machines to classify video representations into action categories (see Section 3.2.3). We use the Support Vector Machines with the linear kernel. Linear classifiers have shown to be efficient and have shown to provide good accuracy with high dimensional video representations (see Section 3.2.2.2). For multi-class classification, we use the one-vs-all approach (see Section 3.2.3).

5.4 Experiments

In this section, we present an evaluation, comparison, and analysis of the proposed VBC descriptor and the proposed action recognition approach.

The following experiments are based on the Fisher vector encoding and 6 codebook sizes: $\{16, 32, 64, 128, 256, 512\}$. Moreover, we investigate the influence of using the Principal Component Analysis (PCA) technique with descriptors on action recognition accuracy.

The experiments are performed on 4 state-of-the-art action recognition datasets.

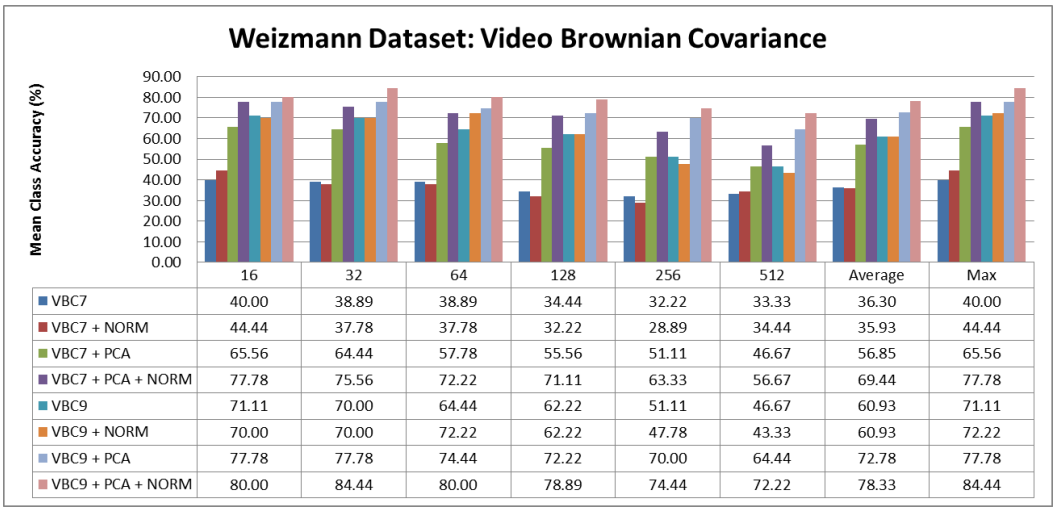
- 2 smaller and less challenging datasets:
  - Weizmann Action Recognition dataset (in short, the Weizmann dataset), which is presented in Section 3.3.3.
  - University of Rochester Activities of Daily Living dataset (in short, the URADL dataset), which is presented in Section 3.3.3.

  These datasets contain a smaller number of actions and videos, *e.g.* 10 actions and 150 videos for the URADL dataset.
- 2 bigger and more challenging datasets:
  - MSR Daily Activity 3D dataset, which is presented in Section 3.3.4.
  - HMDB: A Large Human Motion Database dataset (in short, the HMDB51 dataset), which is presented in Section 3.3.5.

  These datasets contain a greater number of actions and videos, *e.g.* 51 actions and 6766 videos for the HMDB51 dataset.

The selected datasets vary in the number of actions, types of actions, number of videos, and challenges.

The remainder of the section is organized as follows. In Section 5.4.1, we present the evaluation of the descriptor. Then, in Section 5.4.2, we present the evaluation of the approach.



	16	32	64	128	256	512	Average	Max
VBC7	40.00	38.89	38.89	34.44	32.22	33.33	36.30	40.00
VBC7 + NORM	44.44	37.78	37.78	32.22	28.89	34.44	35.93	44.44
VBC7 + PCA	65.56	64.44	57.78	55.56	51.11	46.67	56.85	65.56
VBC7 + PCA + NORM	77.78	75.56	72.22	71.11	63.33	56.67	69.44	77.78
VBC9	71.11	70.00	64.44	62.22	51.11	46.67	60.93	71.11
VBC9 + NORM	70.00	70.00	72.22	62.22	47.78	43.33	60.93	72.22
VBC9 + PCA	77.78	77.78	74.44	72.22	70.00	64.44	72.78	77.78
VBC9 + PCA + NORM	80.00	84.44	80.00	78.89	74.44	72.22	78.33	84.44

Figure 5.3 – Weizmann dataset: Evaluation of Video Brownian Covariance descriptor. The plot presents the mean class accuracy of descriptors with respect to the codebook size.

5.4.1 Descriptor Evaluation

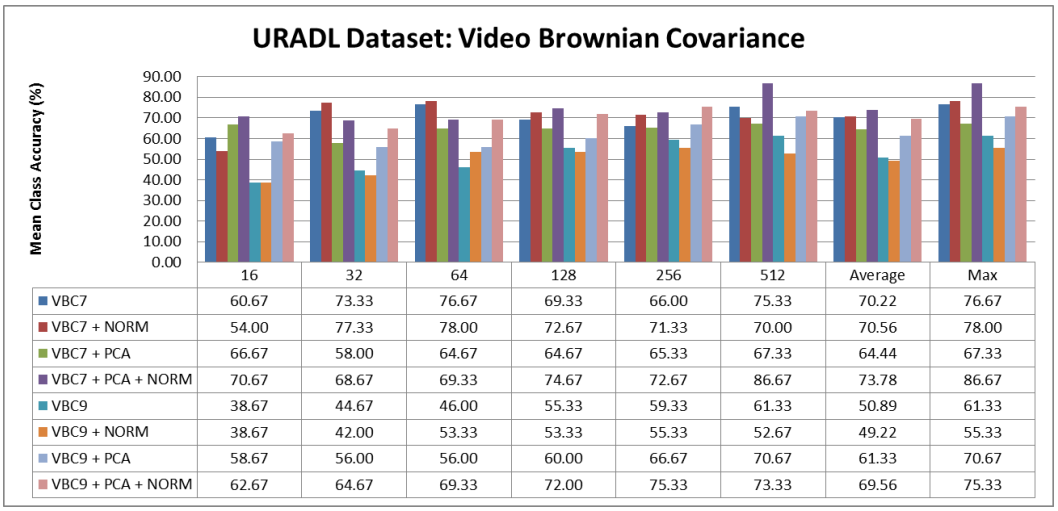
In this section we evaluate the Video Brownian Covariance descriptor using 7 pixel-level features (**VBC7** descriptor) and 9 pixel-level features (**VBC9** descriptor), see Section 5.2.3.

The VBC7 and the VBC9 descriptors are evaluated with and without the use of the normalization (see Section 5.2.5), and with and without the use of the PCA.

The experiments are performed on 4 state-of-the-art action recognition datasets, and the obtained results are as follows.

5.4.1.1 Weizmann Dataset

- The VBC7 descriptor achieves 40% of mean class accuracy without the normalization and 44.44% with the normalization.
- The VBC7 descriptor with PCA achieves 65.56% of mean class accuracy without the normalization and 77.78% with the normalization.
- The VBC9 descriptor achieves 71.11% of mean class accuracy without the normalization and 72.22% with the normalization.
- The VBC9 descriptor with PCA achieves 77.78% of mean class accuracy without the normalization and 84.44% with the normalization.
- **Summary**: Both the normalization and the PCA improve the action recognition accuracy. The VBC9 works better than the VBC7.
- The detail evaluation results are presented in Figure 5.3.

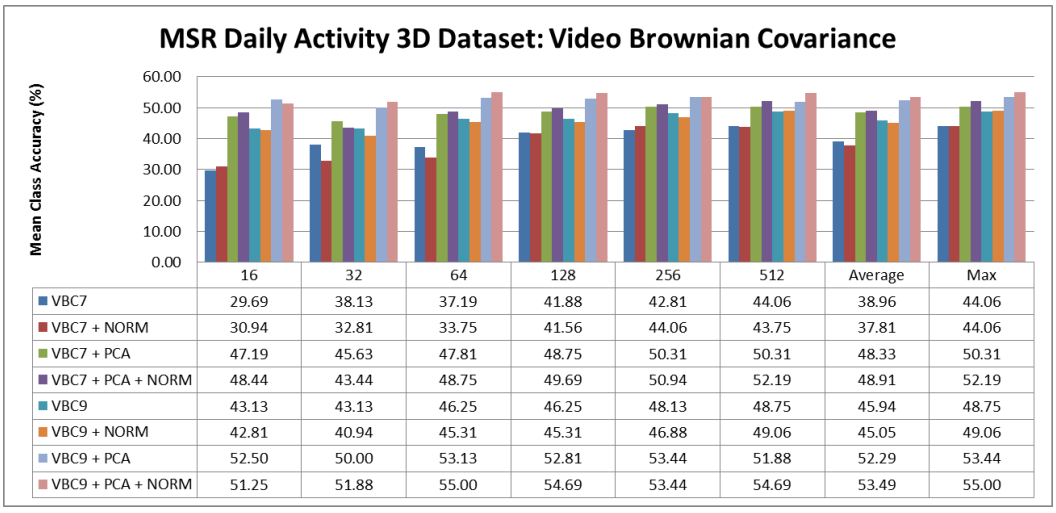


	16	32	64	128	256	512	Average	Max
VBC7	60.67	73.33	76.67	69.33	66.00	75.33	70.22	76.67
VBC7 + NORM	54.00	77.33	78.00	72.67	71.33	70.00	70.56	78.00
VBC7 + PCA	66.67	58.00	64.67	64.67	65.33	67.33	64.44	67.33
VBC7 + PCA + NORM	70.67	68.67	69.33	74.67	72.67	86.67	73.78	86.67
VBC9	38.67	44.67	46.00	55.33	59.33	61.33	50.89	61.33
VBC9 + NORM	38.67	42.00	53.33	53.33	55.33	52.67	49.22	55.33
VBC9 + PCA	58.67	56.00	56.00	60.00	66.67	70.67	61.33	70.67
VBC9 + PCA + NORM	62.67	64.67	69.33	72.00	75.33	73.33	69.56	75.33

Figure 5.4 – URADL dataset: Evaluation of Video Brownian Covariance descriptor. The plot presents the mean class accuracy of descriptors with respect to the codebook size.

5.4.1.2 URADL Dataset

- The VBC7 descriptor achieves 76.67% of mean class accuracy without the normalization and 78% with the normalization.
- The VBC7 descriptor with PCA achieves 67.33% of mean class accuracy without the normalization and 86.67% with the normalization.
- The VBC9 descriptor achieves 61.33% of mean class accuracy without the normalization and 53.33% with the normalization.
- The VBC9 descriptor with PCA achieves 70.67% of mean class accuracy without the normalization and 75.33% with the normalization.
- **Conclusion**: In 3 out of 4 times, the normalization improves the action recognition accuracy, and in 3 out of 4 times the PCA improves the action recognition accuracy. The VBC7 works better than the VBC9.
- The detail evaluation results are presented in Figure 5.4.



	16	32	64	128	256	512	Average	Max
VBC7	29.69	38.13	37.19	41.88	42.81	44.06	38.96	44.06
VBC7 + NORM	30.94	32.81	33.75	41.56	44.06	43.75	37.81	44.06
VBC7 + PCA	47.19	45.63	47.81	48.75	50.31	50.31	48.33	50.31
VBC7 + PCA + NORM	48.44	43.44	48.75	49.69	50.94	52.19	48.91	52.19
VBC9	43.13	43.13	46.25	46.25	48.13	48.75	45.94	48.75
VBC9 + NORM	42.81	40.94	45.31	45.31	46.88	49.06	45.05	49.06
VBC9 + PCA	52.50	50.00	53.13	52.81	53.44	51.88	52.29	53.44
VBC9 + PCA + NORM	51.25	51.88	55.00	54.69	53.44	54.69	53.49	55.00

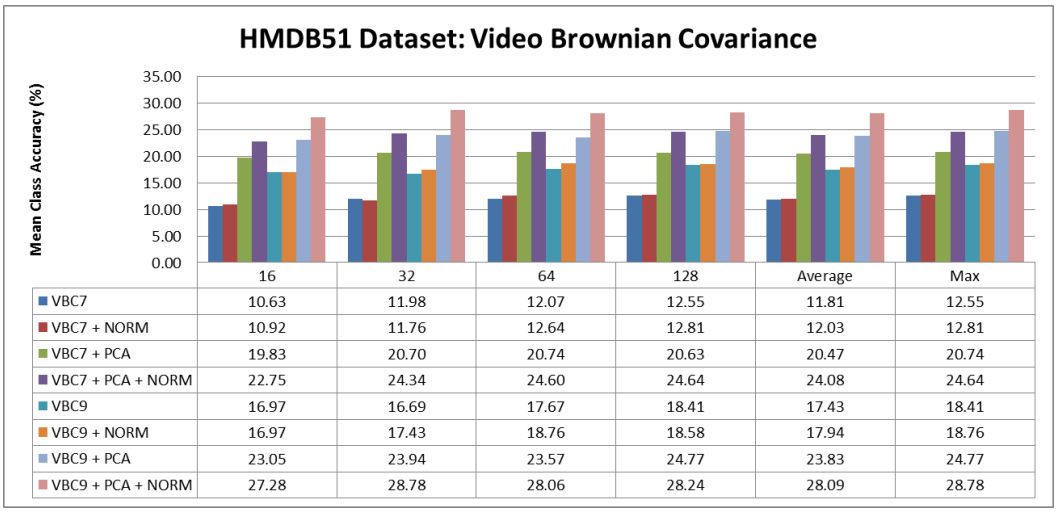
Figure 5.5 – MSR Daily Activity 3D dataset: Evaluation of Video Brownian Covariance descriptor. The plot presents the mean class accuracy of descriptors with respect to the codebook size.

5.4.1.3 MSR Daily Activity 3D Dataset

- The VBC7 descriptor achieves 44.06% of mean class accuracy without the normalization and 44.06% with the normalization.
- The VBC7 descriptor with PCA achieves 50.31% of mean class accuracy without the normalization and 52.19% with the normalization.
- The VBC9 descriptor achieves 48.75% of mean class accuracy without the normalization and 49.06% with the normalization.
- The VBC9 descriptor with PCA achieves 53.44% of mean class accuracy without the normalization and 55% with the normalization.
- **Conclusion**: Both the normalization and the PCA improve the action recognition accuracy. The VBC9 works better than the VBC7.
- The detail evaluation results are presented in Figure 5.5.

HMDB51 Dataset

- The VBC7 descriptor achieves 12.55% of mean class accuracy without the normalization and 12.81% with the normalization.
- The VBC7 descriptor with PCA achieves 20.74% of mean class accuracy without the normalization and 24.64% with the normalization.
- The VBC9 descriptor achieves 18.41% of mean class accuracy without the normalization and 18.76% with the normalization.



	16	32	64	128	Average	Max
VBC7	10.63	11.98	12.07	12.55	11.81	12.55
VBC7 + NORM	10.92	11.76	12.64	12.81	12.03	12.81
VBC7 + PCA	19.83	20.70	20.74	20.63	20.47	20.74
VBC7 + PCA + NORM	22.75	24.34	24.60	24.64	24.08	24.64
VBC9	16.97	16.69	17.67	18.41	17.43	18.41
VBC9 + NORM	16.97	17.43	18.76	18.58	17.94	18.76
VBC9 + PCA	23.05	23.94	23.57	24.77	23.83	24.77
VBC9 + PCA + NORM	27.28	28.78	28.06	28.24	28.09	28.78

Figure 5.6 – HMDB51 dataset: Evaluation of Video Brownian Covariance descriptor. The plot presents the mean class accuracy of descriptors with respect to the codebook size.

- The VBC9 descriptor with PCA achieves 24.77% of mean class accuracy without the normalization and 28.78% with the normalization.
- **Conclusion**: Both the normalization and the PCA improve the action recognition accuracy. The VBC9 works better than the VBC7.
- The detail evaluation results are presented in Figure 5.6.

5.4.1.4 Results summary and analysis

- In 15 out of 16 cases, the accuracy was improved by using the PCA.
- In 15 out of 16 cases, the accuracy was improved by using the normalization.
- For every dataset (4 out of 4), the best result was achieved by using both the normalization and the PCA.

The relationships encoded by the Brownian covariance have very different sample variance (suggested by the above results, and confirmed by the calculations), and without the normalization the distance between the descriptors (used by the Fisher vector encoding) may be governed by the elements with a broad range of values (see Section 5.2.5). Moreover, the PCA may be arbitrary. By applying the data normalization, we make the elements of the descriptors uniform, so that each element contributes proportionately to the final distance, and we also make the PCA less arbitrary.

Therefore, we use both the normalization and the PCA for the Video Brownian Covariance descriptor.

5.4.2 Approach Evaluation

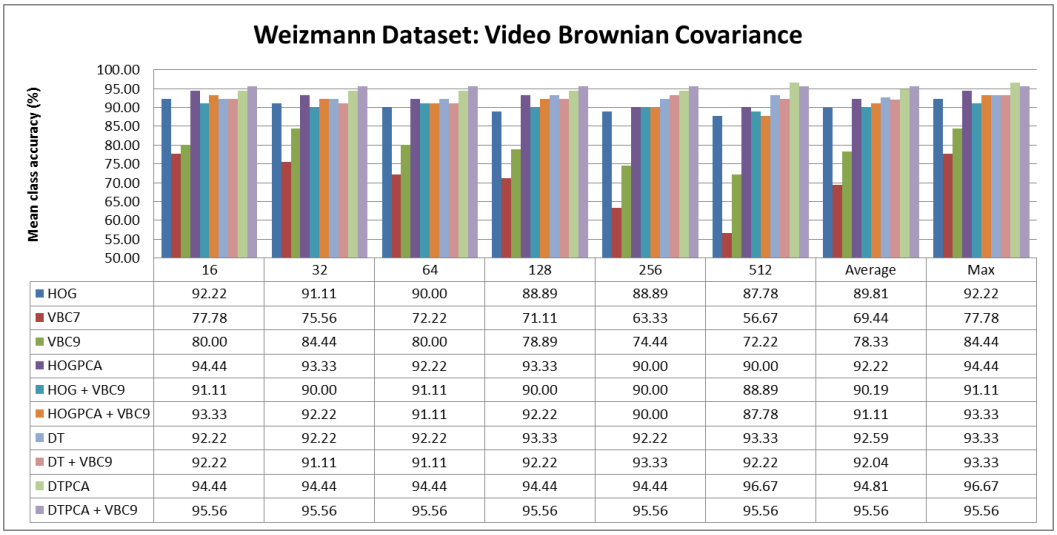
In this section, we evaluate the proposed approach based on the introduced Video Brownian Covariance descriptor using 7 pixel-level features (**VBC7** descriptor) and 9 pixel-level features (**VBC9** descriptor), see Section 5.2.3. The VBC7 and the VBC9 descriptors use the normalization and the PCA (see Section 5.4.1).

We evaluate the accuracy of the HOG, VBC7, and VBC9 descriptors. Moreover, we evaluate the accuracy of the HOG with the PCA (**HOGPCA**). Then, we present the results of the fusion of the HOG/HOGPCA and VBC descriptors. Finally, we fuse the VBC descriptors with the representation of the Dense Trajectories (**DT**, *i.e.* HOG, HOF, MBH, and Trajectory Shape descriptors), and we evaluate the accuracy using the Dense Trajectories without and with the PCA (**DTPCA**).

The experiments are performed on 4 state-of-the-art action recognition datasets, and the obtained results are as follows.

5.4.2.1 Weizmann Dataset

- The state-of-the-art HOG descriptor achieves 92.22% of mean class accuracy.
- The proposed VBC9 descriptor achieves 84.44%, and the VBC7 achieves 77.78%.
- The accuracy of the HOG can be improved by using the PCA, up to 94.44%.
- The fusion HOG + VBC9 improves the average accuracy in comparison to the HOG descriptor (from 89.81% to 90.19%), but slightly decreases the maximum accuracy over codebooks (from 92.22% to 91.11%, *i.e.* roughly 1 video more was miss-classified). The fusion HOGPCA + VBC9 achieves 93.33%.
- The state-of-the-art Dense Trajectories (DT, *i.e.* Trajectory Shape + HOG + HOF + MBH) achieve 93.33%, and the fusion DT + VBC9 achieves 93.33% as well.
- The accuracy of the Dense Trajectories can be improved by using the PCA (DTPCA), up to 96.67%.
- The fusion DTPCA + VBC9 improves the average accuracy in comparison to the DTPCA descriptor (from 94.81% to 95.56%), but slightly decreases the maximum accuracy over codebooks (from 96.67% to 95.56%, *i.e.* roughly 1 video more was miss-classified).
- **Conclusion**: The use of the proposed VBC descriptor with the state-of-the-art HOG / Dense Trajectories improves the action recognition accuracy.
- The detail evaluation results are presented in Figure 5.7.

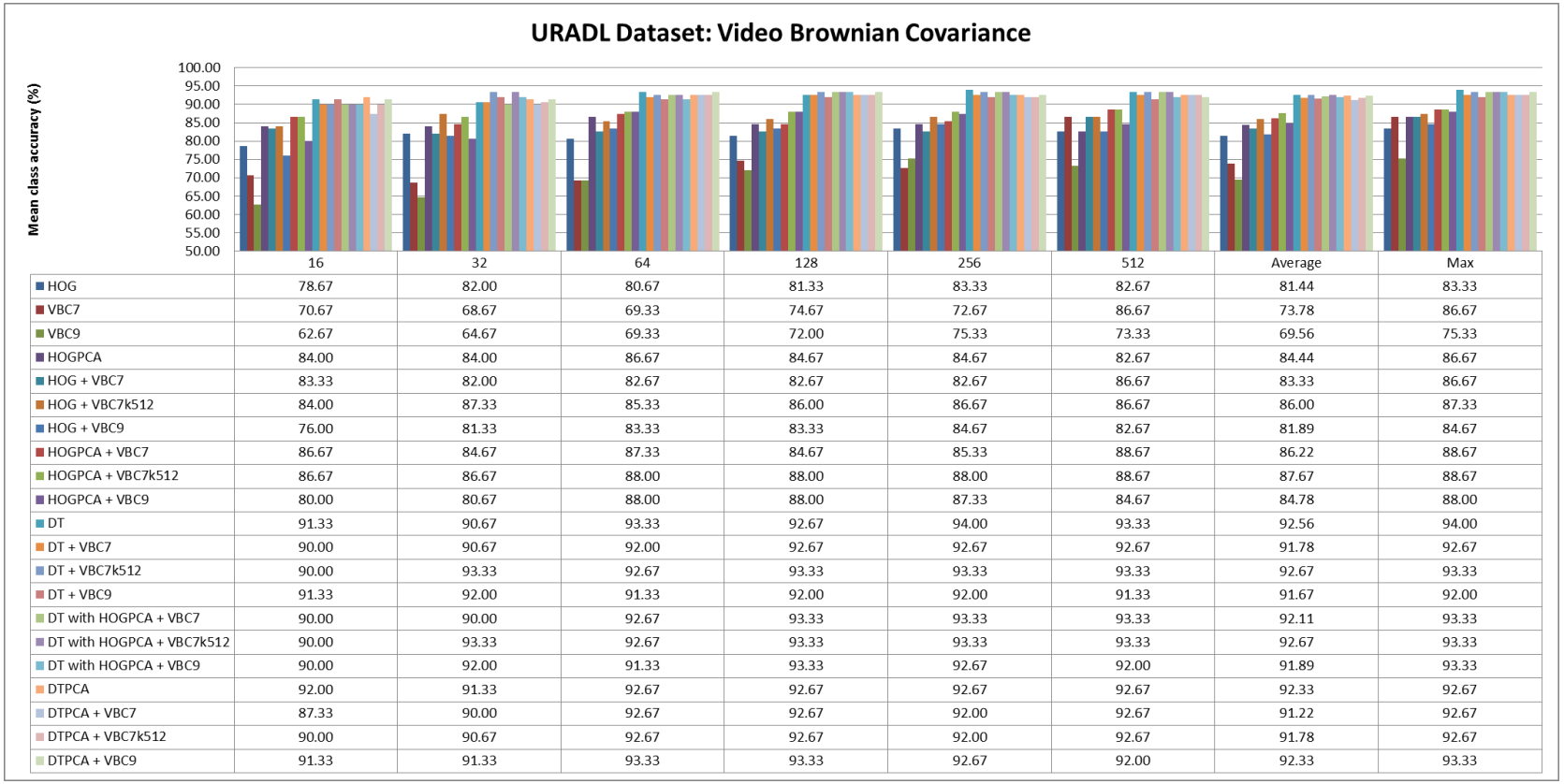


	16	32	64	128	256	512	Average	Max
HOG	92.22	91.11	90.00	88.89	88.89	87.78	89.81	92.22
VBC7	77.78	75.56	72.22	71.11	63.33	56.67	69.44	77.78
VBC9	80.00	84.44	80.00	78.89	74.44	72.22	78.33	84.44
HOGPCA	94.44	93.33	92.22	93.33	90.00	90.00	92.22	94.44
HOG + VBC9	91.11	90.00	91.11	90.00	90.00	88.89	90.19	91.11
HOGPCA + VBC9	93.33	92.22	91.11	92.22	90.00	87.78	91.11	93.33
DT	92.22	92.22	92.22	93.33	92.22	93.33	92.59	93.33
DT + VBC9	92.22	91.11	91.11	92.22	93.33	92.22	92.04	93.33
DTPCA	94.44	94.44	94.44	94.44	94.44	96.67	94.81	96.67
DTPCA + VBC9	95.56	95.56	95.56	95.56	95.56	95.56	95.56	95.56

Figure 5.7 – Weizmann dataset: Evaluation of Video Brownian Covariance approach. The plot presents the mean class accuracy of descriptors with respect to the codebook size.

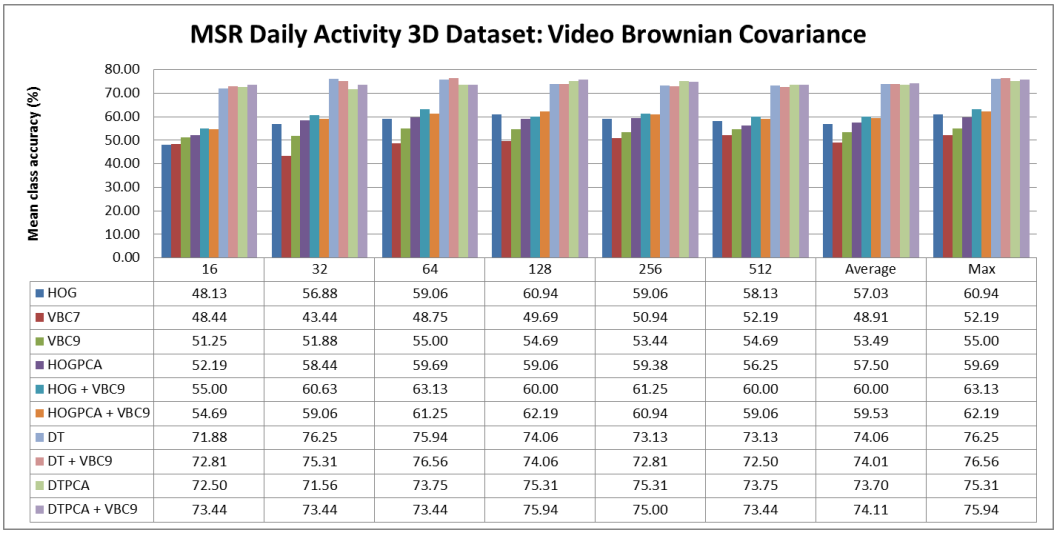
5.4.2.2 URADL Dataset

- The state-of-the-art HOG descriptor achieves 83.33% of mean class accuracy.
- The proposed VBC7 descriptor achieves 86.67%, and the VBC9 achieves 75.33%. The VBC7 descriptor works better than the HOG descriptor.
- The accuracy of the HOG can be improved by using the PCA, up to 86.67%, up to the accuracy of the VBC7 descriptor.
- The fusion HOG + VBC7 improves the action recognition accuracy in comparison to the HOG descriptor, from 83.33% to 86.67%. This result could be further improved up to 87.33% by using the VBC7 descriptor with the codebook size 512 (**VBC7k512**). The VBC7 descriptor works particularly well with the codebook size 512, see Figure 5.8.
- The fusion HOGPCA + VBC7 improves the action recognition accuracy in comparison to the HOGPCA descriptor, from 86.67% to 88.67%. The accuracy is the same using the VBC7k512.
- The state-of-the-art Dense Trajectories (DT) achieve 94%, the fusion DT + VBC7 achieves 92.67%, and the fusion DT + VBC7k512 achieves 93.33%. The fusion DT + VBC7k512 improves the average accuracy over codebooks, from 92.56% to 92.67%.
- The PCA slightly decreases the accuracy of the Dense Trajectories, to as much as 92.67%, and the fusion DTPCA + VBC9 improves the accuracy from 92.67% to 93.33%.



	16	32	64	128	256	512	Average	Max
HOG	78.67	82.00	80.67	81.33	83.33	82.67	81.44	83.33
VBC7	70.67	68.67	69.33	74.67	72.67	86.67	73.78	86.67
VBC9	62.67	64.67	69.33	72.00	75.33	73.33	69.56	75.33
HOGPCA	84.00	84.00	86.67	84.67	84.67	82.67	84.44	86.67
HOG + VBC7	83.33	82.00	82.67	82.67	82.67	86.67	83.33	86.67
HOG + VBC7k512	84.00	87.33	85.33	86.00	86.67	86.67	86.00	87.33
HOG + VBC9	76.00	81.33	83.33	83.33	84.67	82.67	81.89	84.67
HOGPCA + VBC7	86.67	84.67	87.33	84.67	85.33	88.67	86.22	88.67
HOGPCA + VBC7k512	86.67	86.67	88.00	88.00	88.00	88.67	87.67	88.67
HOGPCA + VBC9	80.00	80.67	88.00	88.00	87.33	84.67	84.78	88.00
DT	91.33	90.67	93.33	92.67	94.00	93.33	92.56	94.00
DT + VBC7	90.00	90.67	92.00	92.67	92.67	92.67	91.78	92.67
DT + VBC7k512	90.00	93.33	92.67	93.33	93.33	93.33	92.67	93.33
DT + VBC9	91.33	92.00	91.33	92.00	92.00	91.33	91.67	92.00
DT with HOGPCA + VBC7	90.00	90.00	92.67	93.33	93.33	93.33	92.11	93.33
DT with HOGPCA + VBC7k512	90.00	93.33	92.67	93.33	93.33	93.33	92.67	93.33
DT with HOGPCA + VBC9	90.00	92.00	91.33	93.33	92.67	92.00	91.89	93.33
DTPCA	92.00	91.33	92.67	92.67	92.67	92.67	92.33	92.67
DTPCA + VBC7	87.33	90.00	92.67	92.67	92.00	92.67	91.22	92.67
DTPCA + VBC7k512	90.00	90.67	92.67	92.67	92.00	92.67	91.78	92.67
DTPCA + VBC9	91.33	91.33	93.33	93.33	92.67	92.00	92.33	93.33

Figure 5.8 – URADL dataset: Evaluation of Video Brownian Covariance approach. The plot presents the mean class accuracy of descriptors with respect to the codebook size.



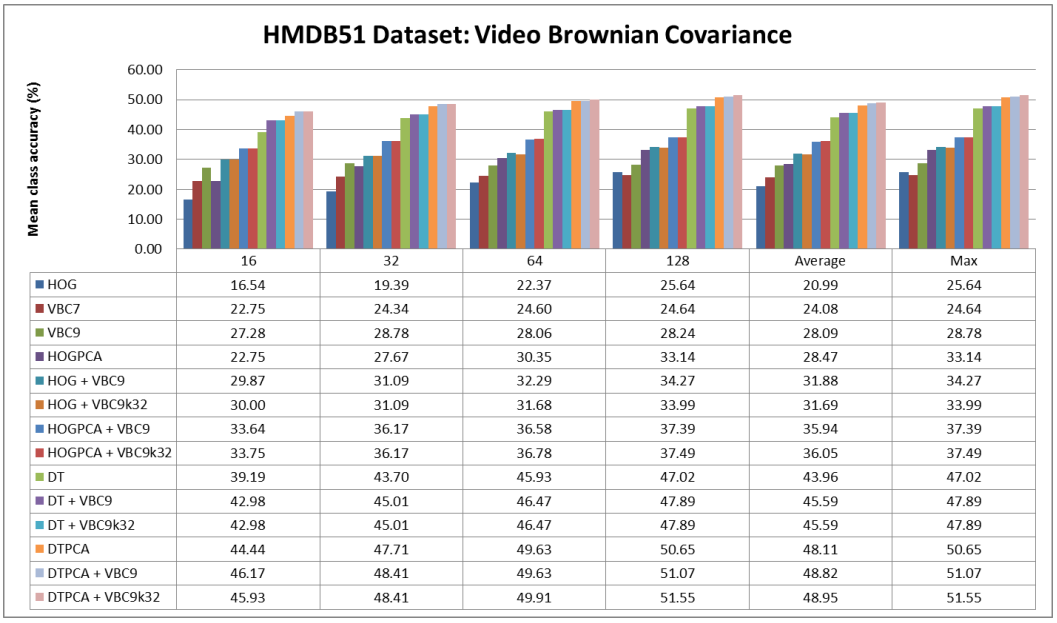
	16	32	64	128	256	512	Average	Max
HOG	48.13	56.88	59.06	60.94	59.06	58.13	57.03	60.94
VBC7	48.44	43.44	48.75	49.69	50.94	52.19	48.91	52.19
VBC9	51.25	51.88	55.00	54.69	53.44	54.69	53.49	55.00
HOGPCA	52.19	58.44	59.69	59.06	59.38	56.25	57.50	59.69
HOG + VBC9	55.00	60.63	63.13	60.00	61.25	60.00	60.00	63.13
HOGPCA + VBC9	54.69	59.06	61.25	62.19	60.94	59.06	59.53	62.19
DT	71.88	76.25	75.94	74.06	73.13	73.13	74.06	76.25
DT + VBC9	72.81	75.31	76.56	74.06	72.81	72.50	74.01	76.56
DTPCA	72.50	71.56	73.75	75.31	75.31	73.75	73.70	75.31
DTPCA + VBC9	73.44	73.44	73.44	75.94	75.00	73.44	74.11	75.94

Figure 5.9 – MSR Daily Activity 3D dataset: Evaluation of Video Brownian Covariance approach. The plot presents the mean class accuracy of descriptors with respect to the codebook size.

- **Conclusion**: The use of the proposed VBC descriptor with the state-of-the-art HOG / Dense Trajectories improves the action recognition accuracy.
- The very detail evaluation results are presented in Figure 5.8.

5.4.2.3 MSR Daily Activity 3D Dataset

- The state-of-the-art HOG descriptor achieves 60.94% of mean class accuracy.
- The proposed VBC9 descriptor achieves 55%, and the VBC7 achieves 52.19%.
- The PCA decreases the accuracy of the HOG descriptor on this dataset, to as much as 59.69%.
- The fusion HOG + VBC9 improves the action recognition accuracy in comparison to the HOG descriptor, from 60.94% to 63.13%.
- The fusion HOGPCA + VBC9 improves the action recognition accuracy in comparison to the HOGPCA descriptor, from 59.69% to 62.19%.
- The state-of-the-art Dense Trajectories (DT) achieve 76.25% without the PCA, and 75.31% with the PCA (DTPCA).
- The fusion DT + VBC9 improves the accuracy from 76.25% to 76.56%, and the fusion DTPCA + VBC9 improves the accuracy from 75.31% to 75.94%.
- **Conclusion**: The use of the proposed VBC descriptor with the state-of-the-art HOG / Dense Trajectories improves the action recognition accuracy.



	16	32	64	128	Average	Max
HOG	16.54	19.39	22.37	25.64	20.99	25.64
VBC7	22.75	24.34	24.60	24.64	24.08	24.64
VBC9	27.28	28.78	28.06	28.24	28.09	28.78
HOGPCA	22.75	27.67	30.35	33.14	28.47	33.14
HOG + VBC9	29.87	31.09	32.29	34.27	31.88	34.27
HOG + VBC9k32	30.00	31.09	31.68	33.99	31.69	33.99
HOGPCA + VBC9	33.64	36.17	36.58	37.39	35.94	37.39
HOGPCA + VBC9k32	33.75	36.17	36.78	37.49	36.05	37.49
DT	39.19	43.70	45.93	47.02	43.96	47.02
DT + VBC9	42.98	45.01	46.47	47.89	45.59	47.89
DT + VBC9k32	42.98	45.01	46.47	47.89	45.59	47.89
DTPCA	44.44	47.71	49.63	50.65	48.11	50.65
DTPCA + VBC9	46.17	48.41	49.63	51.07	48.82	51.07
DTPCA + VBC9k32	45.93	48.41	49.91	51.55	48.95	51.55

Figure 5.10 – HMDB51 dataset: Evaluation of Video Brownian Covariance approach. The plot presents the mean class accuracy of descriptors with respect to the codebook size.

- The detail evaluation results are presented in Figure 5.9.

5.4.2.4 HMDB51 Dataset

- The state-of-the-art HOG descriptor achieves 25.64% of mean class accuracy.
- The proposed VBC9 descriptor achieves 28.78%, and the VBC7 achieves 24.64%. The VBC9 descriptor works better than the HOG descriptor.
- The accuracy of the HOG can be improved by using the PCA, up to 33.14%.
- The fusion HOG + VBC9 significantly improves the action recognition accuracy in comparison to the HOG descriptor, **from** 25.64% **to** 34.27%.
- The fusion HOGPCA + VBC9 significantly improves the action recognition accuracy in comparison to the HOGPCA descriptor, **from** 33.14% **to** 37.39%. This result could be further improved up to 37.49% by using the VBC9 descriptor with the codebook size 32 (**VBC9k32**). The VBC9 descriptor works particularly well with the codebook size 32, see Figure 5.10.
- The state-of-the-art Dense Trajectories (DT, *i.e.* Trajectory Shape + HOG + HOF + MBH) achieve 47.02%, and the fusion DT + VBC9 improves the accuracy up to 47.89%.
- The accuracy of the Dense Trajectories can be improved by using the PCA (DTPCA), up to 50.65%.

- The fusion DTPCA + VBC9 improves the action recognition accuracy up to 51.07%. This result could be further improved up to 51.55% by using the VBC9 with the codebook size 32.
- **Conclusion**: The use of the proposed VBC descriptor with the state-of-the-art HOG / Dense Trajectories improves the action recognition accuracy.
- The detail evaluation results are presented in Figure 5.10.

5.4.2.5 Results Summary and Analysis

Based on the above experimental results we observe that:

- The fusion of the HOG and the VBC descriptors typically improves action recognition accuracy. This confirms that the HOG and the VBC descriptors are complementary to each other, as the former descriptor directly models pixel-level features and the latter descriptor models relations between pixel-level features.
- The use of the proposed VBC descriptor with the state-of-the-art Dense Trajectories (*i.e*. Trajectory Shape + HOG + HOF + MBH) also improves the action recognition accuracy. The is natural as the VBC and HOG representations capture appearance information, and the Trajectory Shape, HOF, and MBH representations capture motion information.

5.5 Conclusion

We have proposed a new local spatio-temporal descriptor for videos to encode local spatio-temporal video volumes. The new descriptor is called the Video Brownian Covariance (VBC). The VBC descriptor is based on a Brownian covariance, and it measures all kinds of possible relationships between low-level features.

We have applied the VBC descriptor to encode appearance information of local spatio-temporal video volumes. Using the Fisher vector encoding and the Support Vector Machines, we have presented an extensive evaluation of the VBC descriptor on four various action recognition datasets. The experiments have shown that the fusion of the HOG and the VBC descriptors typically improves the action recognition accuracy. This is not surprising as the HOG and the VBC descriptors are complementary to each other. The former descriptor directly models low-level features and the latter descriptor models the relations between low-level features. Finally, we have presented that the fusion of the VBC descriptors with the state-of-the-art Dense Trajectories (*i.e*. Trajectory Shape + HOG + HOF + MBH descriptors) also improves the action recognition accuracy. The is also unsurprising as the HOG and the VBC representations capture appearance information, and the Trajectory Shape, HOF, and MBH representations capture motion information.

In future work, we intend to examine the VBC descriptors with motion features, such as optical flow and temporal gradient.

CHAPTER 6

Relative Trajectories

Contents

6.1 Introduction 152
6.2 Relative Trajectories 154
6.2.1 Trajectory Extraction 154
6.2.2 Trajectory Shape and Relative Trajectory Shape Descriptors 155
6.2.3 Dynamic Coordinate System 157
6.2.4 Trajectory Filtering 159
6.3 Approach Overview 161
6.4 Experiments 162
6.4.1 URADL Dataset 163
6.4.2 MSR Daily Activity 3D Dataset 165
6.4.3 KTH Dataset 169
6.4.4 CHU Nice Hospital Dataset 170
6.4.5 Results Summary and Analysis 171
6.5 Conclusion 171

In this chapter, we introduce relative trajectories for action recognition in videos. Our main idea is that action recognition ought to be performed using a dynamic coordinate system corresponding to an object of interest. Therefore, we introduce the Relative Trajectory Shape (RTS) descriptor based on relative positions of a trajectory according to the central point of our dynamic coordinate system. As the center of our dynamic coordinate system, we choose the head position, providing description invariant to camera viewpoint changes. We apply the RTS descriptor to encode local dense trajectories. Then, we present an extensive evaluation of the proposed RTS descriptor on four various action recognition datasets. We show that the RTS representation outperforms the Trajectory Shape (TS) representation, the fusion of the TS and the RTS improves action recognition accuracy in comparison to the TS representation alone, and the fusion of the RTS representation and the Dense Trajectories representation improves the action recognition accuracy in comparison to the Dense Trajectories representation alone.

6.1 Introduction

In the last two chapters we propose local spatio-temporal descriptors to model pairwise relations between pixel-level features:

- In Chapter 4 we propose the Video Covariance Matrix Logarithm (VCML) descriptor, which is based on a covariance matrix representation.
- In Chapter 5 we propose the Video Brownian Covariance (VBC) descriptor, which is based on a Brownian covariance.

The above descriptors were applied to encode appearance information of local spatio-temporal video volumes, and these video volumes were extracted as local neighborhoods surrounding detected local dense trajectories.

In this chapter, instead of modeling neighborhoods surrounding detected local trajectories, we focus on trajectories and their shape.

The shape of trajectories has shown to be informative and has shown to outperform the HOG, HOF, and even MBH descriptors in some scenarios, *e.g*. the Trajectory shape descriptor (see Section 3.2.1.2) has shown to outperform the HOG and HOF descriptors on the Weizmann dataset, and it has shown to outperform the HOG and MBH descriptors on the MSR Daily Activity 3D dataset. More information is presented in Section 3.4.2.

Moreover, the shape of a trajectory can be represented in a form of a short feature vector, of much smaller size than a feature vector representation of the HOG, HOF, and MBH descriptors. The size of the Trajectory shape descriptor is 30, the size of the HOG descriptor is 96, the size of the HOF descriptor is 108, and the size of the MBH descriptor for the horizontal (vertical) component is 96, using the default parameters of the Dense Trajectories (see Section 3.2.1.2).

Local spatio-temporal descriptors, such as the Trajectory shape descriptor, are typically applied with a local feature encoding technique which represents a video sequence using the extracted descriptors.

In Chapter 2 we present the most popular local feature encoding techniques. Among many, the most popular ones are the bag-of-features and the Fisher vector encoding.

Although the existing local feature encoding techniques are very successful in many domains, including Computer Vision, they also contain limitations. One of the main limitations of these models is that they simplify the structure of spatio-temporal video data assuming conditional independence across spatial and temporal domains. They compute only global statistics of local features, ignoring information about the spatio-temporal positions of features. Thus, not using all the available information, they may fail to distinguish similar actions.

In order to overcome this limitation, we propose a new action recognition approach based on local features and inspired by the holistic based methods.

The holistic methods are based on the extraction of information on people localization in videos, and they use a global representation of human body structure, shape and movements for action recognition. A brief literature overview of holistic based methods is presented in Section 2.3.2 of Chapter 2.

Differently from the existing techniques, we propose a new approach based on local features and inspired by the holistic based methods. We introduce a new trajectory descriptor and we propose a new approach for action recognition based on the introduced descriptor:

- Descriptor: We introduce the idea of relative dense trajectories for action recognition in videos. Our main idea is that action recognition ought to be performed using a dynamic coordinate system corresponding to an object of interest. We propose a new feature representation based on dense trajectories and a dynamic coordinate system. We introduce the Relative Trajectory Shape (RTS) descriptor based on relative positions of a trajectory according to the central point of our dynamic coordinate system. As the center of the dynamic coordinate system, we choose the head position, which provides description invariant to camera viewpoint changes. The proposed RTS descriptor introduces spatial information to the local feature encoding technique, and thus enhances the discriminative properties of the action recognition approach.

- Approach: We compute the Dense Trajectories in a video sequence. Then, we propose to represent each trajectory by the Trajectory Shape and the Relative Trajectory Shape descriptors. Moreover, we extract local spatio-temporal video volumes around the trajectories, and we represent each of them by the Histogram of Oriented Gradients, Histogram of Optical Flow, and Motion Boundary Histogram descriptors. Then, we apply the Fisher vector encoding to represent videos. Finally, we fuse the obtained video representations, and we use the Support Vector Machines for action classification.

- Experiments: We present an extensive evaluation of our approach and our descriptor on four various state-of-the-art datasets. We show that the RTS descriptor achieves better results than the Trajectory Shape descriptor. Moreover, we present that the RTS descriptor carries complementary information to the Trajectory Shape descriptor, as their fusion always gives a significant improvement in action recognition accuracy. Finally, we show that the RTS descriptor improves the action recognition in comparison to the state-of-the-art Dense Trajectories.

The remainder of the chapter is organized as follows. In Section 6.2, we introduce the idea of relative trajectories for action recognition in videos and we propose the Relative Trajectory Shape (RTS) descriptor. Section 6.3 presents our action recognition framework.

In Section 6.4, we present experimental results, comparison, and analysis. Finally, we conclude in Section 6.5.

6.2 Relative Trajectories

In this section, we introduce the idea of relative dense trajectories for action recognition in videos. We focus on local trajectories (Section 6.2.1), which in a natural way describe moving objects in a video sequence. Our main idea is that action recognition ought to be performed using a dynamic coordinate system corresponding to an object of interest. In Section 6.2.2, we propose a new feature representation based on dense trajectories and a dynamic coordinate system. We introduce the Relative Trajectory Shape (RTS) descriptor based on relative positions of a trajectory according to the central point of our dynamic coordinate system. As the center of our dynamic coordinate system, we choose the head position, providing description invariant to camera viewpoint changes (see Section 6.2.3). The proposed object-centric local feature representation introduces spatial information to the local feature encoding technique, therefore enhancing the discriminative properties of the action recognition approach. It helps to distinguish similar features detected at different locations, *e.g.* to distinguish similar features appearing on hands and feet. Moreover, we propose to filter noisy and background trajectories using the results of the head estimation framework (Section 6.2.4).

6.2.1 Trajectory Extraction

Over the last years, many techniques have been proposed for the extraction of local trajectories in a video sequence. A brief literature overview of the popular state-of-the-art methods is provided in Section 2.3.1.1.

In this chapter, we use the Dense Trajectories, which are presented in detail in Section 3.2.1.2. We apply a dense sampling to extract interest points and we track these interest points using a dense optical flow field. Each interest point is tracked during the same number of following video frames.

Given a sample feature point $P_1 = (x_1, y_1)$ successfully tracked in $L \geq 1$ consecutive video frames, we define a tracklet T as a concatenation of $L+1$ following positions of the feature point P_1:

$$T = (P_1, P_2, ..., P_{L+1}) = ((x_1, y_1), (x_2, y_2), ..., (x_{L+1}, y_{L+1})). \tag{6.1}$$

In each video sequence we extract a set of trajectories $S = \{T_i\}_{i=1}^{|S|}$ [1], where each trajectory is of equal length.

[1] In order to extract spatio-temporal positions of trajectories, we modified the source code of the Dense Trajectories.

By extracting dense trajectories we provide a good coverage of a video sequence and we ensure extraction of meaningful features.

The Dense Trajectories were selected based on their use in the recent literature and our evaluation of this technique with the bag-of-features approach and the Fisher vector encoding (see Section 3.4.2). However, our descriptor can be used to represent a trajectory extracted by any state-of-the-art trajectory detector.

6.2.2 Trajectory Shape and Relative Trajectory Shape Descriptors

In this section, we present descriptors to encode characteristics of a given trajectory. In particular, we present the Trajectory Shape and Relative Trajectory Shape (TSRTS) descriptor, which captures and combines the discriminative power of the two following descriptors:

- Trajectory Shape (TS) descriptor, which encodes shape characteristics of a trajectory, and it encodes a local motion pattern,
- Relative Trajectory Shape (RTS) descriptor, which encodes relative positions of trajectory's elements according to the central point of the defined dynamic coordinate system.

The Trajectory Shape and Relative Trajectory Shape (TSRTS) descriptor represents a sample trajectory T as a feature vector $TSRTS$ as follows:

$$TSRTS = \left[(\vartheta_X)^T, \quad (\vartheta_Y)^T, \quad (X - H_X)^T, \quad (Y - H_Y)^T\right]^T, \tag{6.2}$$

where the first two elements of the $TSRTS$ descriptor, *i.e.* $(\vartheta_X)^T$ and $(\vartheta_Y)^T$, are referred to as the Trajectory Shape descriptor (see Section 6.2.2.1), and the two remaining parts of the $TSRTS$ descriptor, *i.e.* $(X - H_X)^T$ and $(Y - H_Y)^T$, correspond to the Relative Trajectory Shape descriptor (see Section 6.2.2.2).

6.2.2.1 Trajectory Shape Descriptor

In this section, we present the Trajectory Shape (TS) descriptor to encode a local trajectory. A brief description of this descriptor is presented in Chapter 3.

The Trajectory Shape descriptor is defined as a sequence of displacement vectors normalized by the sum of displacement vectors magnitudes.

Given a trajectory T, defined in the Equation 6.1, we decompose it into two vectors $X = [x_1, x_2, ..., x_{L+1}]^T$ and $Y = [y_1, y_2, ..., y_{L+1}]^T$. Then, we compute displacement vectors θ_X and θ_Y as follows:

$$\theta_X = \Delta(X - \frac{1}{L+1}\sum_{i=1}^{L+1} X_i), \tag{6.3}$$

$$\theta_Y = \Delta(Y - \frac{1}{L+1}\sum_{i=1}^{L+1} Y_i), \tag{6.4}$$

where X_i and Y_i are the i-th elements of the vector X and Y, respectively, and the function $\Delta(\cdot)$ computes differences between the consecutive elements of the given vector, *i.e.*:

$$\Delta(\upsilon) = [\upsilon_2 - \upsilon_1, \upsilon_3 - \upsilon_2, ..., \upsilon_{|\upsilon|} - \upsilon_{|\upsilon|-1}]^T. \tag{6.5}$$

Then, we normalize the calculated displacement vectors θ_X and θ_Y by the sum of their magnitudes, *i.e.* we calculate the vectors ϑ_X and ϑ_Y as follows:

$$\vartheta_X = \frac{\theta_X}{\sum_{i=1}^{L}\sqrt{\theta_{X_i}^2 + \theta_{Y_i}^2}}, \tag{6.6}$$

$$\vartheta_Y = \frac{\theta_Y}{\sum_{i=1}^{L}\sqrt{\theta_{X_i}^2 + \theta_{Y_i}^2}}, \tag{6.7}$$

where θ_{X_i} and θ_{Y_i} represent the i-th elements of the vector θ_X and θ_Y, respectively.

Finally, we define the Trajectory Shape (TS) descriptor as the concatenation of the vector ϑ_X and the vector ϑ_Y, *i.e.*:

$$TS = \left[(\vartheta_X)^T, \quad (\vartheta_Y)^T\right]^T. \tag{6.8}$$

The TS descriptor encodes shape characteristics of a local trajectory and it encodes a local motion pattern.

6.2.2.2 Relative Trajectory Shape Descriptor

In this section, we propose the Relative Trajectory Shape (RTS) descriptor to encode a local trajectory.

We define the Relative Trajectory Shape descriptor as a sequence of relative positions of a trajectory according to the central point of the defined dynamic coordinate system.

We are given a trajectory T (see Eq. 6.1):

$$T = (P_1, P_2, ..., P_{L+1}) = ((x_1, y_1), (x_2, y_2), ..., (x_{L+1}, y_{L+1})), \tag{6.9}$$

for which we define the dynamic coordinate system as a trajectory H:

$$H = (P'_1, P'_2, ..., P'_{L+1}) = ((x'_1, y'_1), (x'_2, y'_2), ..., (x'_{L+1}, y'_{L+1})), \tag{6.10}$$

where for each point $P_i = (x_i, y_i)$ of the trajectory T there exists a point $P'_i = (x'_i, y'_i)$ in the same video frame, which is the central point of our dynamic coordinate system. The

definition of the dynamic coordinate system is presented in Section 6.2.3.

Given a trajectory T (Eq. 6.9), we decompose it into two vectors $X = [x_1, x_2, ..., x_{L+1}]^T$ and $Y = [y_1, y_2, ..., y_{L+1}]^T$. Similarly, we decompose the trajectory H (Eq. 6.10), defined for the trajectory T, into two vectors $H_X = [x'_1, x'_2, ..., x'_{L+1}]^T$ and $H_Y = [y'_1, y'_2, ..., y'_{L+1}]^T$.

We define the Relative Trajectory Shape (RTS) descriptor as a sequence of relative positions of a trajectory according to the central point of the defined dynamic coordinate system:

$$RTS = \left[(X - H_X)^T, \quad (Y - H_Y)^T\right]^T, \tag{6.11}$$

The RTS descriptor encodes shape characteristics of a trajectory with respect to the center of the dynamic coordinate system. Therefore, if we use the RTS descriptor together with a local feature encoding technique, we will introduce relative spatial positions of a trajectory to the local feature encoding approach. The experiments in Section 6.4 show that adding the Trajectory Shape descriptor to the Relative Trajectory Shape descriptor (what is called the Trajectory Shape and Relative Trajectory Shape) significantly improves the action recognition accuracy.

6.2.3 Dynamic Coordinate System

Figure 6.1 – Samples of estimated head positions for the KTH dataset.

The Relative Trajectory Shape descriptor, presented in Section 6.2.2.2, is based on the dynamic coordinate system. As the central point of the dynamic coordinate system, we choose the head position, providing description invariant to camera viewpoint changes. We estimate the head position in each frame of a video, and combine all the estimations from consecutive video frames to obtain a trajectory of a head.

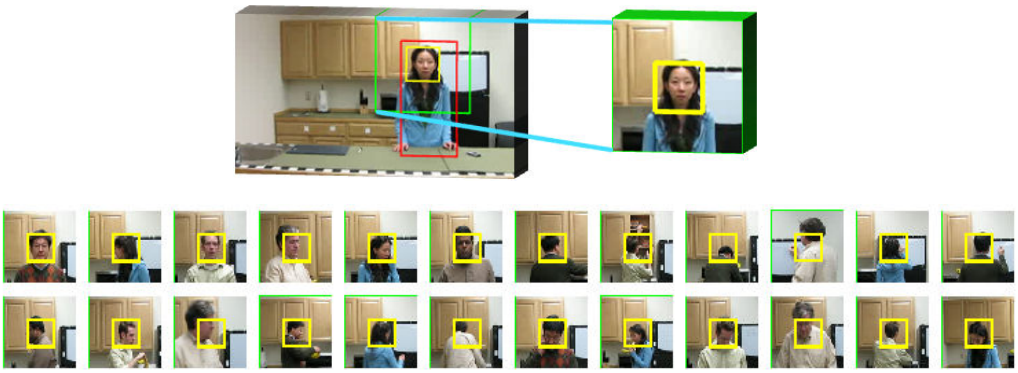


Figure 6.2 – Samples of estimated head positions for the URADL dataset.

Head detection is of particular interest in our action recognition framework, thus we have to ensure robust localization of this body part. To estimate a head position in a video sequence, we combine several state-of-the-art object detectors:

- People detector based on Histogram of Oriented Gradients (HOG) [Dalal 2005],
- Head detector based on Local Binary Patterns (LBP) [Ojala 2002],
- Face detector based on Haar-like features [Viola 2001].

Each of these detectors is applied in each frame independently. Cues provided by these object detectors (people, head, and face) can be additionally combined with motion information from the background subtraction method, which allows to remove erroneous detections.

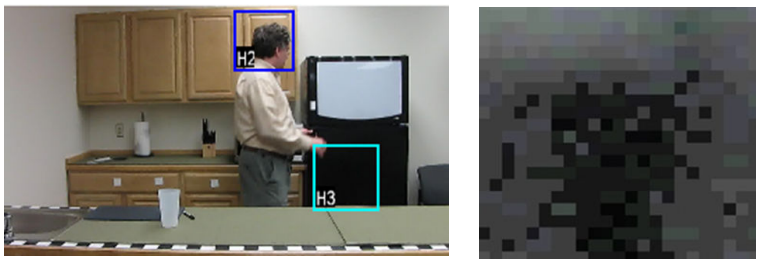


Figure 6.3 – Results of the head detector on a sample video frame from the URADL dataset (left image), and the visualization of the false positive detection (*i.e.* the detection on the fridge) using histogram equalization method on the grayscale image (right image). The histogram equalization method increases the global contrast of an image spreading out the most frequent intensity values. The visualization of the false positive detection resembles the shape of the head.

Then, we associate the obtained object detection results in time to form object trajectories. We follow the tracking by detection approach of [Ferrari 2008], which has

shown to provide excellent results in tracking upper-bodies in videos. We group detections using the Clique Partitioning algorithm. We define a similarity measure between objects as the ratio of the area of the intersection to the area of the union of bounding-boxes. The obtained object trajectories allow to remove single, erroneous detections, see Figure 6.3.

Although the above detectors achieve very good results on our datasets, they fail to provide detections in all video frames. Therefore, we use the obtained detection results as an input to the tracking algorithm to overcome missed detections. We use the Tracking-Learning-Detection (TLD) tracking algorithm [Kalal 2010]. The TLD is a real-time algorithm for tracking of unknown objects in videos, where the object of interest is defined by a bounding box in a single video frame. The TLD simultaneously tracks the object, learns its appearance, and detects it in the video. In order to increase the tracking results, we apply the TLD algorithm for both forward and backward in time tracking, *i.e.* for both next and previous video frames.

The detection based tracking and the TLD tracking methods provide multiple hypothesis along out of which we select the most likely one. This selection is based on the probability framework $\mathcal{P}$. The final position of the head is obtained by maximizing the trajectory-depended probability:

$$\mathcal{P}(l_h|t_{h,i}) = \sum_{z\in\{f,b\}} \mathcal{P}(l_z \mid t_{h,i} \propto t_{z,j})\, \mathcal{P}(t_{h,i} \propto t_{z,j}), \tag{6.12}$$

where l_h is a head location and $t_{h,i}$ is their corresponding trajectory. $\propto$ describes proportional variance of trajectory $t_{h,i}$ w.r.t. trajectory of other body part $t_{z,j}$. f and b refer to the face detection and the full body detection, respectively. Finally, we smooth the obtained head estimation trajectory by replacing rapid object displacements with the interpolated results.

Sample head positions estimated by this method are presented in Figure 6.1 and Figure 6.2. Moreover, sample estimated head positions together with the extracted dense trajectories are presented in Figure 6.4.

6.2.4 Trajectory Filtering

In some scenarios, the Dense Trajectories approach extracts local trajectories not related to the people performing the actions of interest; *e.g.* this can happen due to: video noise, people moving in the background, or moving background. The noisy, background trajectories typically mislead our action recognition framework and they decrease the action recognition accuracy. Therefore, we use the people detection results from the head estimation framework (Section 6.2.3) to remove the noisy, background trajectories. We simply filter these trajectories which are not close enough to the detected people, *i.e.* which are not inside the bounding box of people.

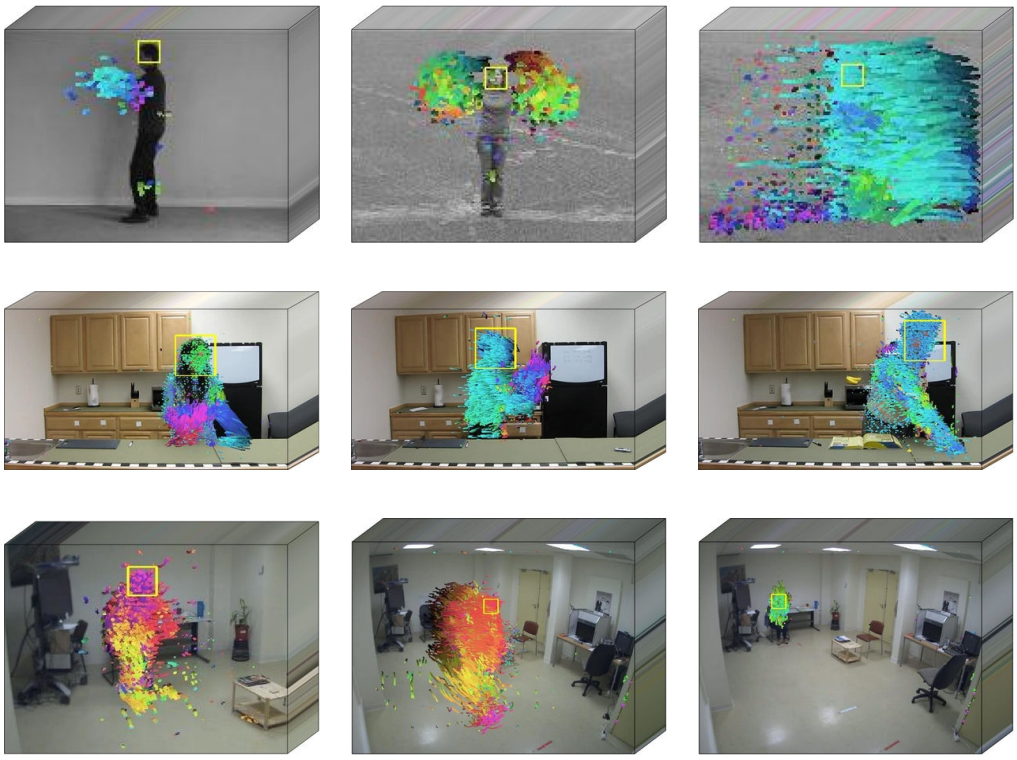


Figure 6.4 – Sample video frames with the extracted trajectories and the estimated head positions for the KTH dataset (first row), the URADL dataset (second row), and the CHU Nice Hospital dataset (third row).

6.3 Approach Overview

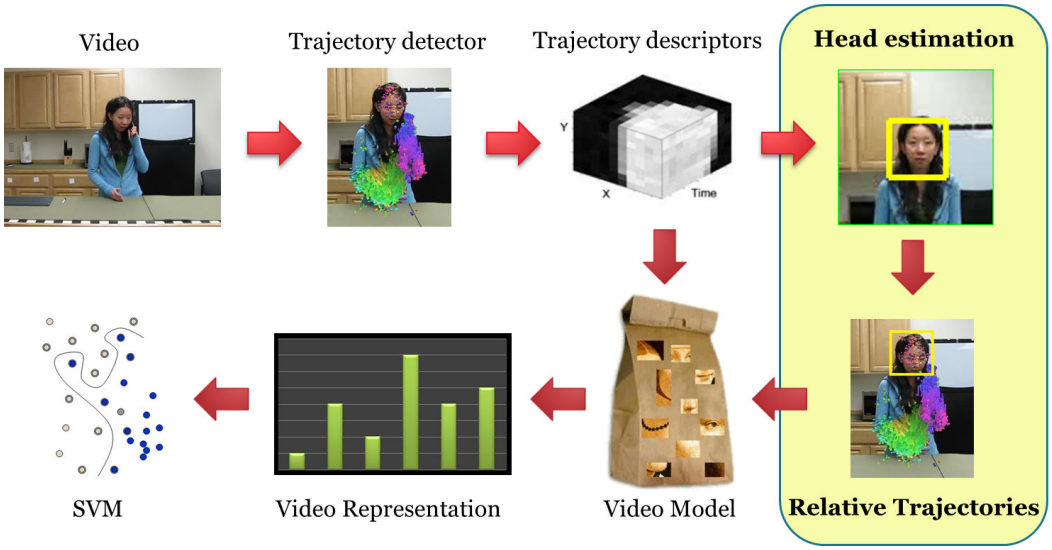


Figure 6.5 – Overview of the proposed action recognition framework.

In this section, we present our action recognition framework based on the introduced Relative Trajectories.

The overview of the proposed approach is presented in Figure 6.5.

In the first step of our approach, we extract local trajectories. In order to do that, we compute the Dense Trajectories in a video sequence; we apply a dense sampling to extract interest points and we track these interest points using a dense optical flow field (see Section 3.2.1.2). We extract spatio-temporal positions of trajectories and we extract local spatio-temporal video volumes around these detected trajectories. By extracting dense trajectories, we provide a good coverage of a video sequence and we ensure extraction of meaningful features. The Dense Trajectories were selected based on their use in the recent literature. However, our descriptor can be used together with any other algorithm extracting local trajectories.

Then, in the second step of our approach, we use the head estimation framework and we use the proposed Trajectory Shape and Relative Trajectory Shape descriptor to represent local trajectories. Moreover, we extract the Histogram of Oriented Gradients, Histogram of Optical Flow, and Motion Boundary Histogram descriptors for each local spatio-temporal video volume, as these descriptors carry complementary information.

Once the descriptors are calculated in a video sequence, we use them to represent this video sequence. We apply the Fisher vector encoding, which was introduced in Section 3.2.2.2. We compute a separate video representation for each descriptor, and we

concatenate the calculated Fisher vector based representations into a single feature vector.

Finally, we apply the Support Vector Machines to classify video representations into action categories (see Section 3.2.3). We use the Support Vector Machines with the linear kernel, we use the one-vs-all approach for multi-class classification.

6.4 Experiments

In this section, we present an evaluation, comparison, and analysis of the proposed Relative Trajectory Shape descriptor and the proposed action recognition approach.

The experiments are performed on 4 state-of-the-art action recognition datasets.

- The main and detail experiments are performed with Fisher vector encoding on:
  - URADL dataset,
  - MSR Daily Activity 3D dataset.
- We also present several experiments with the bag-of-features approach on:
  - KTH dataset,
  - CHU Nice Hospital dataset.

The selected datasets vary in the number of actions, types of actions, number of videos, and challenges. In order to train detectors (people, head, and face), we use three annotated datasets (TrecVid [Smeaton 2006], TUD [Andriluka 2008], and our own laboratory dataset).

In this section we use the following abbreviations:

- TS – Trajectory Shape descriptor,
- RTS – Relative Trajectory Shape descriptor,
- #PCA – # descriptor with PCA,
- #k@ – # descriptor applied with Fisher vector encoding using the codebook size @,
- DT – Dense Trajectories, *i.e.* TS, HOG, HOF, and MBH descriptors.

The remainder of the section is organized as follows. In Section 6.4.1, we present experiments on the URADL dataset. Section 6.4.2 presents experiments on the MSR Daily Activity 3D dataset. In Section 6.4.3, we present experiments on the KTH dataset. Then, in Section 6.4.4, we present experiments on the CHU Nice Hospital dataset. Finally, we present the summary and analysis of the results in Section 6.4.5.

6.4.1 URADL Dataset

The University of Rochester Activities of Daily Living dataset (in short, the URADL dataset) is presented in Section 3.3.3.

The following experiments are based on the Fisher vector encoding and 6 various codebook sizes: $\{16, 32, 64, 128, 256, 512\}$. Moreover, we investigate the influence of using the Principal Component Analysis (PCA) technique with descriptors on action recognition accuracy.

The detail evaluation results are presented in Figure 6.6.

The Trajectory Shape (TS) descriptor achieves 80.67% of mean class accuracy. The TS with PCA (TSPCA) decreases this result to as much as 78.67%.

The proposed Relative Trajectory Shape (RTS) descriptor achieves 78.67% without PCA and 87.33% with PCA (RTSPCA); the PCA improves the result for the RTS descriptor by 8.66%. The proposed descriptor outperforms the TS descriptor by 6.66%.

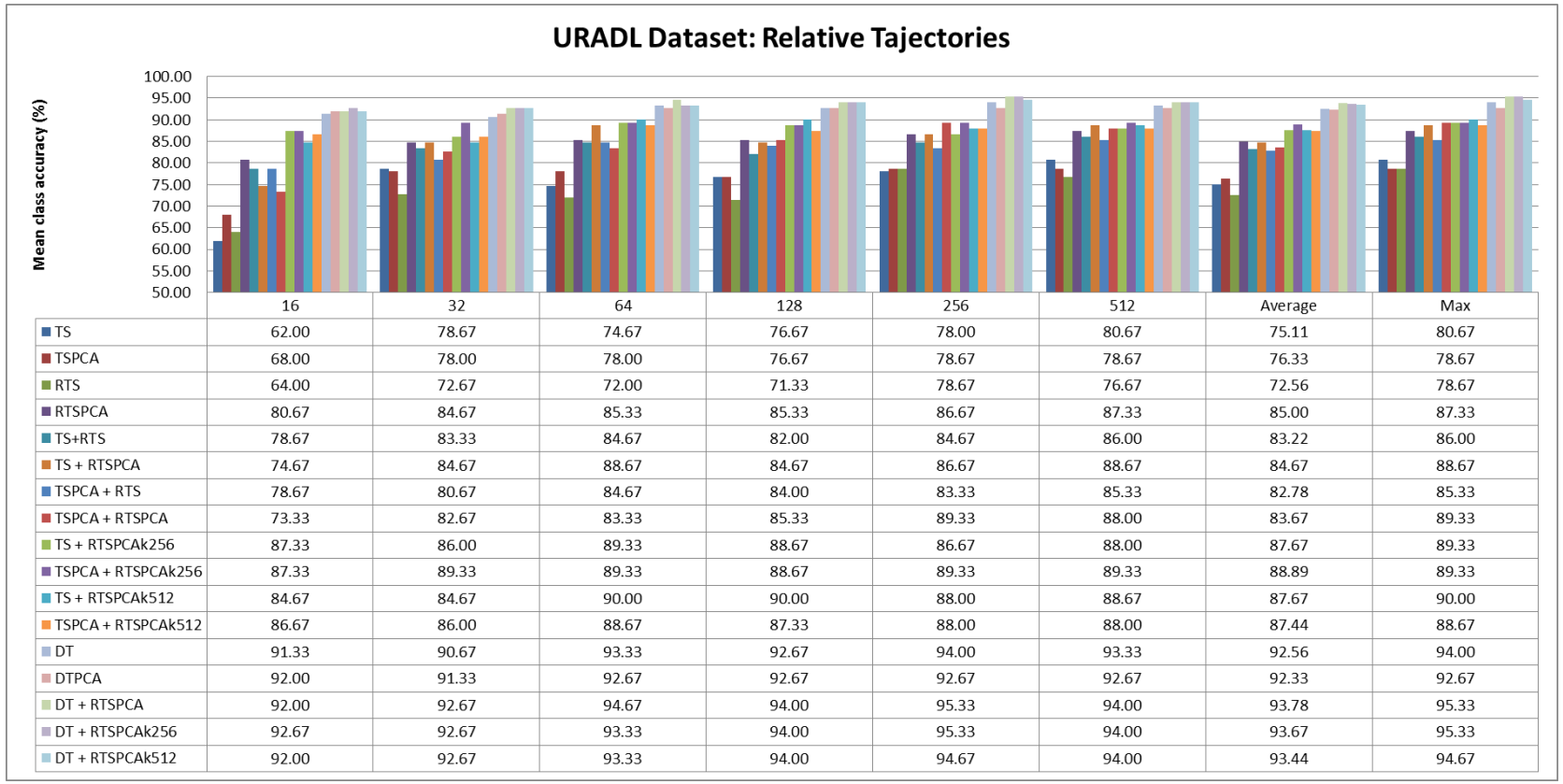
Then, we fuse the TS and RTS descriptors, with and without using the PCA, and we obtain the following results: TS + RTS representation achieves 86%, TS + RTSPCA achieves 88.67%, TSPCA + RTS achieves 85.33%, and TSPCA + RTSPCA achieves 89.33%. The fusion TSPCA + RTSPCA achieves the best action recognition accuracy, and it outperforms the TS descriptor by 8.66%.

Moreover, we observe that the RTSPCA achieves a very good result with the codebook size 256 (RTSPCAk256) and 512 (RTSPCAk512), so we fuse the TS and RTS descriptors using various codebook sizes (see Figure 6.6). The best result 90% is achieved by the TSk64 + RTSPCAk512, and the same results is achieved by the TSk128 + RTSPCAk512.

Then, we evaluate the action recognition accuracy of the Dense Trajectories (*i.e.* TS, HOG, HOF, and MBH descriptors fused together). The Dense Trajectories representation achieves 94% without PCA and 92.67% with PCA.

Finally, we fuse the Dense Trajectories representation and the proposed RTSPCA, and we achieve 95.33% of mean class accuracy, which is the best obtained results.

In summary, the RTS representation outperforms the TS representation, the fusion of the TS and the RTS improves action recognition accuracy in comparison to the TS representation alone, and the fusion of the RTS representation and the Dense Trajectories representation improves the action recognition accuracy in comparison to the DT representation alone.



	16	32	64	128	256	512	Average	Max
TS	62.00	78.67	74.67	76.67	78.00	80.67	75.11	80.67
TSPCA	68.00	78.00	78.00	76.67	78.67	78.67	76.33	78.67
RTS	64.00	72.67	72.00	71.33	78.67	76.67	72.56	78.67
RTSPCA	80.67	84.67	85.33	85.33	86.67	87.33	85.00	87.33
TS+RTS	78.67	83.33	84.67	82.00	84.67	86.00	83.22	86.00
TS + RTSPCA	74.67	84.67	88.67	84.67	86.67	88.67	84.67	88.67
TSPCA + RTS	78.67	80.67	84.67	84.00	83.33	85.33	82.78	85.33
TSPCA + RTSPCA	73.33	82.67	83.33	85.33	89.33	88.00	83.67	89.33
TS + RTSPCAk256	87.33	86.00	89.33	88.67	86.67	88.00	87.67	89.33
TSPCA + RTSPCAk256	87.33	89.33	89.33	88.67	89.33	89.33	88.89	89.33
TS + RTSPCAk512	84.67	84.67	90.00	90.00	88.00	88.67	87.67	90.00
TSPCA + RTSPCAk512	86.67	86.00	88.67	87.33	88.00	88.00	87.44	88.67
DT	91.33	90.67	93.33	92.67	94.00	93.33	92.56	94.00
DTPCA	92.00	91.33	92.67	92.67	92.67	92.67	92.33	92.67
DT + RTSPCA	92.00	92.67	94.67	94.00	95.33	94.00	93.78	95.33
DT + RTSPCAk256	92.67	92.67	93.33	94.00	95.33	94.00	93.67	95.33
DT + RTSPCAk512	92.00	92.67	93.33	94.00	94.67	94.00	93.44	94.67

Figure 6.6 – URADL dataset: Evaluation results of Relative Trajectories. The plot presents the mean class accuracy of descriptors with respect to the codebook size (the “x” axis).

6.4.2 MSR Daily Activity 3D Dataset

The MSR Daily Activity 3D dataset is presented in Section 3.3.4.

The following experiments are based on the Fisher vector encoding and 6 various codebook sizes: $\{16, 32, 64, 128, 256, 512\}$. Moreover, we investigate the influence of using the Principal Component Analysis (PCA) technique with descriptors on action recognition accuracy. We use the detected head positions provided for this dataset.

The detail evaluation results are presented in Figure 6.7.

The Trajectory Shape (TS) descriptor achieves 68.13% of mean class accuracy. The TS with PCA (TSPCA) decreases this result to as much as 67.19%.

The proposed Relative Trajectory Shape (RTS) descriptor achieves 57.60% without PCA and 68.13% with PCA (RTSPCA); the PCA improves the result for the RTS descriptor by 10.53%. The RTS descriptor with PCA achieves the same results as the TS descriptor.

Then, we fuse the TS and RTS descriptors, with and without using the PCA, and we obtain the following results: TS + RTS representation achieves 70.31%, TS + RTSPCA achieves 75%, TSPCA + RTS achieves 68.44%, and TSPCA + RTSPCA achieves 75.63%. The fusion TSPCA + RTSPCA achieves the best result, and it outperforms the TS descriptor by 7.5%.

Then, we evaluate the action recognition accuracy of the Dense Trajectories (DT, *i.e.* TS, HOG, HOF, and MBH descriptors fused together). The Dense Trajectories representation achieves 76.25% without PCA and 75.31% with PCA (DTPCA).

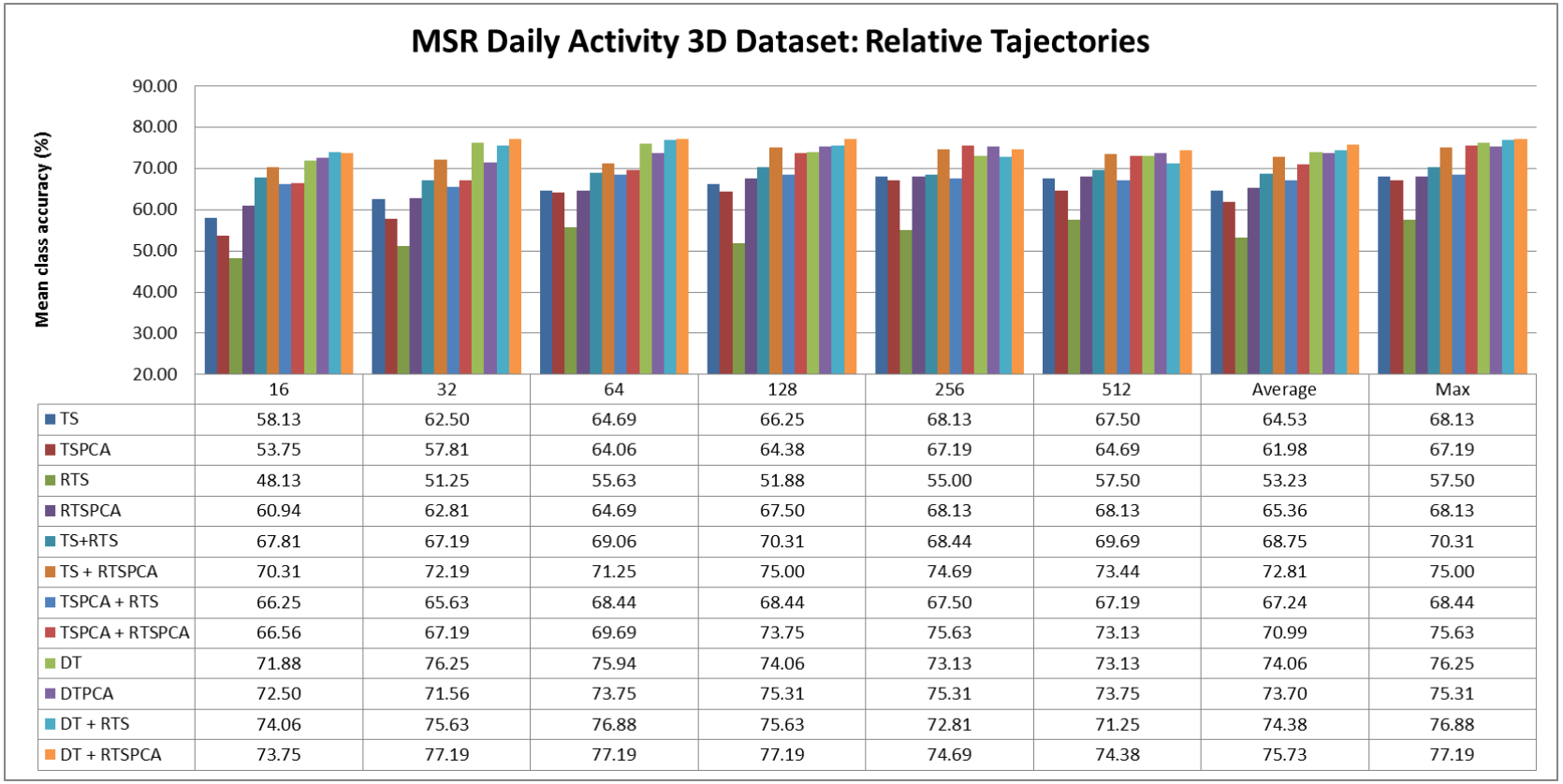
Finally, we fuse the Dense Trajectories representation and the proposed RTSPCA, and we achieve 77.19% of mean class accuracy, which is the best obtained results.

When we visualize the dense trajectories extracted from the MSR Daily Activity 3D dataset, we observe that lots of trajectories do not correspond to the main actors performing actions (see Figure 6.8); they are extracted due to noise, motion of background people, and motion of people reflecting on the glass. Therefore, we also apply the Trajectory Filtering on this dataset (see Section 6.2.4) to remove the background trajectories.

The detail evaluation results are presented in Figure 6.9.

The Trajectory Filtering improves the mean class accuracy of:

- TS representation from 68.13% to 69.38%,
- RTS representation from 57.50% to 64.69%,



	16	32	64	128	256	512	Average	Max
TS	58.13	62.50	64.69	66.25	68.13	67.50	64.53	68.13
TSPCA	53.75	57.81	64.06	64.38	67.19	64.69	61.98	67.19
RTS	48.13	51.25	55.63	51.88	55.00	57.50	53.23	57.50
RTSPCA	60.94	62.81	64.69	67.50	68.13	68.13	65.36	68.13
TS+RTS	67.81	67.19	69.06	70.31	68.44	69.69	68.75	70.31
TS + RTSPCA	70.31	72.19	71.25	75.00	74.69	73.44	72.81	75.00
TSPCA + RTS	66.25	65.63	68.44	68.44	67.50	67.19	67.24	68.44
TSPCA + RTSPCA	66.56	67.19	69.69	73.75	75.63	73.13	70.99	75.63
DT	71.88	76.25	75.94	74.06	73.13	73.13	74.06	76.25
DTPCA	72.50	71.56	73.75	75.31	75.31	73.75	73.70	75.31
DT + RTS	74.06	75.63	76.88	75.63	72.81	71.25	74.38	76.88
DT + RTSPCA	73.75	77.19	77.19	77.19	74.69	74.38	75.73	77.19

Figure 6.7 – MSR Daily Activity 3D dataset: Evaluation results of Relative Trajectories. The plot presents the mean class accuracy of descriptors with respect to the codebook size (the "x" axis).

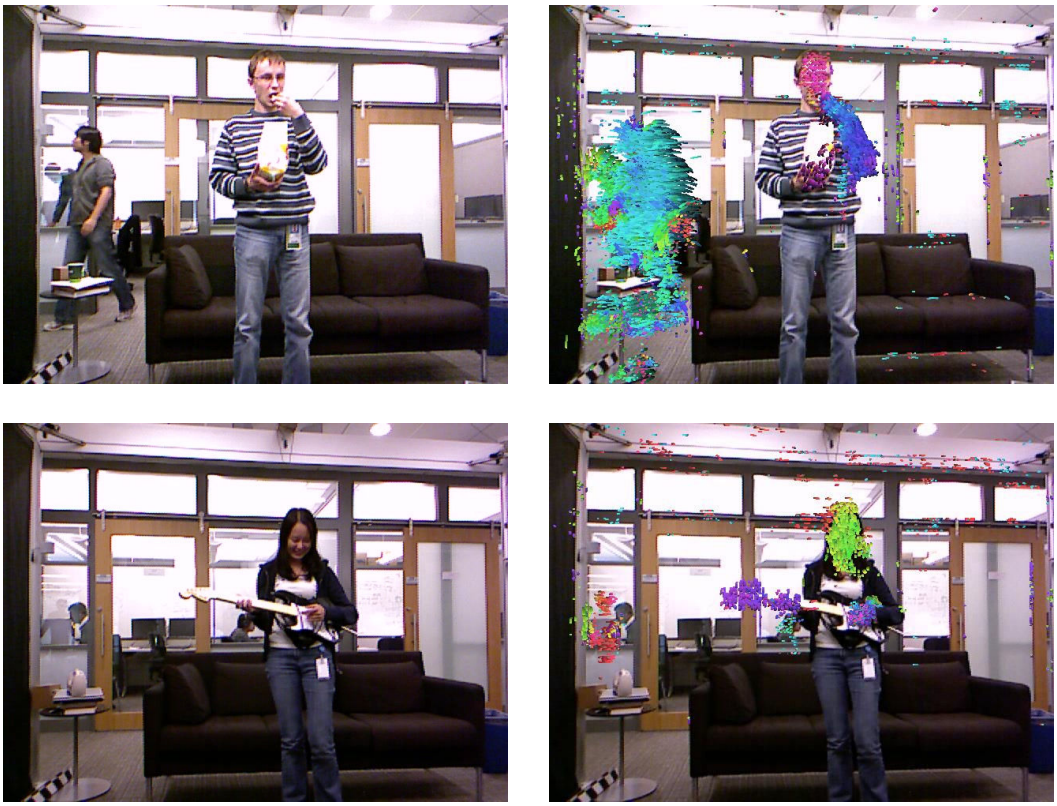
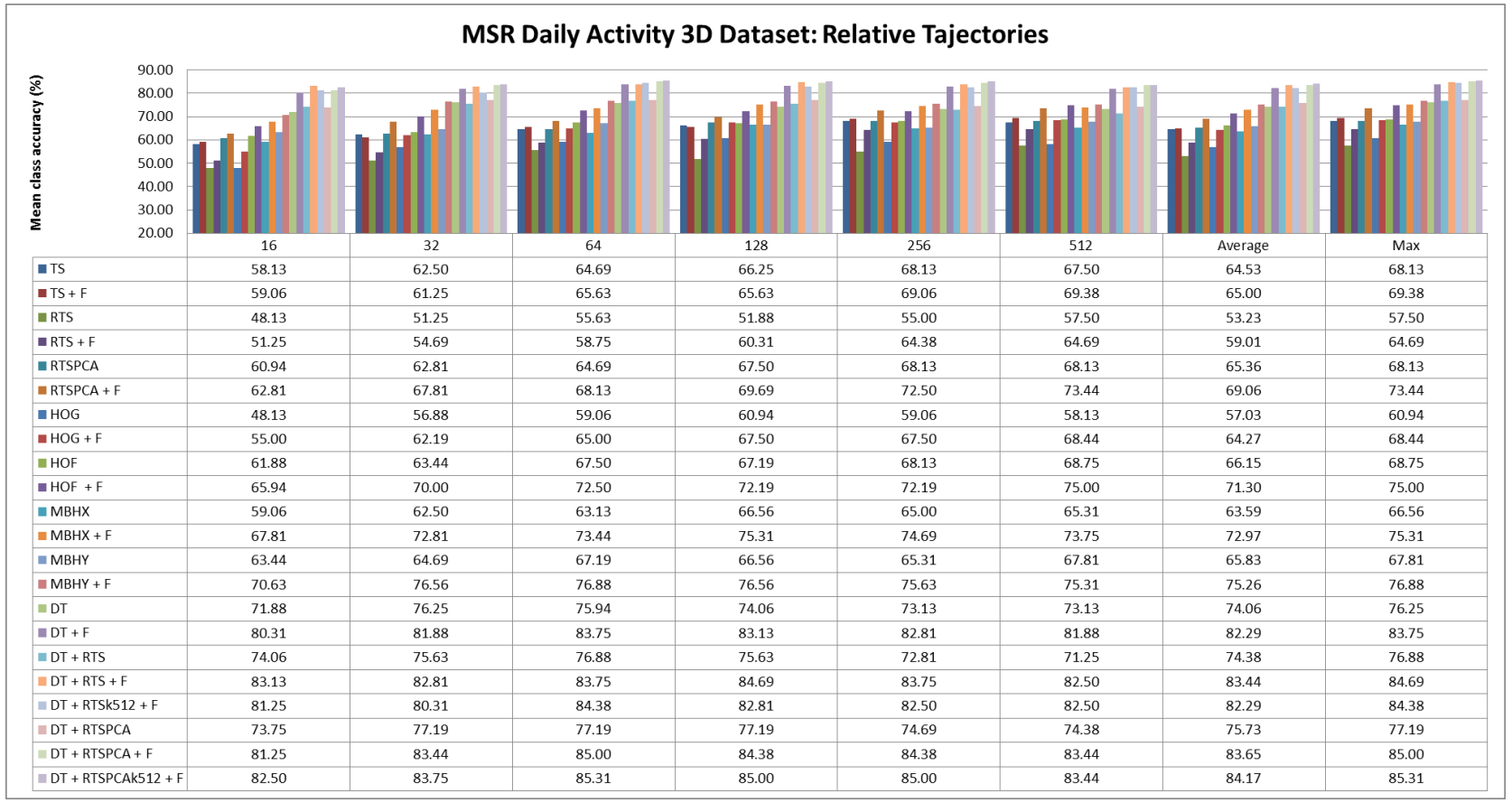


Figure 6.8 – MSR Daily Activity 3D dataset: Sample video frames (left column) with the extracted trajectories (right column). The dense trajectories are extracted due to motion of main actors performing actions, but also due to noise, motion of background people (see images in the first row), and motion of people reflecting on the glass (see images in the second row).



	16	32	64	128	256	512	Average	Max
TS	58.13	62.50	64.69	66.25	68.13	67.50	64.53	68.13
TS + F	59.06	61.25	65.63	65.63	69.06	69.38	65.00	69.38
RTS	48.13	51.25	55.63	51.88	55.00	57.50	53.23	57.50
RTS + F	51.25	54.69	58.75	60.31	64.38	64.69	59.01	64.69
RTSPCA	60.94	62.81	64.69	67.50	68.13	68.13	65.36	68.13
RTSPCA + F	62.81	67.81	68.13	69.69	72.50	73.44	69.06	73.44
HOG	48.13	56.88	59.06	60.94	59.06	58.13	57.03	60.94
HOG + F	55.00	62.19	65.00	67.50	67.50	68.44	64.27	68.44
HOF	61.88	63.44	67.50	67.19	68.13	68.75	66.15	68.75
HOF + F	65.94	70.00	72.50	72.19	72.19	75.00	71.30	75.00
MBHX	59.06	62.50	63.13	66.56	65.00	65.31	63.59	66.56
MBHX + F	67.81	72.81	73.44	75.31	74.69	73.75	72.97	75.31
MBHY	63.44	64.69	67.19	66.56	65.31	67.81	65.83	67.81
MBHY + F	70.63	76.56	76.88	76.56	75.63	75.31	75.26	76.88
DT	71.88	76.25	75.94	74.06	73.13	73.13	74.06	76.25
DT + F	80.31	81.88	83.75	83.13	82.81	81.88	82.29	83.75
DT + RTS	74.06	75.63	76.88	75.63	72.81	71.25	74.38	76.88
DT + RTS + F	83.13	82.81	83.75	84.69	83.75	82.50	83.44	84.69
DT + RTSk512 + F	81.25	80.31	84.38	82.81	82.50	82.50	82.29	84.38
DT + RTSPCA	73.75	77.19	77.19	77.19	74.69	74.38	75.73	77.19
DT + RTSPCA + F	81.25	83.44	85.00	84.38	84.38	83.44	83.65	85.00
DT + RTSPCAk512 + F	82.50	83.75	85.31	85.00	85.00	83.44	84.17	85.31

Figure 6.9 – MSR Daily Activity 3D dataset: Evaluation results of Relative Trajectories with Trajectory Filtering. The plot presents the mean class accuracy of descriptors with respect to the codebook size (the "x" axis).

- RTSPCA representation from 68.13% to 73.44%,
- HOG representation from 60.94% to 68.44%,
- HOF representation from 68.75% to 75%,
- MBHX representation from 66.56% to 75.31%,
- MBHY representation from 67.81% to 76.88%,
- DT representation from 76.25% to 83.75%,
- DT + RTS representation from 76.88% to 84.69%,
- DT + RTSPCA representation from 77.19% to 85%.

We observe that the Trajectory Filtering improves the action recognition accuracy for every descriptor individually and for the fusions of the descriptors.

The best result is achieved by the DT + RTSPCA representation, and it gives 77.19% without the Trajectory Filtering and 85% with the Trajectory Filtering.

Moreover, we observe that the RTS/RTSPCA achieves very good result with the codebook size 512, and this is why we fuse the DT with RTS/RTSPCA using various codebook sizes (see Figure 6.9). The fusion of DT + RTSPCAk512 slightly improves the action recognition accuracy from 85% to 85.31%.

6.4.3 KTH Dataset

The KTH Action dataset (in short, the KTH dataset) is presented in Section 3.3.2.

We evaluate the TS, RTS, and the fusion of the TS and RTS representations on this dataset. The following experiments are based on the bag-of-features approach and the codebook size 1000. We do not use the face detector on this dataset as people are too far from the camera.

There are two commonly used experimental setups to evaluate an approach on the KTH dataset: official splitting-based evaluation scheme and leave-one-person-out cross-validation evaluation scheme (see Section 3.3.2). We present the evaluation results using both the experimental setups.

The KTH dataset contains videos recorded in four different scenarios. The detail evaluation results for each scenario individually and all together are presented in Table 6.1.

Using official splitting-based evaluation scheme, overall, the TS achieves 91.67% of mean class accuracy, the RTS achieves 91.21%, and their fusion achieves 94.91%.

	Descriptors	Mean class accuracy (%) s1	s2	s3	s4	overall
Official Split	TS	98.15%	88.89%	88.89%	90.74%	91.67%
	RTS	96.30%	88.89%	87.04%	92.60%	91.21%
	TS+RTS	**98.15%**	**92.59%**	**92.59%**	**96.30%**	**94.91%**
LOOCV	TS	98.00%	92.00%	93.29%	94.67%	94.49%
	RTS	98.67%	89.33%	95.30%	96.67%	94.99%
	TS+RTS	**99.33%**	**93.33%**	**97.32%**	**98.67%**	**97.16%**

Table 6.1 – KTH dataset: Evaluation results of Relative Trajectories for each scenario individually and all together.

Using leave-one-person-out cross-validation evaluation scheme, overall, the TS achieves 94.49% of mean class accuracy, the RTS achieves 94.99%, and their fusion achieves 97.16%.

In both experimental setups the RTS representation significantly improves action recognition accuracy.

6.4.4 CHU Nice Hospital Dataset

The CHU Nice Hospital dataset (in short, the CHU dataset) is our locally collected dataset, and it is presented in Section 3.3.6.

We evaluate the TS and RTS representations on this dataset. Moreover, we evaluate the fusion of TS and RTS representations, and the fusion of TS, RTS, HOG, and HOF representations. The following experiments are based on the bag-of-features approach and the codebook size 1000.

Descriptors	Mean class accuracy
ST	86.3%
RST	87.2%
ST+RST	91.5%
ST+RST + HOG+HOF	93.0%

Table 6.2 – CHU Nice Hospital dataset: Evaluation results of Relative Trajectories.

The evaluation results are presented in Table 6.2. The TS representation achieves 86.3% of mean class accuracy, the RTS achieves 87.2%, and their fusion achieves 91.5%. The fusion of ST+RST and HOG+HOF representations achieves 93%.

The above results confirm that the RTS representation improves action recognition accuracy.

6.4.5 Results Summary and Analysis

Based on the above experimental results we observe that:

- The fusion of the Trajectory Shape and the Relative Trajectory Shape descriptors significantly improves the action recognition accuracy.
- The Trajectory Shape descriptor works better without the PCA.
- The Relative Trajectory Shape descriptors works better with PCA.
- The Dense Trajectories representation works better without the PCA on the URADL and the MSR Daily Activity 3D datasets.
- The fusion of the Dense Trajectories representation (*i.e*. the Trajectory Shape, Histogram of Oriented Gradients, Histogram of Oriented Flow, and Motion Boundary Histogram representations) and the Relative Trajectory Shape with PCA representation improves action recognition accuracy.

The experimental results confirm that the spatial information is important for action recognition. The proposed Relative Trajectory Shape descriptor introduces spatial information to a local feature encoding technique, and it enhances the discriminative properties of a trajectory representation and the proposed action recognition approach.

6.5 Conclusion

We have introduced the relative dense trajectories for action recognition in videos. We have proposed the Relative Trajectory Shape (RTS) descriptor based on relative positions of a trajectory according to the central point of our dynamic coordinate system. As the center of our dynamic coordinate system, we have chosen the head position, providing description invariant to camera viewpoint changes.

We have applied the RTS descriptor to encode local dense trajectories, and we have proposed to filter background trajectories. Using the Fisher vector encoding (the bag-of-features approach) and the Support Vector Machines, we have presented an extensive evaluation of our approach on four various action recognition datasets. The experiments have shown that the proposed RTS descriptor significantly improves action recognition accuracy. The RTS representation outperforms the TS representation, the fusion of the TS and the RTS improves action recognition accuracy in comparison to the TS representation alone, and the fusion of the RTS representation and the Dense Trajectories representation improves the action recognition accuracy in comparison to the Dense Trajectories representation alone.

CHAPTER 7

Geometric and Appearance Relations of Pairwise Features

Contents

7.1 Introduction 174
7.2 Geometric and Appearance Relations of Pairwise Features 176
7.2.1 Local Feature Extraction 176
7.2.2 Pairwise Features 176
7.2.3 Geometric and Appearance Relations of Pairwise Features 177
7.3 Approach Overview 178
7.4 Experiments 180
7.4.1 Implementation Details 181
7.4.2 KTH Dataset 181
7.4.3 URADL Dataset 182
7.4.4 HMDB51 Dataset 182
7.4.5 Results Summary and Analysis 183
7.5 Conclusion 183

In this chapter, we introduce a new representation of pairwise features, and we propose a new approach for action recognition based on the introduced features. The new representation is called the Geometric and Appearance Relations of Pairwise Features (GARPF). The GARPF representation is based on local spatio-temporal features, and it encodes relations among local features as pairwise features. The main idea is to capture the appearance relations among features (both visual and motion), and use geometric information to describe how these appearance relations are mutually arranged in the spatio-temporal space. Using three benchmark datasets for human action recognition, we demonstrate that our representation enhances the discriminative power of local features and improves action recognition accuracy.

7.1 Introduction

In Chapter 2 we present the most popular local feature encoding techniques. Among many, the most popular ones are the bag-of-features approach and the Fisher vector encoding.

Although the existing local feature encoding techniques are very successful, they also contain limitations. One of the main limitations of these models is that they simplify the structure of spatio-temporal video data assuming conditional independence across spatial and temporal domains. They compute only global statistics of local features, ignoring information about the spatio-temporal positions of features, relations among the features, and local densities of features. Thus, not using all the available information, they may fail to distinguish similar actions. A common way to overcome these limitations is to use either spatio-temporal grids [Laptev 2008] or multi-scale pyramids [Lazebnik 2006]. However, these methods are still limited in terms of a detailed description providing only a coarse representation.

Over the last years, several solutions have been proposed to overcome the above limitations of local feature encoding techniques. Most of the state-of-the-art solutions are based on higher-level feature representations, *i.e.* pairwise features and contextual features, see Section 2.3.2. Pairwise features and contextual features have been used to capture relations among features and to enrich local feature representations. Pairwise features capture relations among pairs of features while the contextual features describe relations among features within local neighbourhoods.

In this chapter, we focus on pairwise relations among features, *i.e.* we focus on pairwise features.

Pairwise features encode relations among local spatio-temporal features, such as spatio-temporal interest points and trajectories. They are very useful to distinguish videos with similar global distributions of local features, but with different placements and order of local features.

A brief literature overview of existing pairwise features based techniques is presented in Section 2.3.2.1.

The existing pairwise features based techniques use the discriminative power of individual features and they capture appearance relations [1] among features. However, they typically ignore information about the spatio-temporal geometric relations [2] between features (*i.e.* Δx, Δy, Δt). Moreover, some of the existing pairwise features based techniques can only handle small codebooks, and they ignore associations between geometric and appearance relations among features. Therefore, a new and optimized

[1] The appearance relation means here a relation between two appearance features.

[2] The geometric relation corresponds to the distance and orientation in the spatio-temporal space between two features.

representation is needed to create a finer description of pairwise features.

Different from the existing techniques, we introduce a new representation of pairwise features, and we propose a new approach for action recognition based on the introduced features:

- Pairwise Features: We introduce a new representation of pairwise features, called the Geometric and Appearance Relations of Pairwise Features (GARPF). The GARPF representation captures statistics of pairwise co-occurring local spatio-temporal features. It encodes geometric and appearance (visual and motion) relations among features, and also associations between these two types of information, all in a single descriptor. Calculating video representations with different geometrical arrangements among the features, we keep an important association between the appearance and the geometric information.
- Approach: We apply the Spatio-Temporal Interest Points approach to extract local features in a video sequence. We calculate the proposed Geometric and Appearance Relations of Pairwise Features. Then, we use the Fisher vector encoding to represent videos. The Fisher vector encoding is applied both with local features and the proposed GARPF features. Then, we combine the obtained video representations and, finally, we use the Support Vector Machines for action classification.
- Experiments: We present an evaluation of our approach on three publicly available state-of-the-art datasets for human action recognition. We show that the proposed representation enhances the discriminative power of local features and improves action recognition accuracy.

Compared to the state-of-the-art pairwise features, we encode more precise geometric information (orientation and space-time distances among features), we combine it with visual and motion appearance information (relations among any descriptors, such as HOG, HOF, and MBH); moreover, this combination is done in a single descriptor, without losing the association between geometric and appearance relations among features.

The remainder of the chapter is organized as follows. In Section 7.2, we propose the Geometric and Appearance Relations of Pairwise Features. Section 7.3 presents our action recognition framework. In Section 7.4, we present experimental results, comparison, and analysis. Finally, we conclude in Section 7.5.

7.2 Geometric and Appearance Relations of Pairwise Features

We propose a new representation of pairwise features, called the Geometric and Appearance Relations of Pairwise Features. The GARPF representation captures relations among pairs of features. The main idea is to capture the appearance (both visual and motion) relations among features, and use geometric information to describe how these appearance relations are mutually arranged in the spatio-temporal space. It allows to describe correlations among features detected in different body parts (*e.g.* correlation between head-body center features or hand-hand features).

The remainder of the section is organized as follows. In Section 7.2.1, we present local feature extraction. Section 7.2.2 presents the calculation of pairwise features. Finally, we propose the GARPF representation in Section 7.2.3.

7.2.1 Local Feature Extraction

Firstly, we extract local spatio-temporal features in videos. For each video sequence we extract a set of local features $\mathbb{P} = \{P_i\}_{i=1}^{|\mathbb{P}|}$, where $P_i = [P_i(x), P_i(y), P_i(t)]^T$ is a local feature located at spatial position ($P_i(x)$,$P_i(y)$) and time index $P_i(t)$. For every local feature P_i, we compute its local appearance descriptor(s) $\Xi(P_i)$.

A brief literature overview of local feature detectors and descriptors is presented in Section 2.3.1. Due to the large popularity, good results, and sparsity of the Spatio-Temporal Interest Points, we use the Harris3D detector, along with HOG and HOF descriptors. These algorithms were selected based on their use in the literature. However, our approach can be used with any other algorithms extracting local features.

7.2.2 Pairwise Features

Given detected local features $\mathbb{P} = \{P_i\}_{i=1}^{|\mathbb{P}|}$ in a video sequence V, $V \rightarrow \mathbb{P}$, for each spatio-temporal feature P_i we calculate all the features contained in its feature-centric space-time neighborhood:

$$\mathbb{N}_{m_{pd}}(P_i, \mathbb{P}) = \{P_j | P_j \in \mathbb{P} \wedge D(P_i, P_j) \leq m_{pd}\}, \tag{7.1}$$

where $D(\cdot, \cdot)$ is the Euclidean distance on 3D coordinates between two spatio-temporal features, *i.e.* interest points, and m_{pd} can be calculated from training videos as an average pairwise distance over action samples.

Then, we represent each video sequence V as a set of pairs of neighboring local spatio-temporal features:

$$\mathbb{V}(\mathbb{P}) = \{(P_i, P_j) | P_i \in \mathbb{P} \wedge P_j \in \mathbb{N}_{m_{pd}}(P_i, \mathbb{P})\}. \tag{7.2}$$

Every two local features P_i and P_j from the set $\mathbb{P}$ generate two pairs of local features for the set $\mathbb{V}(\mathbb{P})$, *i.e.* the pair (P_i,P_j) and the pair (P_j,P_i). Since both pairs have the same

geometric relation among the points (*i.e.* $[|\Delta x|, |\Delta y|, |\Delta t|]$), we remove all the redundant pairs of points so as to reduce the space and speed up the algorithm. We sort all the points of the video volume V by the spatio-temporal coordinates, and we select each pair of points only once, creating a set of unique pairs of points:

$$\begin{aligned}\mathbb{V}_U(\mathbb{P}) =&\{(P_i, P_j)|P_i \in \mathbb{P} \wedge P_j \in \mathbb{N}_{m_{pd}}(P_i, \mathbb{P}) \\ &\wedge [\sum_{d\in\{x,y,t\}} w_d \text{sgn}(P_i(d) - P_j(d))] > 0\},\end{aligned} \tag{7.3}$$

where sgn$(\cdot)$ is the signum function [3], and parameter w_d is the weight for the dimension d, which determines the order of local features in 3-dimensional space, and it is explained in Section 7.4.

7.2.3 Geometric and Appearance Relations of Pairwise Features

Given N_{tr} training videos $\mathbb{V}_{tr} = \{V_{tr}^{(i)}\}_{i=1}^{N_{tr}}$ and pairwise features $\mathbb{V}_U^{(tr)}$ that are extracted from the training videos, $\mathbb{V}_U^{(tr)} = \bigcup_{i=1}^{N_{tr}} \mathbb{V}_U(V_{tr}^{(i)} \rightarrow \mathbb{P})$, we show how to compute the Geometric and Appearance Relations of Pairwise Features (GARPF).

Firstly, for every pair of local features (P_i,P_j), where both local features are extracted from the same training video, we calculate the relative position of the latter local feature P_j to the reference former local feature P_i. In other words, we apply a dynamic coordinate system and assume that for each pair of local features (P_i,P_j) the center of the coordinate system is the first feature P_i. We combine all such created coordinate systems to get a single, Global Coordinate System (GCS). The GCS contains information about all the geometric relations among local features from the training videos, and it can be represented as a set of vectors of differences between pairs of points ($\mathbb{GCS}(\mathbb{V}_{tr}) = \{[\Delta x, \Delta y, \Delta t]\}$):

$$\mathbb{GCS}(\mathbb{V}_{tr}) = \{[P_j - P_i]|(P_i, P_j) \in \bigcup_{V_{tr}\in\mathbb{V}_{tr}} \mathbb{V}_U(V_{tr} \rightarrow \mathbb{P})\}. \tag{7.4}$$

Then, we cluster all the local features from the set $\mathbb{GCS}(\mathbb{V}_{tr})$ into k_g groups, and we map all the pairwise features extracted from the training videos to the closest clusters:

$$\begin{aligned}\mathbb{V}_U^{(\alpha)}(\mathbb{V}_{tr}) = \{(P_i, P_j) \in \bigcup_{V_{tr}\in\mathbb{V}_{tr}} \mathbb{V}_U(V_{tr} \rightarrow \mathbb{P}) : \\ \alpha = \arg\min_{\gamma} ||[P_j - P_i] - \mu_g(\gamma)||_2\}_{1\leq\alpha,\gamma\leq k_g},\end{aligned} \tag{7.5}$$

where $\mu_g(\gamma)$ is the center of the γ-th group. In other words, we decompose (map) the set of pairwise features $\mathbb{V}_U^{(tr)}$ extracted from all the training videos into k_g subsets

[3] The signum function of a real number x is defined as follows: 1 if $x > 0$, -1 if $x < 0$, and 0 otherwise.

$\{\mathbb{V}_U^{(\alpha)}(\mathbb{V}_{tr})\}_{\alpha=1}^{k_g}$ based on the geometric relations among the local features.

Section 7.2.1 defines that each local feature P_i is represented by both spatio-temporal coordinates and appearance descriptor(s) denoted as $\Xi(P_i)$, *e.g.* HOG and HOF descriptors. To enhance the discriminative power of pairwise features, we incorporate appearance descriptors to our GARPF representation. Therefore, for every pair of local features from each created set $\mathbb{V}_U^{(\alpha)}(\mathbb{V}_{tr})$ of decomposed pairwise features, we create a new descriptor by simply concatenating descriptors of the two local features, as follows:

$$\mathbb{V}_D^{(\alpha)}(\mathbb{V}_{tr}) = \{\Xi(P_i)||\Xi(P_j)|(P_i, P_j) \in \mathbb{V}_U^{(\alpha)}(\mathbb{V}_{tr})\}, \tag{7.6}$$

where $||$ is the concatenation operator. Then, we cluster such obtained descriptors into k_a groups, in each decomposed set of pairwise features separately.

In summary, we firstly decompose the set of pairwise features $\mathbb{V}_U^{(tr)}$ based on the geometric relations among local features. Then, we decompose such obtained subsets of pairwise features based on the appearance relations among local features. Such representation captures the appearance pairwise features (both visual and motion), and use geometric information to describe how these appearance relations among features are mutually arranged in the spatio-temporal space. Note that the combination of geometric and appearance information is done in a single descriptor, without losing the association between these two types of features.

The size of the new pairwise descriptor is two times larger than the size of the local feature descriptor. The size of the descriptor can be reduced by using the Principal Component Analysis or the Linear Discriminant Analysis, if necessary. However, calculating pairs of points only in their close neighborhoods, removing redundant pairs of points, decomposing the set of pairwise features into smaller subsets, and clustering features to small codebooks (*e.g.* employing the Fisher vector encoding for video representation), we strongly reduce the computation time of the proposed GARPF features as compared to the state-of-the-art pairwise features. Additionally, since all the clustering tasks are independent of each other, we can easily parallelize them.

7.3 Approach Overview

In this section, we present our action recognition framework based on the introduced GARPF features.

The overview of the proposed approach is presented in Figure 7.1.

In the first step of our approach, we extract local spatio-temporal features from video sequences. In order to do that, we use the Spatio-Temporal Interest Points proposed by Laptev [Laptev 2005], see Section 3.2.1.1. Firstly, we apply the Harris3D corner

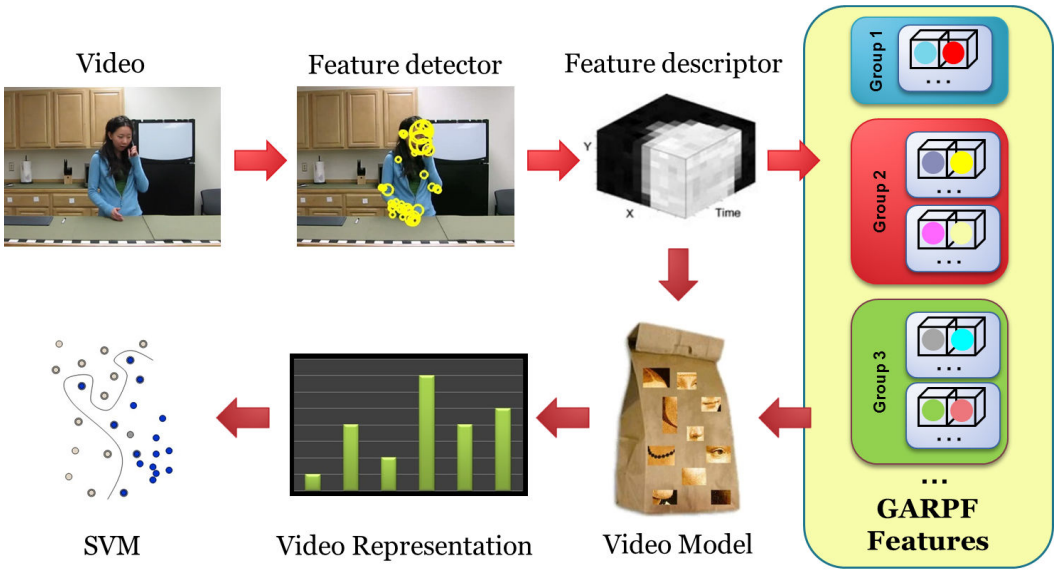


Figure 7.1 – Overview of the proposed action recognition framework.

detector to extract points of interest, and we detect interest points in multiple spatial and temporal scales. Then, we describe the neighbourhood of each interest point by two descriptors: Histogram of Oriented Gradients and Histogram of Optical Flow. The detector and descriptors were selected based on their use in the literature and achieved high action recognition accuracy on various datasets (see Section 3.4). Due to the large popularity of the Spatio-Temporal Interest Points, the selected detector and descriptors provide a good baseline for comparison with the state-of-the-art techniques. Note that our approach can be used with any other algorithm extracting local features, *e.g.* with the Dense Trajectories [Wang 2011a]. However, the Harris3D detector is relatively sparse, what allows for fast calculation of the GARPF features.

Then, we calculate the proposed Geometric and Appearance Relations of Pairwise Features. Therefore, we obtain 2 levels of features for each video sequence:

- Level 1 is based on descriptors of the Spatio-Temporal Interest Points. Thus, this level consists of one set of appearance features.

- Level 2 is based on the GARPF features. Thus, it is represented as k_g sets of appearance features that vary in geometric relations among features (see Section 7.2.3).

In summary, for each video sequence, we create 2 levels of features that consists of $1 + k_g$ sets of features.

Once the descriptors are calculated in a video sequence, we use them to represent this video sequence.

We apply the Fisher vector encoding, which was introduced in Section 3.2.2.2. We compute a separate video representation for each feature set independently, and we concatenate the calculated Fisher vector based representations into a single feature vector. Finally, we apply the Support Vector Machines with the linear kernel to classify video representations into action categories (see Section 3.2.3).

Alternatively, we can use the bag-of-features approach instead of the Fisher vector encoding. Then, to optimally fuse our video content representations, we apply the Multiple Kernel Learning (MKL) approach [Vishwanathan 2010], formulated for multi-class classification problem. Given a set of base kernel functions $\{K_i\}_{i=1}^{n}$, we search for the linear combination of the base kernel functions, that maximizes a global performance measure:

$$\begin{aligned} K(H_a, H_b) &= \sum_{i}^{n} \beta_i K_i(H_a, H_b) \\ \mathbf{s.t.} \quad \beta_k \geq 0, \quad &\sum_{k} \beta_k = 1, \quad k = 1 \dots n, \end{aligned} \tag{7.7}$$

where β_i is the weight for the kernel K_i, and the distance between two video representations is defined as the exponential χ^2 kernel (see Section 3.2.3.3).

The MKL learns the weights for the video content representations and discovers their most discriminative combination for the predefined actions. Both the weights and the MKL parameters are jointly learnt using only training and validation data.

7.4 Experiments

In this section, we present an evaluation, comparison, and analysis of the proposed Geometric and Appearance Relations of Pairwise Features.

The experiments are performed on 3 state-of-the-art action recognition datasets:

- KTH dataset,
- URADL dataset,
- HMDB51 dataset.

The selected datasets vary in the number of actions, types of actions, number of videos, and challenges.

The remainder of the section is organized as follows. In Section 7.4.1, we present implementation details of the proposed approach. In Section 7.4.2, we present experiments on the KTH dataset. Section 7.4.3 presents experiments on the URADL dataset. Section 7.4.4 presents experiments on the HMDB51 dataset. Finally, we present the summary and analysis of the results in Section 7.4.5.

7.4.1 Implementation Details

To reduce the computational cost, we limit the number of extracted pairwise features for each video sequence to $F_{MAX} = 10^5$ (using a random sampling), which is a good compromise between the amount of data extracted from a video sequence and the time required to create a codebook. Then, we create geometric codebooks of sizes $k_g \in \{2i\}_{i=1}^{4}$ and appearance codebooks of sizes $k_a \in \{2^i\}_{i=7}^{8}$ for the Fisher vector encoding and $k_a = 1000$ for the bag-of-features approach. We sort the extracted feature points according to the dimension t, y and then x; however, typically, the change of the sorting order does not affect significantly the results.

7.4.2 KTH Dataset

The KTH Action dataset (in short, the KTH dataset) is presented in Section 3.3.2.

There are two commonly used experimental setups to evaluate an approach on the KTH dataset: official splitting-based scheme and leave-one-person-out cross-validation evaluation scheme (see Section 3.3.2). We follow recent evaluations on this dataset [Wu 2011b, Wu 2011a, Jiang 2011, Wu 2014] using the leave-one-person-out cross-validation evaluation scheme. In general, it assesses the performance of an approach with much more reliability than splitting-based evaluation schemes, because it is much more comprehensive.

We evaluate the baseline approach, *i.e.* the Spatio-Temporal Interest Points, and the proposed Geometric and Appearance Relations of Pairwise Features approach. The following experiments are based on the bag-of-features approach.

Approach	s1	s2	s3	s4	**All**
Baseline	98.00%	91.33%	76.67%	94.67%	90.17%
Our Approach	98.67%	95.33%	93.20%	98.00%	**96.30%**

Table 7.1 – KTH dataset: Evaluation results of the Geometric and Appearance Relations of Pairwise Features and the baseline approach, *i.e.* the Spatio-Temporal Interest Points. Results are presented for each scenario individually and all scenarios together.

The KTH dataset contains videos recorded in four different scenarios. The evaluation results for each scenario individually and all together are presented in Table 7.1.

The baseline approach achieves 98% of mean class accuracy for scenario s1, 91.33% for scenario s2, 76.67% for scenario s3, 94.67% for scenario s4, and 90.17% overall for all scenarios.

The proposed GARPF approach achieves 98.67% of mean class accuracy for scenario s1, 95.33% for scenario s2, 93.20% for scenario s3, and 98% for scenario s4. Overall, the proposed approach achieves 96.30% of mean class accuracy.

The results clearly show that the GARPF representation enhances the discriminative power of local features and improves action recognition accuracy.

7.4.3 URADL Dataset

The University of Rochester Activities of Daily Living dataset (in short, the URADL dataset) is presented in Section 3.3.3.

We evaluate the baseline approach, *i.e*. the Spatio-Temporal Interest Points, and the proposed Geometric and Appearance Relations of Pairwise Features approach. The following experiments are based on the bag-of-features approach.

Approach	Accuracy
Baseline	90.67%
Our Approach	92%

Table 7.2 – URADL dataset: Evaluation results of the Geometric and Appearance Relations of Pairwise Features and the baseline approach, *i.e*. the Spatio-Temporal Interest Points.

The evaluation results are presented in Table 7.2. The baseline approach achieves 90.67% of mean class accuracy, and the proposed GARPF approach achieves 92% of mean class accuracy. The results confirm that the proposed GARPF representation enhances the discriminative power of local features and improves action recognition accuracy.

7.4.4 HMDB51 Dataset

The HMDB: A Large Human Motion Database dataset (in short, the HMDB51 dataset) is presented in Section 3.3.5.

We evaluate the baseline approach, *i.e*. the Spatio-Temporal Interest Points, and the proposed Geometric and Appearance Relations of Pairwise Features approach. The following experiments are based on the Fisher vector encoding.

The evaluation results are presented in Table 7.3. The baseline approach achieves 27.5% of mean class accuracy, and the proposed GARPF approach achieves 29.3% of mean class accuracy. The results confirm that the proposed GARPF representation enhances the discriminative power of local features and improves action recognition accuracy.

Approach	Accuracy
Baseline	27.5%
Our Approach	29.3%

Table 7.3 – HMDB51 dataset: Evaluation results of the Geometric and Appearance Relations of Pairwise Features and the baseline approach, *i.e*. the Spatio-Temporal Interest Points.

7.4.5 Results Summary and Analysis

Based on the experimental results on the KTH dataset (Section 7.4.2), URADL dataset (Section 7.4.3), and HMDB51 dataset (Section 7.4.4), we have shown that the Geometric and Appearance Relations of Pairwise Features (GARPF) representation enriches local feature representation and improves action recognition accuracy on all three datasets.

The GARPF representation captures relations among local features, what allows to encode correlations among features detected in different body parts. The use of the GARPF representation enhances the discriminative properties of a local feature encoding technique, therefore increasing the discriminative properties of a final video representation.

7.5 Conclusion

We have proposed new pairwise features for videos to encode relations among local spatio-temporal features. The new features are called the Geometric and Appearance Relations of Pairwise Features (GARPF). The main idea of the GARPF representation is to capture the appearance relations among features (both visual and motion), and use geometric information to describe how these appearance relations are mutually arranged in the spatio-temporal space.

We have applied the GARPF representation with the Spatio-Temporal Interest Points, and we have presented an evaluation of the proposed method on three popular action recognition dataset. The obtained results have shown that the GARPF representation improves action recognition accuracy in comparison to the baseline individual local features, *i.e*. the Spatio-Temporal Interest Points.

In future work, we intend to examine the GARPF representation with the popular Dense Trajectories.

CHAPTER 8

Spatio-Temporal Ordered Contextual Features

Contents

8.1 Introduction 186
8.2 Spatio-Temporal Ordered Contextual Features 187
8.2.1 Local Feature Quantization 187
8.2.2 Feature-Centric Neighbourhood 188
8.2.3 Spatio-Temporal Ordered Contextual Features 189
8.3 Approach Overview 191
8.4 Experiments 192
8.4.1 Implementation Details 193
8.4.2 KTH Dataset 193
8.4.3 URADL Dataset 194
8.4.4 Results Summary and Analysis 195
8.5 Conclusion 195

In this chapter, we introduce a new representation of contextual features for videos and we propose a new approach for action recognition based on the introduced features. The new representation is called the Spatio-Temporal Ordered Contextual Features (STOCF). The STOCF representation is based on quantized local spatio-temporal features. For each detected local feature, we define its neighbourhoods and we calculate statistics of pairwise co-occurring visual words within such neighbourhoods. Our representation captures not only local density of features, but also local pairwise relationships among the features and information about the space-time order of features. The STOCF representation enhances the discriminative power of local features incorporating feature-centric information about the local spatio-temporal distribution and order of local features. Then, we present an evaluation of the proposed STOCF features on two state-of-the-art action recognition datasets. We show that the STOCF representation improves action recognition accuracy in comparison to the local features.

8.1 Introduction

In Chapter 2 we present the most popular local feature encoding techniques and we discuss their limitations.

One of the main limitations of local features encoding techniques is that they simplify the structure of spatio-temporal video data assuming conditional independence across spatial and temporal domains. They compute only global statistics of local features, ignoring information about the spatio-temporal positions of features, relations among the features, and local densities of features. Thus, not using all the available information, the existing local feature encoding techniques may fail to distinguish similar actions.

In previous chapters we present the motivation behind using higher-level feature representations, and we propose two approaches that overcome the limitations of local feature encoding techniques, in particular:

- In Chapter 6 we propose the Relative Trajectory Shape descriptor, which introduces spatial information to a local feature encoding technique.
- In Chapter 7 we propose new Pairwise Features, which capture spatio-temporal relations among local features, and introduce this information to a local feature encoding technique.

In this chapter, we focus on another group of higher-level feature representations, *i.e.* we focus on the contextual features. A brief literature overview of the contextual features is presented in Section 2.3.2.2.

The contextual features capture relations among features in local neighbourhoods. They are very useful to distinguish videos with similar global distributions of local features, but with different local distributions of features.

The existing contextual features based techniques use the discriminative power of individual features, and they capture local densities of features in feature-centric neighborhoods. To capture structural information in feature-centric neighborhoods and the spatio-temporal order among features, they use the spatio-temporal grid approach; however, as mentioned before, the spatio-temporal grid is limited in terms of detailed description providing only a coarse representation.

Different from the existing techniques, we propose a new representation of contextual features, and we propose a new approach for action recognition based on the introduced features:

- Contextual Features: We propose a new representation of contextual features, called the Spatio-Temporal Ordered Contextual Features (STOCF). The STOCF represen-

tation is based on quantized local spatio-temporal features. For each detected local feature, we define its neighbourhoods and we calculate statistics of pairwise co-occurring visual words within such neighbourhoods. Our representation captures not only local density of features, but also local pairwise relationships among the features and information about the space-time order of features. The STOCF representation enhances the discriminative power of local features incorporating feature-centric information about the local spatio-temporal distribution and order of local features.

- Approach: We extract the Spatio-Temporal Interest Points in a video sequence. Then, we represent the context of each local feature by the proposed STOCF representation. Then, we apply the bag-of-features approach to represent videos. Finally, we use the Support Vector Machines for action classification.

- Experiments: We present an evaluation of our approach on two publicly available state-of-the-art datasets for human action recognition. We show that the proposed representation enhances the discriminative power of local features and improves action recognition accuracy.

The remainder of the chapter is organized as follows. In Section 8.2, we propose the Spatio-Temporal Ordered Contextual Features. Section 8.3 presents our action recognition framework. In Section 8.4, we present experimental results, comparison, and analysis. Finally, we conclude in Section 8.5.

8.2 Spatio-Temporal Ordered Contextual Features

In this section, we propose a new representation of contextual features for videos, called the Spatio-Temporal Ordered Contextual Features (STOCF). The STOCF representation is based on quantized local spatio-temporal features. For each detected local feature, we define its neighbourhoods and we calculate statistics of pairwise co-occurring visual words within such neighbourhoods. Our representation captures not only local density of features, but also local pairwise relationships among the features and information about the space-time order of features. The STOCF representation enhances the discriminative power of local features incorporating feature-centric information about the local spatio-temporal distribution and order of local features.

In Section 8.2.1 we discuss local feature extraction and quantization. Then, in Section 8.2.2 we define feature-centric neighbourhoods of detected local features. Finally, we propose the STOCF representation in Section 8.2.3.

8.2.1 Local Feature Quantization

Firstly, we extract local spatio-temporal features $\mathbb{P} = \{P_1, ..., P_n\}$ and their descriptors for each video sequence. A brief literature overview of the most popular local feature

detectors and descriptors is presented in Section 2.3.1.

Then, we cluster all the extracted descriptors from the training videos into k classes, called visual words, *e.g.* using the k-means algorithm. Finally, for each video sequence $\mathbb{V}$, we map the extracted local features to the closest visual words using associated local descriptors:

$$\mathbb{V} = \{(P_1, c_1), \ldots, (P_n, c_n)\}, \tag{8.1}$$

where P_i is the spatio-temporal position $P_i = [P_i^{(x)}, P_i^{(y)}, P_i^{(t)}]^T$ of the i-th local feature, and c_i is the index of the closest visual word for the local feature P_i.

8.2.2 Feature-Centric Neighbourhood

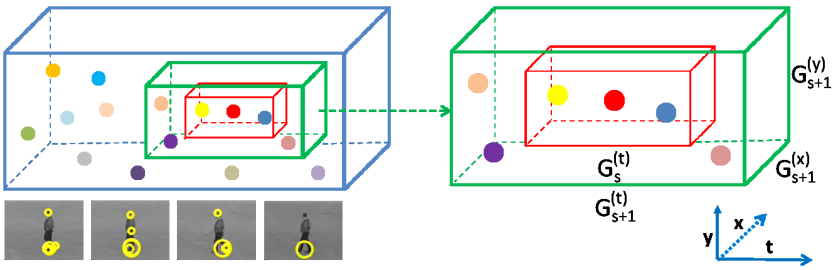


Figure 8.1 – Multi-scale feature-centric neighbourhoods used to calculate the Spatio-Temporal Ordered Contextual Features.

Once local features are extracted and assigned to the visual words, we define the neighbourhoods of the detected local features.

For each detected local feature P_i, we compute a set $\mathbb{S} = \{S_1, ..., S_{|S|}\}$ of multi-scale blocks around it. For simplicity, we define the s-th scale neighbourhood of the feature P_i as a cuboid with side lengths $G_s^{(x)}$, $G_s^{(y)}$, and $G_s^{(t)}$, see Figure 8.1. The points that belong to such s-th scale cuboid can be defined as:

$$\mathbb{N}_{i,s} = \{P_j \in \mathbb{P} : \bigcap_{d \in \{x,y,t\}} |P_j^{(d)} - P_i^{(d)}| \leq W_s^{(d)}\}, \tag{8.2}$$

where $\forall_{d \in \{x,y,t\}} G_s^{(d)} = 2W_s^{(d)} + 1$.

Section 8.2.1 defines that each local feature is assigned to a certain visual word v. Therefore, we define $\mathbb{N}_{i,s}^{(v)}$ as a set of points in the neighbourhood $\mathbb{N}_{i,s}$, which are assigned to the codebook element v:

$$\mathbb{N}_{i,s}^{(v)} = \{P_j \in \mathbb{N}_{i,s} : c_j = v\}, \tag{8.3}$$

where c_j is the index of the visual world assigned to the point P_j (see Equation 8.1).

8.2.3 Spatio-Temporal Ordered Contextual Features

Given extracted local spatio-temporal features and their neighbourhoods, we show how to compute the Spatio-Temporal Ordered Contextual Features (STOCF) for a local feature P_i and its neighbourhood $\mathbb{N}_{i,s}$.

We define the Spatio-Temporal Ordered Contextual Features (STOCF) as feature-centric statistics of local features within spatio-temporal video patches. The STOCF features are represented as histograms of pairwise co-occurring visual words. Each element of the histogram encodes information about the relationship between two visual words; the value x for a pair of visual words (C_a, C_b) means that there is x pairs of local features, where the first local feature is assigned to the visual word C_a, the second local feature is assigned to the visual word C_b, and that there is a spatio-temporal order of features, where the first local feature occurs before the second local feature. More precisely, we compute a non-negative matrix $\mathbb{M}_{i,s}^{(C)}$:

$$\mathbb{M}_{i,s}^{(C)} = \begin{bmatrix} \mathfrak{R}_{i,s}^{(1,1)} & \dots & \mathfrak{R}_{i,s}^{(1,k)} \\ \vdots & \ddots & \vdots \\ \mathfrak{R}_{i,s}^{(k,1)} & \dots & \mathfrak{R}_{i,s}^{(k,k)} \end{bmatrix}, \tag{8.4}$$

where $\mathfrak{R}_{i,s}^{(a,b)}$ is the cardinality of the set $\mathbb{R}_{i,s}^{(a,b)}$, *i.e.* $\mathfrak{R}_{i,s}^{(a,b)} = |\mathbb{R}_{i,s}^{(a,b)}|$, and $\mathbb{R}_{i,s}^{(a,b)}$ is the set of pairs of co-occurring local features, which are organized in space and time:

$$\mathbb{R}_{i,s}^{(a,b)} = \left\{ (P_j, P_k) \in (\mathbb{N}_{i,s}^{(a)} \times \mathbb{N}_{i,s}^{(b)}) : \left[\sum_{d \in \{x,y,t\}} w_d \mathrm{sgn}(P_k^{(d)} - P_j^{(d)}) \right] > 0 \right\}, \tag{8.5}$$

where w_d is the weight for the dimension d (explained in Section 8.4), and $\mathrm{sgn}(\cdot)$ is the signum function [1].

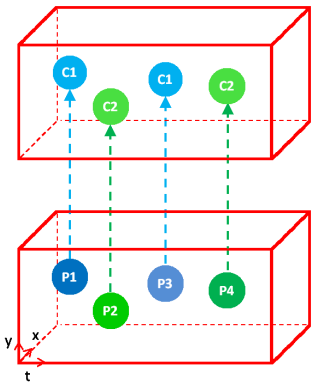
Finally, we define the Spatio-Temporal Ordered Contextual Features $STOCF_{i,s}$, calculated for the local feature P_i and the s-th scale neighbourhood $\mathbb{N}_{i,s}$, as a matrix $\mathbb{M}_{i,s}^{(C)}$ reshaped to a single dimensional vector:

$$STOCF_{i,s} = [\, \mathfrak{R}_{i,s}^{(1,1)}, \dots, \mathfrak{R}_{i,s}^{(1,k)}, \mathfrak{R}_{i,s}^{(2,1)}, \dots, \mathfrak{R}_{i,s}^{(k,k)} \,]^T. \tag{8.6}$$

If necessary, the size of the STOCF features can be reduced using *e.g.* Principal Component Analysis or Linear Discriminant Analysis. However, due to good results and efficiency of local features with small codebooks (see Section 3.4), methods for the dimensionality reduction were not applied during our experiments.

An example of the STOCF feature calculation process is presented in Figure 8.2.

[1] The signum function of a real number x is defined as follows: 1 if $x > 0$, 0 if $x = 0$, and -1 otherwise.



$$\mathbb{N}_{3,s} = \{P_1, P_2, P_3, P_4\},$$

$$\mathbb{C} = \{C_1, C_2\},$$

$$\mathbb{N}_{3,s}^{(C_1)} = \{P_1, P_3\},$$

$$\mathbb{N}_{3,s}^{(C_2)} = \{P_2, P_4\},$$

$$\mathbb{M}_{3,s}^{(F)} = \begin{array}{c} \\ P_1 \\ P_2 \\ P_3 \\ P_4 \end{array} \begin{array}{c} \begin{array}{cccc} P_1 & P_2 & P_3 & P_4 \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{array},$$

$$\mathbb{M}_{3,s}^{(C)} = \begin{array}{c} \\ C_1 \\ C_2 \end{array} \begin{array}{c} \begin{array}{cc} C_1 & C_2 \end{array} \\ \begin{bmatrix} 1 & 3 \\ 1 & 1 \end{bmatrix} \end{array},$$

$$STOCF_{3,s} = [1, 3, 1, 1]^T.$$

Figure 8.2 – STOCF representation: an example of the calculation process. The red cuboids represent a feature-centric s-th scale neighbourhood of a sample local feature P_3. $\mathbb{N}_{3,s}$ is the set of local features that belong to this neighbourhood. $\mathbb{C}$ represents the set of visual words, and $N_{3,s}^{(C_j)}$ is the set of local features that belong to the set $\mathbb{N}_{3,s}$ and that are assigned to the visual word C_j. The spatio-temporal order of local features is represented by the binary matrix $\mathbb{M}_{3,s}^{(F)}$, where $\mathbb{M}_{3,s}^{(F)}(P_a, P_b) = 1$ means that the local feature P_a occurs before the local feature P_b. The matrix $\mathbb{M}_{3,s}^{(C)}$ is obtained from corresponding points from the matrix $\mathbb{M}_{3,s}^{(F)}$ using point to codebook mapping. Finally, the STOCF representation is marked as the $STOCF_{3,s}$. Related elements of the matrices $\mathbb{M}_{3,s}^{(F)}$ and $\mathbb{M}_{3,s}^{(C)}$, and the vector $STOCF_{3,s}$ are indicated by identical colors.

The use of point to codebook mapping is very important for the STOCF representation, and it has two advantages:

- The STOCF representation is of equal size for all the videos. The number of extracted local features is different for each video sequence, but the size of the codebook is fixed.
- The size of the STOCF representation is smaller. Typically, the size of the codebook is smaller than the number of detected local features in a video sequence.

8.3 Approach Overview

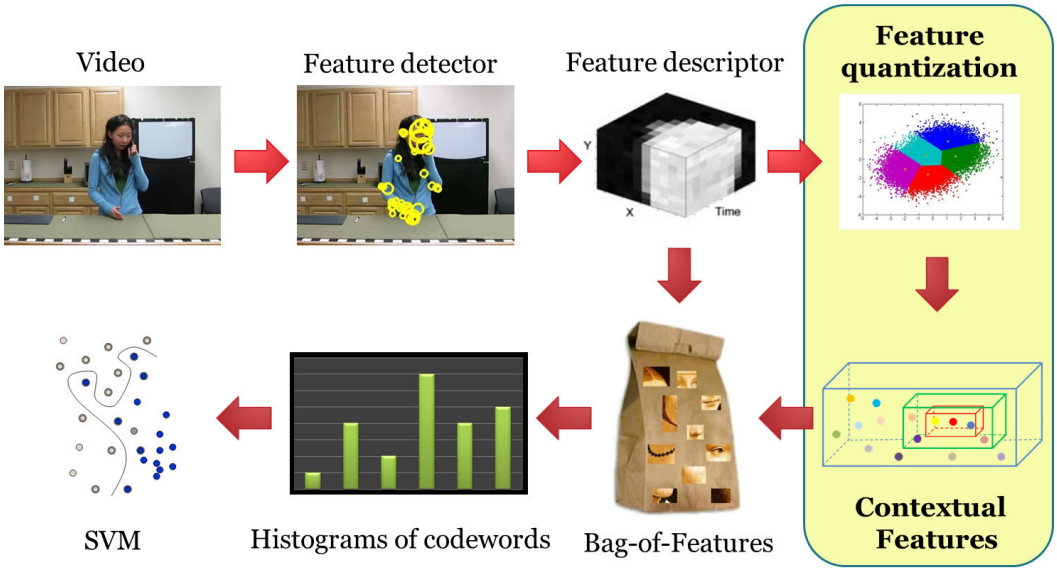


Figure 8.3 – Overview of the proposed action recognition framework.

In this section, we present our action recognition framework based on the introduced STOCF features.

The overview of the proposed approach is presented in Figure 8.3.

Similarly to the proposed approach in Chapter 7, in the first step of our approach, we extract local spatio-temporal features from video sequences. In order to do that, we use the Spatio-Temporal Interest Points proposed by [Laptev 2005] (see Section 3.2.1.1). Firstly, we apply the Harris3D corner detector to extract points of interest. We detect interest points in multiple spatial and temporal scales. Then, we describe the neighbourhood of each interest point by two descriptors: Histogram of Oriented Gradients and Histogram of Optical Flow. The detector and descriptors were selected based on their use in the literature and achieved high action recognition accuracy on various datasets (see Section 3.4). Due

to the large popularity of the Spatio-Temporal Interest Points, the selected detector and descriptors provide a good baseline for comparison with the state-of-the-art techniques. Note that our action recognition approach is independent of the type of detector and descriptor, and can be used together with any other algorithm extracting local features, *e.g.* with the Dense Trajectories proposed by Wang *et al.* [Wang 2011a] (see Section 3.2.1.2). However, the Harris3D detector is relatively sparse, what allows for fast calculation of the STOCF features.

Then, we represent the context of each local spatio-temporal feature by the proposed Spatio-Temporal Ordered Contextual Features (see Section 8.2), *i.e.* the local features are quantized, the feature-centric neighbourhoods are extracted, and the STOCF representations are calculated. Thus, each local feature is represented by three descriptors: HOG, HOF, and STOCF. The Harris3D detector is relatively sparse, what allows for fast calculation of the STOCF representations.

Once the descriptors are calculated, we apply the bag-of-words approach (see Section 3.2.2.1) for each feature class (HOG-HOF and STOCF) independently. We construct visual vocabularies from training videos clustering computed features using the k-means algorithm. To increase the precision, we initialize the k-means algorithm ten times and we keep the codebook with the lowest error. Then, we assign each feature to its closest visual world. The concatenated histograms of visual world occurrences over video form the final video representation.

Finally, we apply the Support Vector Machines to classify video representations into action categories (see Section 3.2.3). We use the Support Vector Machines with the exponential χ^2 kernel, and we use the one-vs-all approach for multi-class classification.

8.4 Experiments

In this section, we present an evaluation, comparison, and analysis of the proposed Spatio-Temporal Ordered Contextual Features.

The experiments are performed on 2 state-of-the-art action recognition datasets:

- KTH dataset,
- URADL dataset.

The selected datasets vary in the number of actions, types of actions, number of videos, and challenges.

The remainder of the section is organized as follows. In Section 8.4.1, we present implementation details of the proposed approach. In Section 8.4.2, we present experiments on the KTH dataset. Section 8.4.3 presents experiments on the URADL dataset. Finally, we present the summary and analysis of the results in Section 8.4.4.

8.4.1 Implementation Details

In order to quantize local features, we use the k-means clustering technique and the nearest neighbour algorithm. To compute the bag-of-features representation, features are quantized to the codebook size 1000, which has shown empirically to give good results. As a metric to calculate a distance between features and visual words, we use the L_2 norm. We set the weights w as $w^{(x)} = 1$, $w^{(y)} = 2$, $w^{(t)} = 4$. To compute STOCF features, the HOG-HOF descriptors are quantized to small codebook sizes (10, 15, 20, and 25), and feature-centric neighbourhoods are calculated for 8 different scales ($W_s^{(x)}$, $W_s^{(y)}$, $W_s^{(t)} \in \{4, 8, ..., 32\}$). The proper selection of neighbourhood size is important. Too small neighbourhood can contain only a few points and might not be discriminative. Too large volume may employ too many points and might also result in being not discriminative. Choosing an appropriate scale can be done in two ways: using Multiple Kernel Learning or cross-validation. In all our experiments, we calculate several codebooks to quantize local features (Section 8.2.1), and several multi-scale neighbourhoods to compute STOCF features. Then, we apply the cross-validation technique to both gauge the generalizability of the proposed approach, and select the most discriminative parameters. We use the leave-one-person-out cross-validation technique (see Section 3.2.4.1), where videos of one person are used as the validation data, and the remaining videos as the training data. This is done repeatedly so that videos of each person are used once as the validation data.

8.4.2 KTH Dataset

The KTH Action dataset (in short, the KTH dataset) is presented in Section 3.3.2.

There are two commonly used experimental setups to evaluate an approach on the KTH dataset: official splitting-based scheme and leave-one-person-out cross-validation evaluation scheme (see Section 3.3.2). We follow recent evaluations on this dataset [Wu 2011b, Wu 2011a, Jiang 2011, Wu 2014] using the leave-one-person-out cross-validation evaluation scheme. In general, it assesses the performance of an approach with much more reliability than splitting-based evaluation schemes, because it is much more comprehensive.

We evaluate the baseline approach, *i.e.* the Spatio-Temporal Interest Points, and the proposed Spatio-Temporal Ordered Contextual Features approach.

Approach	s1	s2	s3	s4	**All**
Baseline	98.00%	91.33%	76.67%	94.67%	90.17%
Our Approach	98.67%	95.33%	92.62%	98.00%	**96.16%**

Table 8.1 – KTH dataset: Evaluation results of the Spatio-Temporal Ordered Contextual Features and the baseline approach, *i.e.* the Spatio-Temporal Interest Points. Results are presented for each scenario individually and all scenarios together.

The KTH dataset contains videos recorded in four different scenarios. The evaluation results for each scenario individually and all together are presented in Table 8.1.

The baseline approach achieves 98% of mean class accuracy for scenario s1, 91.33% for scenario s2, 76.67% for scenario s3, 94.67% for scenario s4, and 90.17% overall for all scenarios. The proposed STOCF approach achieves 98.67% of mean class accuracy for scenario s1, 95.33% for scenario s2, 92.62% for scenario s3, and 98% for scenario s4, selecting the codebook size 15, 25, 10, and 10, respectively. Overall, the proposed approach achieves 96.16% of mean class accuracy. The results clearly show that the STOCF representation enhances the discriminative power of local features and improves action recognition accuracy.

	Neighbourhood		
Codebook	4	12	20
10	1.82 ms	3.09 ms	5.63 ms
20	1.87 ms	3.22 ms	6.36 ms

Table 8.2 – KTH dataset: Average computation time of the STOCF features using various precomputed codebooks and various neighbourhoods ($W^{(x)} = W^{(y)} = W^{(t)}$).

Moreover, we examine the average computation time of the STOCF features using various codebooks and neighbourhoods. Results are presented in Table 8.2. We observe that using small codebooks the STOCF features are very fast to calculate.

8.4.3 URADL Dataset

The University of Rochester Activities of Daily Living dataset (in short, the URADL dataset) is presented in Section 3.3.3.

We evaluate the baseline approach, *i.e.* the Spatio-Temporal Interest Points, and the proposed Spatio-Temporal Ordered Contextual Features approach.

Approach	Accuracy
Baseline	90.67%
Our Approach	93.33%

Table 8.3 – URADL dataset: Evaluation results of the Spatio-Temporal Ordered Contextual Features and the baseline approach, *i.e.* the Spatio-Temporal Interest Points.

The evaluation results are presented in Table 8.3. The baseline approach achieves 90.67% of mean class accuracy, and the proposed STOCF approach achieves 93.33% of

mean class accuracy. The results confirm that the proposed STOCF representation enhances the discriminative power of local features and improves action recognition accuracy.

8.4.4 Results Summary and Analysis

Based on the experimental results on the KTH dataset (Section 8.4.2) and the URADL dataset (Section 8.4.3), we have shown that the Spatio-Temporal Ordered Contextual Features representation enriches local feature representation and improves action recognition accuracy on both datasets.

The STOCF representation captures local density of features, and also local pairwise relationships among the features and information about the space-time order of features. The use of the STOCF representation enhances the discriminative properties of a local feature encoding technique, therefore increasing the discriminative properties of a final video representation.

8.5 Conclusion

We have proposed new contextual features for videos to capture statistics of space-time ordered local features. The new features are called the Spatio-Temporal Ordered Contextual Features (STOCF). The STOCF representation enhances the discriminative power of local features incorporating feature-centric information about the local spatio-temporal distribution and order of local features.

We have applied the STOCF representation with the Spatio-Temporal Interest Points. Using the bag-of-features approach and the Support Vector Machines, we have presented an evaluation of the STOCF features on two popular action recognition dataset. The obtained results have shown that the STOCF representation improves action recognition accuracy in comparison to the baseline individual local features, *i.e*. the Spatio-Temporal Interest Points.

In future work, we intend to extend the STOCF representation by adding information about the spatio-temporal relations and/or distances between local features. This can be achieved extending the 2-dimensional matrix $\mathbb{M}_{i,s}^{(C)}$ to the 3-dimensional matrix. Moreover, we intend to examine the STOCF representation with the popular Dense Trajectories.

CHAPTER 9

Comparison of Approaches

Contents

9.1 Spatio-Temporal Appearance Descriptors 198
9.2 Trajectory Shape Descriptors 199
9.3 Pairwise and Contextual Features 200
9.4 Comparison with State-of-The-Art 201
9.4.1 Weizmann Dataset 201
9.4.2 KTH Dataset 201
9.4.3 URADL Dataset 204
9.4.4 MSR Daily Activity 3D Dataset 204
9.4.5 HMDB51 Dataset 205
9.5 Conclusion 207

In this chapter, we present a comparison of the proposed techniques. Firstly, we compare spatio-temporal appearance descriptors (see Section 9.1), trajectory shape descriptors (see Section 9.2), and pairwise and contextual features (see Section 9.3). Then, we compare the proposed techniques with the state-of-the-art (see Section 9.4). Finally, we conclude and we recommend which technique to use depending on videos and actions (see Section 9.5).

Descriptor(s)	Weizmann	URADL	MSR	HMDB51
HOG	92.22	83.33	60.94	25.64
HOG3D [Klaser 2008, Shi 2013]	90.7	–	–	33.3
HOG + PCA	94	86.67	59.69	33.14
VCML	92.22	88	59.38	36.34
HOG + VCML	**94.44**	**92.67**	**63.44**	**40.52**
VBC	84.44	86.67	55	28.78
HOG + VBC	93.33	88.67	63.13	37.49

Table 9.1 – Comparison of local spatio-temporal appearance descriptors on: Weizmann, URADL, MSR Daily Activity 3D, and HMDB51 datasets.

9.1 Spatio-Temporal Appearance Descriptors

We present two local spatio-temporal video appearance descriptors:

- In Chapter 4 we propose the Video Covariance Matrix Logarithm (VCML) descriptor, which is based on a covariance matrix representation.
- In Chapter 5 we propose the Video Brownian Covariance (VBC) descriptor, which is based on a Brownian covariance.

The comparison of local spatio-temporal appearance descriptors with each other and with similar descriptors from the state-of-the-art is presented in Table 9.1. The table presents the best descriptors per dataset.

- The VCML descriptor outperforms (or achieves at least the same result as) the HOG descriptor in 3 out of 4 cases, the HOG3D in 2 out of 2 cases, and the VBC descriptor in 4 out of 4 cases.
- The VBC descriptor outperforms (or achieves at least the same result as) the HOG descriptor in 2 out of 4 cases.
- The fusion of HOG + VCML always outperforms the HOG, HOG3D, and HOG + PCA.
- The fusion of HOG + VBC always outperforms the HOG, HOG3D, and in 3 out of 4 cases the HOG + PCA.
- The fusion of HOG + VCML always outperforms all the other descriptors.

The results clearly show that the relations between pixel-level features are informative and useful for action recognition in videos. Moreover, the results confirm that the

HOG and the VCML/VBC descriptors are complementary to each other and outperform the state-of-the-art appearance descriptors. The former descriptor directly models pixel-level features and the latter descriptors model relations between pixel-level features. Moreover, the VCML descriptor works better than the VBC descriptor, and there are several possible reasons for that:

- Brownian covariance captures all kinds of possible relations between low-level features, and thus it may also capture more noise.
- Brownian covariance based representation may be too specific. In 15 out of 16 cases, the accuracy was improved by using the PCA, and it may be due to the fact that the PCA makes the VBC descriptor more generic.
- Information about linear relations between low-level features is more informative for action recognition than information about nonlinear relations.
- In contrast to the Brownian covariance, the classical covariance captures the direction of a linear relationship between low-level features (see Figure 5.1), and this information may be informative for action recognition.

Descriptor Normalization

In Section 5.2.5 we present the normalization of the VBC descriptor, and in Section 5.4.1 we present the influence of the normalization on the action recognition accuracy; in 15 out of 16 cases, the accuracy was improved by using the normalization.

Therefore, we apply the same normalization to the VCML descriptor (results are available in Chapter A); in 7 out of 16 cases, the accuracy was improved by using normalization, but 4 out of 7 cases were achieved on the Weizmann dataset.

In general, the normalization does not improve the accuracy of the VCML descriptor, so we use the proposed normalization only for the VBC descriptor.

9.2 Trajectory Shape Descriptors

In Chapter 6 we propose the Relative Trajectory Shape (RTS) descriptor, and we compare it with the Trajectory Shape descriptor (see Section 3.2.1.2). The comparison is presented in Chapter 6.

We summarize the main observations:

- The Relative Trajectory Shape (RTS) descriptor outperforms (or achieves at least the same result as) the Trajectory Shape (TS) descriptor in 4 out of 5 datasets.
- The fusion of TS + RTS always outperforms the TS and RTS descriptors on their own.

Descriptor(s)	KTH	URADL	HMDB51
STIPs	90.17	90.67	27.5
Matikainen *et al*. [Matikainen 2010]	–	70	–
Ta *et al*. [Ta 2010]	93	–	–
Ryoo *et al*. [Ryoo 2009]	93.8	–	–
Banerjee *et al*. [Banerjee 2011]	93.98	–	–
GARPF	96.30	92	29.3

Table 9.2 – Comparison of pairwise features on: KTH (with leave-one-person-out cross-validation evaluation scheme), URADL, and HMDB51 datasets.

The results clearly show that the trajectory information is informative and useful for action recognition in videos. Moreover, the results confirm that the TS and the RTS descriptors are complementary to each other, as the former directly models the shape of a trajectory and the latter models relations between a trajectory and a person.

9.3 Pairwise and Contextual Features

Moreover, we present two higher-level feature representations:

- In Chapter 7 we propose a new representation of pairwise features, called the Geometric and Appearance Relations of Pairwise Features (GARPF).

- In Chapter 8 we propose a new representation of contextual features, called the Spatio-Temporal Ordered Contextual Features (STOCF).

The comparison of pairwise features with the simple local features and the state-of-the-art is presented in Table 9.2, and the comparison of contextual features with the local features and the state-of-the-art is presented in Table 9.3. The main reason is that both the GARPF and STOCF representations improve the action recognition accuracy in comparison to simple local features (*i.e*. Spatio-Temporal Interest Points). Moreover, the GARPF and STOCF representations outperform or achieve very similar results to the top state-of-the-art pairwise features / contextual features based techniques. The STOCF representation achieves slightly better performance in comparison to the GARPF representation.

The results clearly show that the information about relations between features is informative and useful for action recognition in videos. Both the GARPF and the STOCF representations allow to describe correlations among features detected in different body parts (*e.g*. correlation between head-body center features or hand-hand features).

Descriptor(s)	KTH	URADL
STIPs	90.17	90.67
Wang *et al*. [Wang 2011b]	93.8	96
Oshin *et al*. [Oshin 2011]	94.1	89.3
Wu *et al*. [Wu 2011b]	94.5	–
Kovashka *et al*. [Kovashka 2010]	94.53	–
Gilbert *et al*. [Gilbert 2009]	96.7	–
STOCF	96.16	93.33

Table 9.3 – Comparison of contextual features on: KTH (with leave-one-person-out cross-validation evaluation scheme) and URADL datasets.

9.4 Comparison with State-of-The-Art

In this section we compare the proposed approaches with the state-of-the-art. The comparison is presented as follows.

9.4.1 Weizmann Dataset

The comparison of the proposed approaches with the state-of-the-art on the Weizmann dataset is presented in Table 9.4.

The VCML approach achieves 96.67% (3 videos were misclassified), and the VBC approach achieves 95.56%. Although other techniques get better results on this dataset, they are more specific and they may have problems to generalize to more complex datasets, *e.g*. Gorelick *et al*. [Gorelick 2007] regard human actions as three-dimensional shapes induced by the silhouettes in the space-time volume. The extraction of silhouettes in realistic and challenging datasets, such as HMDB51 dataset, is very difficult. Similarly, Fathi and Mori [Fathi 2008] use a background subtraction technique, which may fail on realistic and challenging datasets.

9.4.2 KTH Dataset

The comparison of the proposed approaches with the state-of-the-art on the KTH dataset (using leave-one-person-out cross-validation evaluation scheme) is presented in Table 9.5.

The Relative Trajectories approach achieves 97.16% of mean class accuracy and it outperforms the existing state-of-the-art techniques. The GARPF approach achieves 96.30% and the STOCF approach achieves 96.16%.

Approach	Chapter / Year	Accuracy (%)
Video Covariance Matrix Logarithm	4	96.67
Video Brownian Covariance	5	95.56
Niebles *et al.* [Niebles 2006]	2006	90
Klaser *et al.* [Klaser 2008]	2008	90.7
Ta *et al.* [Ta 2010]	2010	94.5
Bregonzio *et al.* [Bregonzio 2009]	2009	96.66
Banerjee *et al.* [Banerjee 2011]	2011	98.76
Gorelick *et al.* [Gorelick 2007]	2007	100
Fathi and Mori [Fathi 2008]	2008	100

Table 9.4 – Weizmann dataset: Comparison of the proposed approaches with the state-of-the-art.

Approach	Chapter / Year	Accuracy (%)
Relative Trajectories	6	97.16
Geometric and Appearance Relations of Pairwise Features	7	96.30
Spatio-Temporal Ordered Contextual Features	8	96.16
Liu *et al.* [Liu 2009]	2009	93.80
Ryoo *et al.* [Ryoo 2009]	2009	93.80
Wu *et al.* [Wu 2011b]	2011	94.50
Kim *et al.* [Kim 2007]	2007	95.33
Zhang *et al.* [Zhang 2012]	2012	95.50
Wu *et al.* [Wu 2011a]	2011	95.70
Jiang *et al.* [Jiang 2011]	2011	95.77
Wu *et al.* [Wu 2014]	2014	97.0

Table 9.5 – KTH dataset: Comparison of our approach with state-of-the-art methods in the literature using leave-one-person-out cross-validation evaluation scheme.

Approach	Chapter / Year	Accuracy (%)
Relative Trajectories	6	94.91
Laptev *et al*. [Laptev 2008]	2008	91.8
Sun *et al*. [Sun 2014]	2014	93.1
Yuan *et al*. [Yuan 2009b]	2009	93.3
Wang *et al*. [Wang 2011b]	2011	93.8
Zhang *et al*. [Zhang 2012]	2012	94.1
Wang *et al*. [Wang 2011a]	2011	94.2
Gilbert *et al*. [Gilbert 2011]	2011	94.5
Kovashka *et al*. [Kovashka 2010]	2010	94.53
Kaaniche *et al*. [Kaaniche 2010]	2012	94.67
Zhang *et al*. [Zhang 2014]	2014	94.8

Table 9.6 – KTH dataset: Comparison of Relative Trajectories with state-of-the-art methods in the literature using official splitting-based evaluation scheme.

Approach	Accuracy (%)				
	s1	s2	s3	s4	s1 - s4
Wu *et al*. [Wu 2011b]	96.7%	91.3%	93.3%	96.7%	94.5%
Jiang *et al*. [Jiang 2011]	98.83%	94.00%	94.78%	95.48%	95.77%
Relative Trajectories	99.33%	93.33%	97.32%	98.67%	97.16%

Table 9.7 – KTH dataset: Comparison of Relative Trajectories with state-of-the-art methods in the literature for each scenario separately using leave-one-person-out cross-validation evaluation scheme.

Then, we compare the best approach, *i.e*. the Relative Trajectories, with the state-of-the-art using the official splitting-based evaluation scheme. The results are presented in Table 9.6. The Relative Trajectories approach outperforms the state-of-the-art techniques again.

Moreover, we compare the Relative Trajectories with the state-of-the-art on each scenario independently and all together (using leave-one-person-out cross-validation evaluation scheme). The results are presented in Table 9.7. In 9 out of 10 cases, the Relative Trajectories approach outperforms the remaining techniques.

Approach	Chapter / Year	Acc. (%)
Video Covariance Matrix Logarithm	4	94
Video Brownian Covariance	5	93.33
Relative Trajectories	6	95.33
Geometric and Appearance Relations of Pairwise Features	7	92
Spatio-Temporal Ordered Contextual Features	8	93.33
Matikainen *et al.* [Matikainen 2010]	2010	70.0
Satkin *et al.* [Satkin 2010]	2010	80.00
Benabbas *et al.* [Benabbas 2010]	2010	81.00
Raptis *et al.* [Raptis 2010]	2010	82.67
Messing *et al.* [Messing 2009]	2009	89.00
Wang *et al.* [Wang 2011b]	2011	96.00

Table 9.8 – URADL dataset: Comparison of the proposed approaches with the state-of-the-art.

9.4.3 URADL Dataset

The comparison of the proposed approaches with the state-of-the-art on the URADL dataset is presented in Table 9.8.

The VCML approach achieves 94%, the VBC approach achieves 93.33%, the Relative Trajectories approach achieves 95.33%, the GARPF approach achieves 92%, and the STOCF approach achieves 93.33%. The Relative Trajectories approach outperforms all the other techniques proposed in this thesis, and the VCML approach achieves the second highest score.

Although Wang *et al.* [Wang 2011b] achieve better result on this dataset (96% vs. 95.33%), the difference is small, and it is not clear where the improvement comes from (the evaluation protocol is not clearly described). Moreover, they achieve lower result in comparison to our approach on the KTH dataset (93.8% vs. 94.91%).

9.4.4 MSR Daily Activity 3D Dataset

The comparison of the proposed approaches with the state-of-the-art on the MSR Daily Activity 3D dataset is presented in Table 9.9.

The VCML approach achieves 78.13%, the VBC approach achieves 76.56%, the Relative Trajectories approach achieves 77.19%, and the Relative Trajectories with Trajectory Filtering approach achieves 85.31%. The VCML and Relative Trajectories are again among the best proposed techniques.

Approach	Ch. / Year	Acc. (%)	Depth & Skeleton
Video Covariance Matrix Logarithm	4	78.13	Not Req.
Video Brownian Covariance	5	76.56	Not Req.
Relative Trajectories	6	77.19	Not Req.
Relative Trajectories with Filtering	6	85.31	Not Req.
Local Occupancy Pattern [Wang 2012]	2012	42.5	Required
Muller *et al.* [Müller 2006, Oreifej 2013]	2006	54	Required
Joint Position Features [Wang 2012]	2012	68	Required
Koperski *et al.* [Koperski 2014]	2014	72	Depth Req.
Fourier Temporal Pyramid [Wang 2012]	2012	78	Required
Oreifej *et al.* [Oreifej 2013]	2013	80	Required
Actionlet Ensemble [Wang 2012]	2012	85.75	Required

Table 9.9 – MSR Daily Activity 3D dataset: Comparison of the proposed approaches with the state-of-the-art.

The proposed methods are among the top approaches on the MSR Daily Activity 3D dataset. However, the main advantage of all our techniques is that neither depth nor skeleton is required by our approaches.

Note that we do not use the depth and the skeleton information to provide a general action recognition approach. Typically, the skeleton is very difficult to extract in realistic scenarios. The skeleton extraction is of sufficient quality on this dataset only because a person is facing the camera and it is close to the camera.

9.4.5 HMDB51 Dataset

The comparison of the proposed approaches with the state-of-the-art on the HMDB51 dataset is presented in Table 9.10.

The VCML approach achieves 52.85% and the VBC approach achieves 51.55%. Moreover, we evaluate the fusion of the VCML and the VBC approaches. The fusion of the VCML and the VBC decreases the accuracy on this dataset, possibly due to the redundancy of relations between low-level features.

The proposed VCML approach achieves the second highest score on this dataset. The top score is achieved by Wang *et al.* [Wang 2013b], who also use the Dense Trajectories and improve the quality of the extracted trajectories. The authors estimate the camera motion and remove trajectories consistent with it, and they use a human detector so as to improve the camera motion estimation. The human detector requires additional manual annotations for training, and it is not clear how generic and reproducible the proposed approach is.

Approach	Chapter / Year	Accuracy (%)
Video Covariance Matrix Logarithm (VCML)	4	52.85
Video Brownian Covariance (VBC)	5	51.55
VMCL + VBC	4, 5	52.07
Wang *et al.* [Wang 2013b]	2013	57.2
Jain *et al.* [Jain 2013]	2013	52.1
Shi *et al.* [Shi 2013]	2013	47.6
Kantorov *et al.* [Kantorov 2014]	2014	46.7
Wang *et al.* [Wang 2013a]	2013	46.6
Jiang *et al.* [Jiang 2012]	2012	40.7
Can *et al.* [Can 2013]	2013	39.0
Klaser *et al.* [Klaser 2008, Shi 2013]	2013	33.3
Klipper-Gross *et al.* [Kliper-Gross 2012]	2012	29.17
Solmaz *et al.* [Solmaz 2012]	2012	29.2
Sadanand *et al.* [Sadanand 2012]	2012	26.9
C2 [Kuehne 2011]	2011	22.83
HOG/HOF [Kuehne 2011]	2011	20.44

Table 9.10 – HMDB51 dataset: Comparison of the proposed approaches with the state-of-the-art.

By the use of the VCML descriptors we improve the accuracy of the Dense Trajectories from 47.02% up to 52.85%. Therefore, we believe that by the use of the VCML descriptors we can improve the accuracy of the Improved Dense Trajectories [Wang 2013b] as well, and then we can achieve the best score on this dataset. This is however out of the scope of this thesis, and it is planned for future work.

9.5 Conclusion

The proposed techniques obtain better or similar performance in comparison to the state-of-the-art on various human action recognition datasets. In general, the best results are achieved by the Relative Trajectories, and then by the Video Covariance Matrix Logarithm approach.

Our recommendation is to use the Relative Trajectories if we are able to estimate a head position in videos (*i.e.* if camera is static, people are visible, scene is not cluttered, and videos are of good quality). Otherwise, we propose to use the Video Covariance Matrix Logarithm approach.

The Video Covariance Matrix Logarithm, Video Brownian Covariance, and Relative Trajectories were proposed for the dense trajectories, which extract a large amount of features in a video sequence. Different from these techniques, the GARPF and STOCF representations were proposed for the sparse Spatio-Temporal Interest Points (there are too many relations to consider between the dense trajectories). The GARPF and STOCF representations capture relations among features, and the STOCF representation achieves slightly better performance in comparison to the GARPF representation. Therefore, if we are not able to extract and process large amount of features (such as the ones from the Dense Trajectories) we recommend to use the STOCF representation to reduce the processing time.

CHAPTER 10

Conclusion and Perspectives

Contents

10.1 Key Contributions 209
10.2 Limitations 210
10.3 Future Work 211
10.3.1 Short-Term Perspectives 211
10.3.2 Long-Term Perspectives 212

We have presented and evaluated several novel methods for human action recognition in videos. We have demonstrated that the proposed methods outperform the state-of-the-art on various and challenging datasets. We conclude our work pointing out the key contributions (Section 10.1) and their limitations (Section 10.2). Finally, we discuss future perspectives (Section 10.3), indicating interesting directions for future research in this field.

10.1 Key Contributions

Evaluation of local features and local feature encoding methods

We have reviewed, evaluated, and compared the most popular and the most prominent state-of-the-art techniques (local spatio-temporal features and local feature encoding techniques), and we have proposed our action recognition framework based on local features, which we use throughout this thesis work embedding the novel algorithms.

Two new local spatio-temporal descriptors

We have proposed two local spatio-temporal descriptors for videos (VCML and VBC). The first descriptor is based on a covariance matrix representation, and it models linear relations between low-level features. The second descriptor is based on a Brownian covariance, and it models all kinds of possible relations between low-level features.

Then, we have presented an extensive evaluation of the descriptors on four various datasets, and we have shown that the descriptors are complementary to the HOG descriptor (and the representation of the Dense Trajectories) as their fusion improves action recognition accuracy.

Three higher-level feature representations

Then, we have proposed three higher-level feature representations to go beyond the limitations of the local feature encoding techniques.

The first representation is based on the idea of relative dense trajectories. We have proposed an object-centric local feature representation of motion trajectories, which allows to use the spatial information by a local feature encoding technique.

The second representation encodes relations among local features as pairwise features. The main idea is to capture the appearance relations among features (both visual and motion), and use geometric information to describe how these appearance relations are mutually arranged in the spatio-temporal space.

The third representation captures statistics of pairwise co-occurring visual words within multi-scale feature-centric neighbourhoods. The proposed contextual features based representation encodes information about local density of features, local pairwise relations among the features, and spatio-temporal order among features.

We have presented an extensive evaluation of all the above feature representations on various datasets, and we have shown that the proposed feature representations improve action recognition accuracy.

CHU Nice Hospital dataset

We have proposed a new dataset for the recognition of realistic human actions of daily living (the CHU Nice Hospital dataset).

Evaluation and comparison

We have presented an extensive evaluation of the above techniques, and we have shown that the proposed methods obtain better or similar performance in comparison to the state-of-the-art on various, real, and challenging human action recognition datasets (Weizmann, KTH, URADL, MSR Daily Activity 3D, HMDB51, and CHU Nice Hospital).

10.2 Limitations

All the presented techniques assume videos and actions of sufficient length and motion, so as the Spatio-Temporal Interest Points / Dense Trajectories.

The proposed VCML and VBC descriptors can be applied to any action recognition dataset.

The main limitation of the Relative Trajectory approach is the requirement of the head position estimation, what could be difficult but possible on challenging datasets such as

HMDB51.

The main limitation of the proposed pairwise features (GARPF) and contextual features (STOCF) is the processing time. Therefore, the GARPF and STOCF representations were applied with the sparse Spatio-Temporal Interest Points, and not the Dense Trajectories, although the Dense Trajectories have shown superior results in comparison to the Spatio-Temporal Interest Points.

10.3 Future Work

10.3.1 Short-Term Perspectives

In short-term, we would like to investigate several possible improvements to the presented approaches. Moreover, we would like to present more experiments. The short-term perspective are as follows:

Evaluation

We would like to evaluate our approaches on other challenging datasets, such as UCF101 [Soomro 2012], Hollywood2 [Marszalek 2009], and TRECVID [Over 2014].

Convolutional Neural Networks (CNNs)

The CNNs have shown very good performance for image classification. Typically, the good performance of CNNs requires a large amount of training samples, and thus the CNNs are usually applied for images. Recently, a new action recognition dataset have been proposed, *i.e*. UCF101 [Soomro 2012] which contains 13320 videos. We would like to investigate the CNNs and the extracted features on this big action recognition datasets, and compare with our approaches.

Low-level motion features

We expect that the VCML and VBC descriptors achieve very good results also with motion features, such as optical flow and temporal gradient. It would be interesting to explore the accuracy of these descriptors with only motion features, and with appearance and motion features together. Moreover, the appearance and motion features could be used as a one or two sets of low-level features (to save the processing time, or two encode relations between appearance and motion features, respectively).

STOCF 3D

We intend to extend the STOCF representation by adding information about the spatio-temporal relations and/or distances between local features. This can be achieved extending the two-dimensional matrix to three-dimensional matrix representation.

Improved Dense Trajectories

We would like to evaluate the proposed techniques with the Improved Dense Trajectories, which have shown superior results in comparison to the Dense Trajectories. We intend to examine the STOCF and GARPF representations with the Improved Dense Trajectories as well.

Relative Trajectories with various Dynamic Coordinate Systems

We would like to investigate the Relative Trajectories using various dynamic coordinate systems (*e.g*. human body center), and using several dynamic coordinate systems at the same time, what could additionally enhance the discrminative power of trajectories.

A single "video frame representation" per each cell of the spatio-temporal grid

The VCML and VBC descriptors calculate a separate video frame descriptor for each video frame in each cell of the grid (see Section 4.2.3 and Section 5.2.4). It would be interesting to investigate the VCML and VBC descriptors calculating for each cell of the grid a single (Brownian) covariance based representation from all the low-level features extracted in all video frames of the cell. Additionally, the temporal position of features could be added to the list of low-level features.

A whole video sequence representation

Moreover, we would like to represent a whole video sequence using a (Brownian) covariance representation and local spatio-temporal features (such as the Spatio-Temporal Interest Points or the Dense Trajectories), as such representation may be complementary to the bag-of-features / Fisher vector based representation.

Affine-Invariant Riemannian Metric

The Log-Euclidean Riemannian Metric allows for fast and easy use of covariance based descriptors with local feature encoding techniques. However, this metric approximates the distance between two covariance metrics. Therefore, we would like to apply our descriptors with more accurate (but slower) metric, *i.e*. the Affine-Invariant Riemannian Metric (see Section 4.2.4.1).

10.3.2 Long-Term Perspectives

Generality: cross-dataset action recognition

Most of the existing techniques rely on the availability of the training data. When being required to recognize actions in a different dataset, they have to re-train the action recognition model. Firstly, we would like to evaluate the performance of our techniques on the cross-dataset action recognition task. Then, our next goal is to propose an approach which reduces the requirement of training labels and is able to handle the cross-dataset action recognition with just a few or no extra training labels.

Real time action localization and prediction

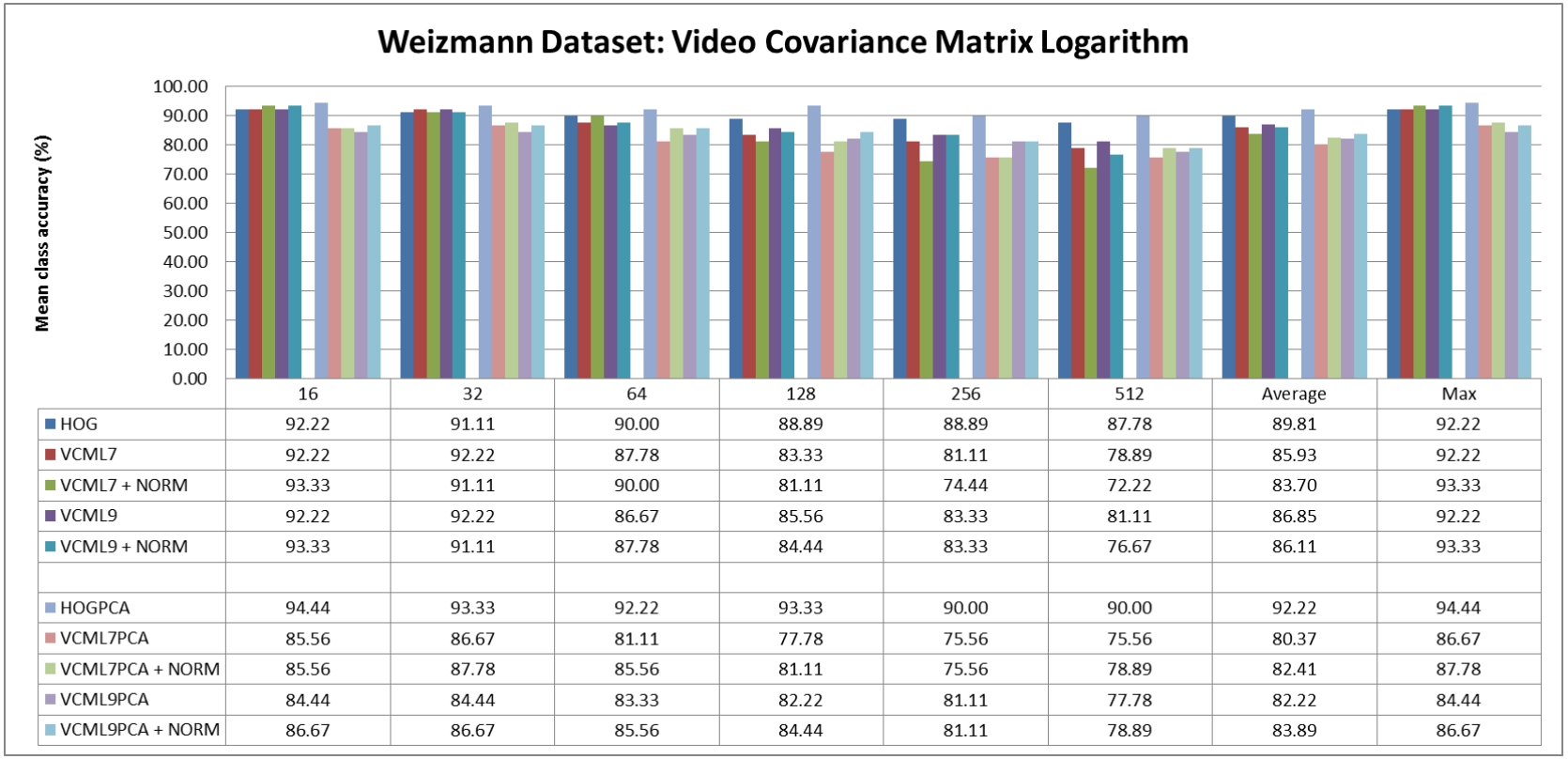
Our next goal is to localize human actions both in space and in time, and we would like to do it in real time. Moreover, we would like to predict what other people are going to do next. Research shows that humans are able to predict what will happen in the future based on the observations, and our goal is to create a system which will do the same.

Distinction between similar actions

The state-of-the-art techniques achieve very good performance in discriminating rather different action categories. Our next goal is the distinction between very similar actions, and even the distinction between people performing the same action. The CHU Nice Hospital dataset contains videos of people that are classified into healthy controls, mild cognitive impairments, and Alzheimer patients. By analyzing the way actions are performed, we would like to recognize actions and discriminate people performing actions. Our goal is to perform as well as doctors and we would like to compare the performance of automatic detection with medical diagnosis.

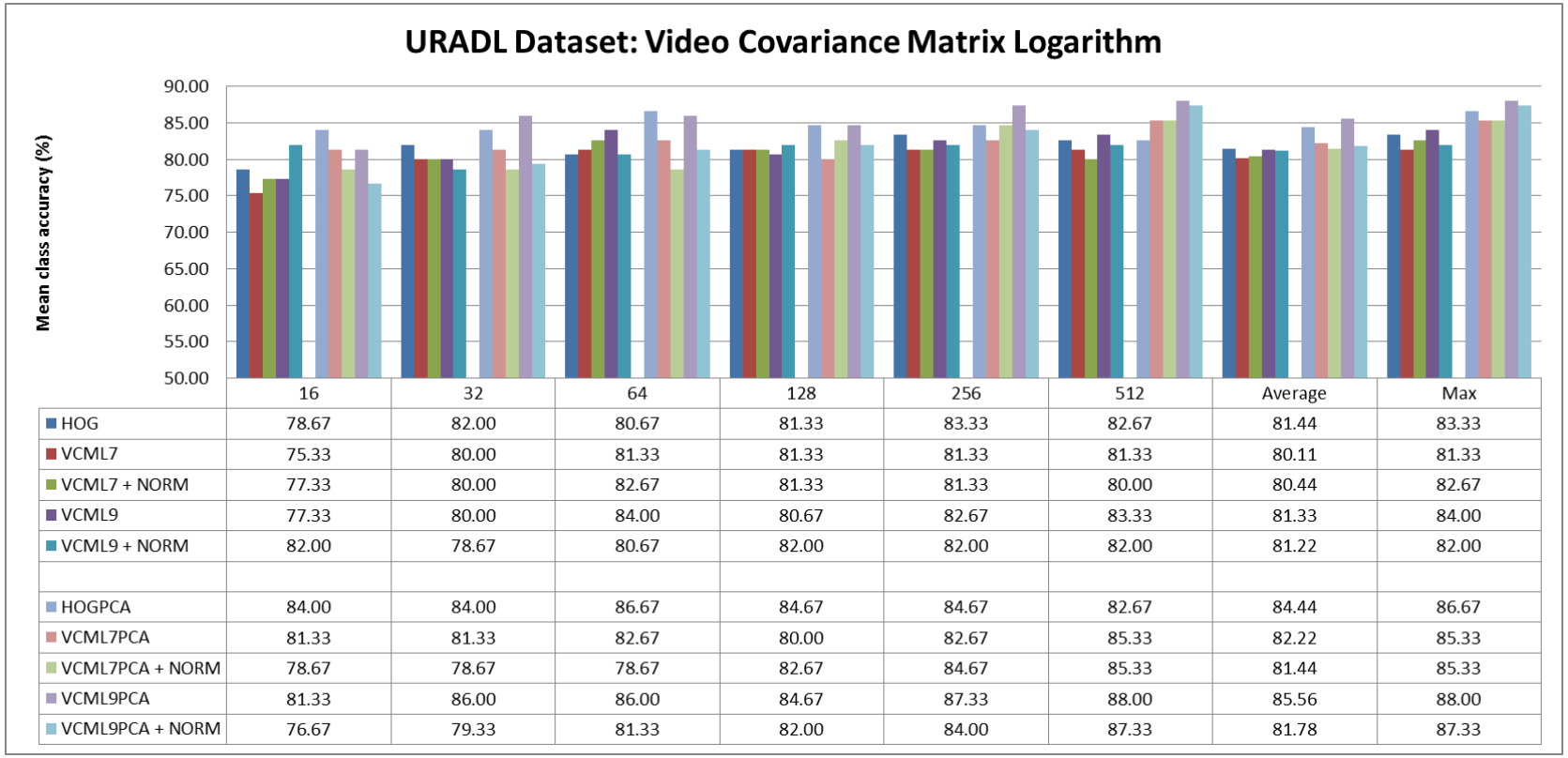
APPENDIX A

Video Covariance Matrix Logarithm: Additional Experiments



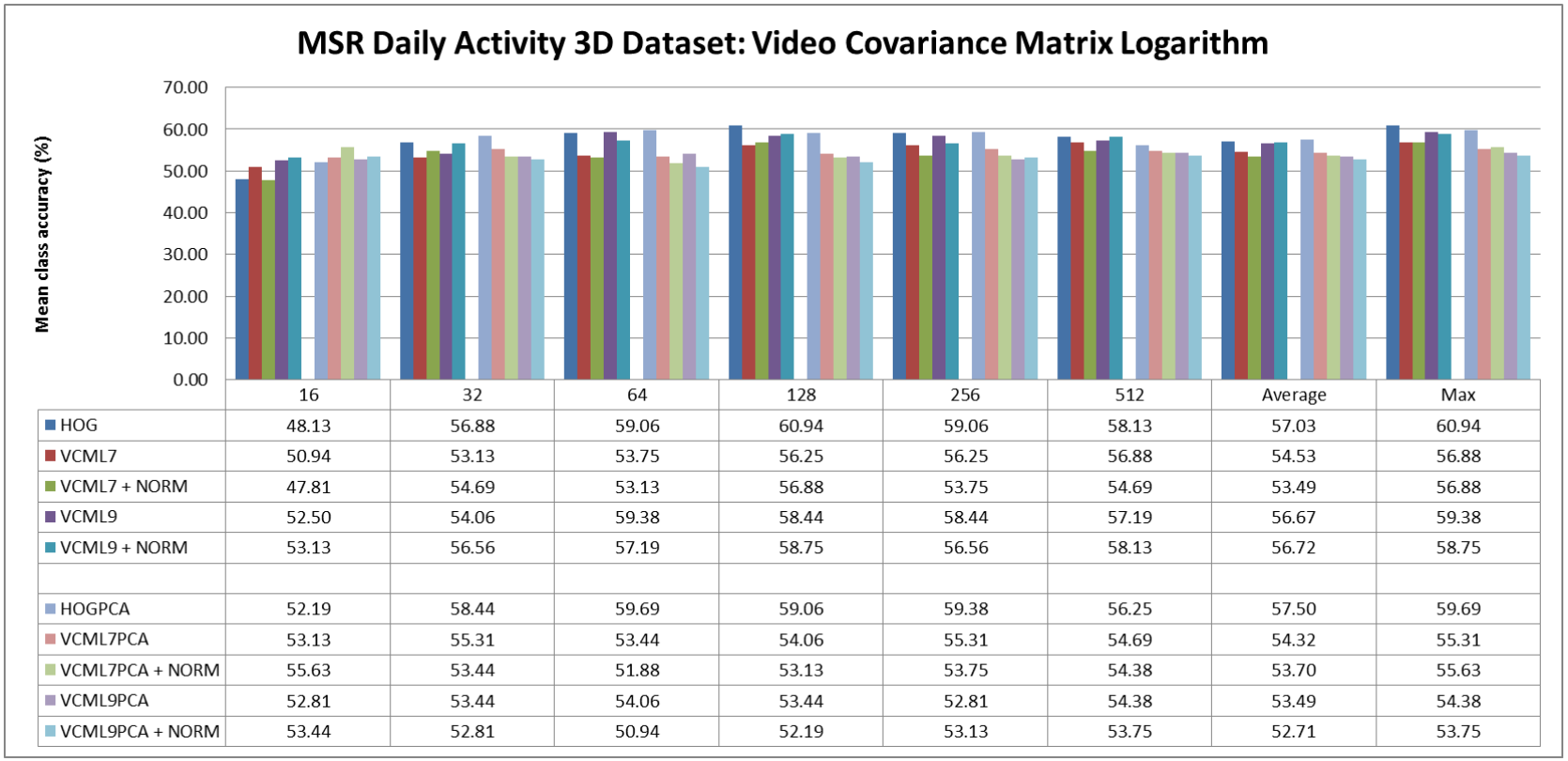
	16	32	64	128	256	512	Average	Max
HOG	92.22	91.11	90.00	88.89	88.89	87.78	89.81	92.22
VCML7	92.22	92.22	87.78	83.33	81.11	78.89	85.93	92.22
VCML7 + NORM	93.33	91.11	90.00	81.11	74.44	72.22	83.70	93.33
VCML9	92.22	92.22	86.67	85.56	83.33	81.11	86.85	92.22
VCML9 + NORM	93.33	91.11	87.78	84.44	83.33	76.67	86.11	93.33
HOGPCA	94.44	93.33	92.22	93.33	90.00	90.00	92.22	94.44
VCML7PCA	85.56	86.67	81.11	77.78	75.56	75.56	80.37	86.67
VCML7PCA + NORM	85.56	87.78	85.56	81.11	75.56	78.89	82.41	87.78
VCML9PCA	84.44	84.44	83.33	82.22	81.11	77.78	82.22	84.44
VCML9PCA + NORM	86.67	86.67	85.56	84.44	81.11	78.89	83.89	86.67

Figure A.1 – Weizmann dataset: Evaluation results of Video Covariance Matrix Logarithm descriptors. The plot presents the mean class accuracy of descriptors with respect to the codebook size (the “x” axis).



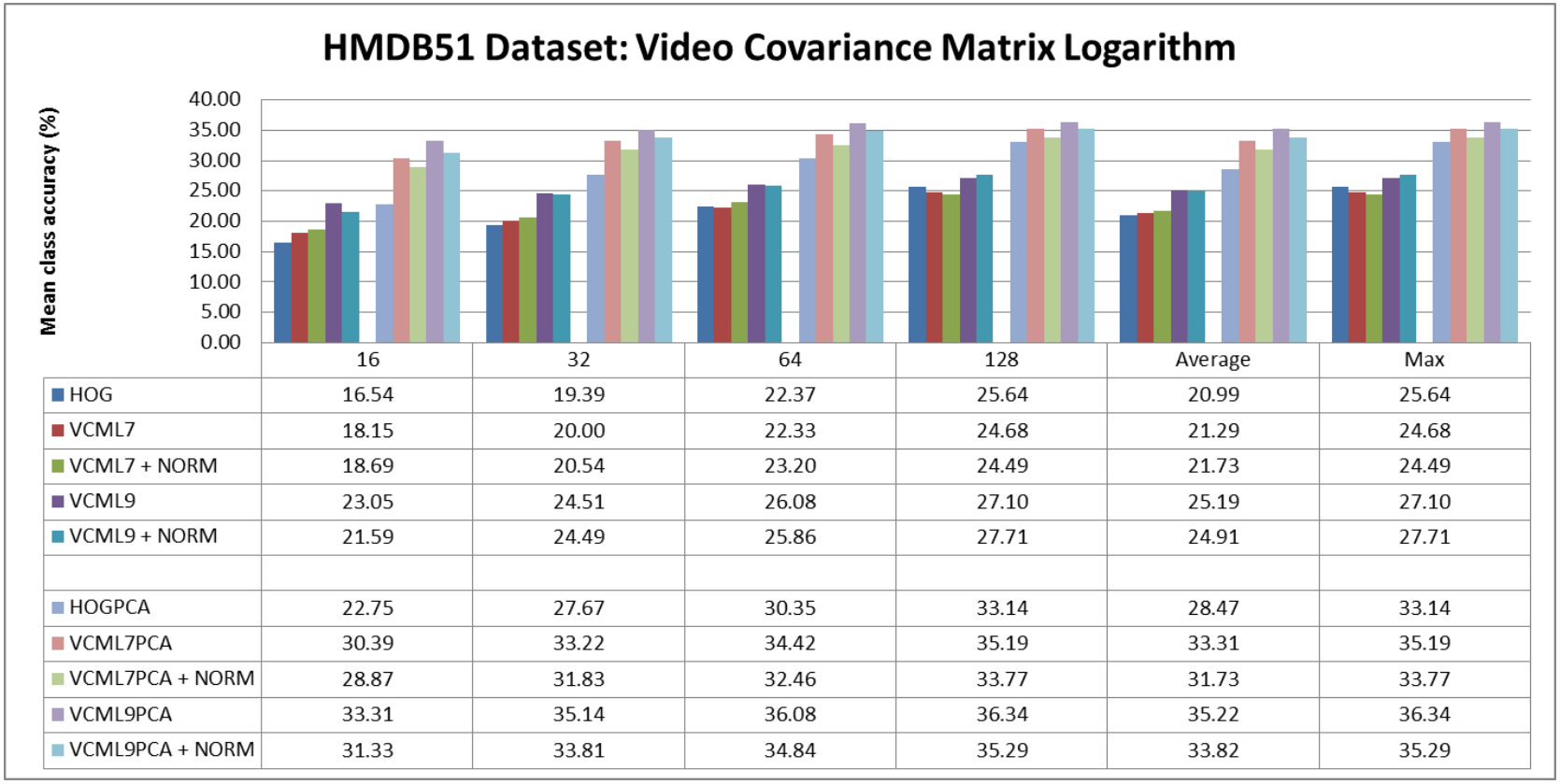
	16	32	64	128	256	512	Average	Max
HOG	78.67	82.00	80.67	81.33	83.33	82.67	81.44	83.33
VCML7	75.33	80.00	81.33	81.33	81.33	81.33	80.11	81.33
VCML7 + NORM	77.33	80.00	82.67	81.33	81.33	80.00	80.44	82.67
VCML9	77.33	80.00	84.00	80.67	82.67	83.33	81.33	84.00
VCML9 + NORM	82.00	78.67	80.67	82.00	82.00	82.00	81.22	82.00
HOGPCA	84.00	84.00	86.67	84.67	84.67	82.67	84.44	86.67
VCML7PCA	81.33	81.33	82.67	80.00	82.67	85.33	82.22	85.33
VCML7PCA + NORM	78.67	78.67	78.67	82.67	84.67	85.33	81.44	85.33
VCML9PCA	81.33	86.00	86.00	84.67	87.33	88.00	85.56	88.00
VCML9PCA + NORM	76.67	79.33	81.33	82.00	84.00	87.33	81.78	87.33

Figure A.2 – URADL dataset: Evaluation results of Video Covariance Matrix Logarithm descriptors. The plot presents the mean class accuracy of descriptors with respect to the codebook size (the “x” axis).



	16	32	64	128	256	512	Average	Max
HOG	48.13	56.88	59.06	60.94	59.06	58.13	57.03	60.94
VCML7	50.94	53.13	53.75	56.25	56.25	56.88	54.53	56.88
VCML7 + NORM	47.81	54.69	53.13	56.88	53.75	54.69	53.49	56.88
VCML9	52.50	54.06	59.38	58.44	58.44	57.19	56.67	59.38
VCML9 + NORM	53.13	56.56	57.19	58.75	56.56	58.13	56.72	58.75
HOGPCA	52.19	58.44	59.69	59.06	59.38	56.25	57.50	59.69
VCML7PCA	53.13	55.31	53.44	54.06	55.31	54.69	54.32	55.31
VCML7PCA + NORM	55.63	53.44	51.88	53.13	53.75	54.38	53.70	55.63
VCML9PCA	52.81	53.44	54.06	53.44	52.81	54.38	53.49	54.38
VCML9PCA + NORM	53.44	52.81	50.94	52.19	53.13	53.75	52.71	53.75

Figure A.3 – MSR Daily Activity 3D dataset: Evaluation results of Video Covariance Matrix Logarithm descriptors. The plot presents the mean class accuracy of descriptors with respect to the codebook size (the “x” axis).



	16	32	64	128	Average	Max
HOG	16.54	19.39	22.37	25.64	20.99	25.64
VCML7	18.15	20.00	22.33	24.68	21.29	24.68
VCML7 + NORM	18.69	20.54	23.20	24.49	21.73	24.49
VCML9	23.05	24.51	26.08	27.10	25.19	27.10
VCML9 + NORM	21.59	24.49	25.86	27.71	24.91	27.71
HOGPCA	22.75	27.67	30.35	33.14	28.47	33.14
VCML7PCA	30.39	33.22	34.42	35.19	33.31	35.19
VCML7PCA + NORM	28.87	31.83	32.46	33.77	31.73	33.77
VCML9PCA	33.31	35.14	36.08	36.34	35.22	36.34
VCML9PCA + NORM	31.33	33.81	34.84	35.29	33.82	35.29

Figure A.4 – HMDB51 dataset: Evaluation results of Video Covariance Matrix Logarithm descriptors. The plot presents the mean class accuracy of descriptors with respect to the codebook size (the "x" axis).

APPENDIX B

CHU Nice Hospital Dataset: Sample Video Frames

Figure B.1 – CHU dataset: 6 sample video frames for playing cards action.

Figure B.2 – CHU dataset: 6 sample video frames for reading action.

Figure B.3 – CHU dataset: 6 sample video frames for matching ABCD sheets of paper action.

Figure B.4 – CHU dataset: 6 sample video frames for sitting down and standing up action.

Figure B.5 – CHU dataset: 6 sample video frames for turning back action.

Figure B.6 – CHU dataset: 6 sample video frames for standing up and moving ahead action.

Figure B.7 – CHU dataset: 6 sample video frames for walking1 action.

Figure B.8 – CHU dataset: 6 sample video frames for walking2 action.

Bibliography

[Aggarwal 2011] JK Aggarwal and Michael S Ryoo. *Human activity analysis: A review*. ACM Computing Surveys (CSUR), vol. 43, no. 3, page 16, 2011. (Cited on page 17.)

[Ali 2007] Saad Ali, Arslan Basharat and Mubarak Shah. *Chaotic invariants for human action recognition*. In Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on, pages 1–8. IEEE, 2007. (Cited on pages 18 and 19.)

[Andriluka 2008] Mykhaylo Andriluka, Stefan Roth and Bernt Schiele. *People-tracking-by-detection and people-detection-by-tracking*. In Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, pages 1–8. IEEE, 2008. (Cited on page 162.)

[Arsigny 2006] Vincent Arsigny, Pierre Fillard, Xavier Pennec and Nicholas Ayache. *Log-Euclidean metrics for fast and simple calculus on diffusion tensors*. Magnetic resonance in medicine, vol. 56, no. 2, pages 411–421, 2006. (Cited on pages 105 and 110.)

[Avila 2011] Sandra Avila, Nicolas Thome, Matthieu Cord, Eduardo Valle and A de A Araujo. *Bossa: Extended bow formalism for image classification*. In Image Processing (ICIP), 2011 18th IEEE International Conference on, pages 2909–2912. IEEE, 2011. (Cited on page 38.)

[Avila 2013] S Avila, N Thome, M Cord, E Valle and AA de Araújo. *Pooling in image representation: the visual codeword point of view*. Computer Vision and Image Understanding, vol. 117, no. 5, pages 453–465, 2013. (Cited on pages 36, 37, 38 and 56.)

[Bak 2012a] Slawomir Bak, Guillaume Charpiat, Etienne Corvee, Francois Bremond and Monique Thonnat. *Learning to match appearances by correlations in a covariance metric space*. In Computer Vision–ECCV 2012, pages 806–820. Springer Berlin Heidelberg, 2012. (Cited on page 110.)

[Bak 2012b] Slawomir Bak, Etienne Corvee, Francois Bremond and Monique Thonnat. *Boosted human re-identification using riemannian manifolds*. Image and Vision Computing, vol. 30, no. 6, pages 443–452, 2012. (Cited on page 100.)

[Bak 2013] Sławomir Bak, Ratnesh Kumar, François Bremond*et al. Brownian descriptor: a Rich Meta-Feature for Appearance Matching*. In WACV: Winter Conference on Applications of Computer Vision, 2013. (Cited on page 130.)

[Banerjee 2011] Prithviraj Banerjee and Ram Nevatia. *Learning neighborhood cooccurrence statistics of sparse features for human activity recognition*. In Advanced Video and Signal-Based Surveillance (AVSS), 2011 8th IEEE International Conference on, pages 212–217. IEEE, 2011. (Cited on pages 32, 33, 200 and 202.)

[Bay 2006] Herbert Bay, Tinne Tuytelaars and Luc Van Gool. *Surf: Speeded up robust features*. pages 404–417. Springer, 2006. (Cited on page 29.)

[Beaudet 1978] P. R. Beaudet. *Rotationally invariant image operators*. In Proceedings of the 4th International Joint Conference on Pattern Recognition, pages 579–583, Kyoto, Japan, November 1978. (Cited on page 24.)

[Benabbas 2010] Yassine Benabbas, Adel Lablack, Nacim Ihaddadene and Chabane Djeraba. *Action recognition using direction models of motion*. In Pattern Recognition (ICPR), 2010 20th International Conference on, pages 4295–4298. IEEE, 2010. (Cited on page 204.)

[Bilinski 2011] Piotr Bilinski and François Bremond. *Evaluation of local descriptors for action recognition in videos*. In International Conference on Computer Vision Systems, Sophia Antipolis, France, September 2011. (Cited on pages 12 and 48.)

[Bilinski 2012a] Piotr Bilinski and François Bremond. *Contextual Statistics of Space-Time Ordered Features for Human Action Recognition*. In AVSS, 2012. (Cited on page 14.)

[Bilinski 2012b] Piotr Bilinski and Francois Bremond. *Statistics of Pairwise Co-occurring Local Spatio-Temporal Features for Human Action Recognition*. In Computer Vision–ECCV 2012. Workshops and Demonstrations, pages 311–320. Springer Berlin Heidelberg, 2012. (Cited on page 13.)

[Bilinski 2013] Piotr Bilinski, Etienne Corvee, Slawomir Bak and Francois Bremond. *Relative Dense Tracklets for Human Action Recognition*. In 10th IEEE International Conference on Automatic Face and Gesture Recognition, pages 1–7, Shanghai, Chine, April 2013. IEEE. (Cited on page 13.)

[Bilinski 2014] Piotr Bilinski, Michal Koperski, Slawomir Bak and François Bremond. *Representing Visual Appearance by Video Brownian Covariance Descriptor for Human Action Recognition*. In AVSS - 11th IEEE International Conference on Advanced Video and Signal-Based Surveillance, Seoul, Corée, République De, August 2014. IEEE. (Cited on page 13.)

[Blank 2005] Moshe Blank, Lena Gorelick, Eli Shechtman, Michal Irani and Ronen Basri. *Actions as space-time shapes*. In Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on, volume 2, pages 1395–1402. IEEE, 2005. (Cited on pages 22, 42 and 66.)

[Blei 2003] David M Blei, Andrew Y Ng and Michael I Jordan. *Latent dirichlet allocation*. the Journal of machine Learning research, vol. 3, pages 993–1022, 2003. (Cited on page 39.)

[Bobick 2001] Aaron F. Bobick and James W. Davis. *The recognition of human movement using temporal templates*. Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 23, no. 3, pages 257–267, 2001. (Cited on page 22.)

[Bregonzio 2009] Matteo Bregonzio, Shaogang Gong and Tao Xiang. *Recognising action as clouds of space-time interest points*. In Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, pages 1948–1955. IEEE, 2009. (Cited on page 202.)

[Breiman 2001] Leo Breiman. *Random forests*. Machine learning, vol. 45, no. 1, pages 5–32, 2001. (Cited on page 41.)

[Burges 1998] Christopher J. C. Burges. *A Tutorial on Support Vector Machines for Pattern Recognition*. Data Min. Knowl. Discov., vol. 2, no. 2, pages 121–167, 1998. (Cited on page 41.)

[Can 2013] Ethem Can and R. Manmatha. *Formulating Action Recognition as a Ranking Problem*. In CVPR Workshop ACTS, pages 251–256, June 2013. (Cited on page 206.)

[Chang 2011] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: A library for support vector machines*. ACM Transactions on Intelligent Systems and Technology, vol. 2, pages 27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`. (Cited on page 62.)

[Chatfield 2011] K. Chatfield, V. Lempitsky, A. Vedaldi and A. Zisserman. *The devil is in the details: an evaluation of recent feature encoding methods*. In British Machine Vision Conference, 2011. (Cited on pages 36, 38 and 56.)

[Cinbis 2013] Ramazan Gokberk Cinbis, Jakob Verbeek and Cordelia Schmid. *Segmentation Driven Object Detection with Fisher Vectors*. In ICCV 2013 - IEEE International Conference on Computer Vision, Sydney, Australie, December 2013. IEEE. (Cited on page 60.)

[Cortes 1995] Corinna Cortes and Vladimir Vapnik. *Support-Vector Networks*. Machine Learning, vol. 20, no. 3, pages 273–297, September 1995. (Cited on pages 12 and 60.)

[Cover 2006] T. Cover and P. Hart. *Nearest Neighbor Pattern Classification*. IEEE Trans. Inf. Theor., vol. 13, no. 1, pages 21–27, September 2006. (Cited on page 40.)

[Cristianini 2010] Nello Cristianini and John Shawe-Taylor. An introduction to support vector machines and other kernel-based learning methods. Cambridge University Press, 2010. (Cited on page 41.)

[Crow 1984] Franklin C. Crow. *Summed-area tables for texture mapping*. In SIGGRAPH '84: Proceedings of the 11th annual conference on Computer graphics and interactive techniques, pages 207–212, New York, NY, USA, 1984. ACM. (Cited on page 112.)

[Csurka 2004] Gabriella Csurka, Christopher R. Dance, Lixin Fan, Jutta Willamowski and CĂŠdric Bray. *Visual categorization with bags of keypoints*. In In Workshop

on Statistical Learning in Computer Vision, ECCV, pages 1–22, 2004. (Cited on page 36.)

[Cula 2001] O.G. Cula and K.J. Dana. *Compact representation of bidirectional texture functions*. In Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, volume 1, pages I–1041–I–1047 vol.1, 2001. (Cited on page 36.)

[Dalal 2005] Navneet Dalal and Bill Triggs. *Histograms of oriented gradients for human detection*. In Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, volume 1, pages 886–893. IEEE, 2005. (Cited on pages 28, 51 and 158.)

[Dalal 2006] Navneet Dalal, Bill Triggs and Cordelia Schmid. *Human Detection Using Oriented Histograms of Flow and Appearance*. In ECCV, pages 428–441, 2006. (Cited on pages 29 and 55.)

[Deerwester 1990] Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas and Richard A. Harshman. *Indexing by Latent Semantic Analysis*. JASIS, vol. 41, no. 6, pages 391–407, 1990. (Cited on page 39.)

[Dietterich 2000] ThomasG. Dietterich. *An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization*. Machine Learning, vol. 40, no. 2, pages 139–157, 2000. (Cited on page 42.)

[Dollar 2005] P. Dollar, V. Rabaud, G. Cottrell and S. Belongie. *Behavior Recognition via Sparse Spatio-temporal Features*. In Proceedings of the 14th International Conference on Computer Communications and Networks, ICCCN '05, pages 65–72, Washington, DC, USA, 2005. IEEE Computer Society. (Cited on pages 24, 28, 30, 32, 36, 42 and 101.)

[Efros 2003] A. A. Efros, A. C. Berg, G. Mori and J. Malik. *Recognizing Action at a Distance*. In ICCV, 2003. (Cited on page 42.)

[Farnebäck 2003] Gunnar Farnebäck. *Two-Frame Motion Estimation Based on Polynomial Expansion*. In Proceedings of the 13th Scandinavian Conference on Image Analysis, LNCS 2749, pages 363–370, Gothenburg, Sweden, June-July 2003. (Cited on page 54.)

[Fathi 2008] Alireza Fathi and Greg Mori. *Action recognition by learning mid-level motion features*. In In CVPR, 2008. (Cited on pages 42, 201 and 202.)

[Ferrari 2008] V. Ferrari, M. Marin-Jimenez and A. Zisserman. *Progressive Search Space Reduction for Human Pose Estimation*. In CVPR, 2008. (Cited on page 158.)

[Förstner 1999] Wolfgang Förstner and Boudewijn Moonen. *A Metric for Covariance Matrices*. In F. Krumm and V. S. Schwarze, editeurs, Festschrift for Erik W. Grafarend

on the occasion of his 60th birthday. Also appeared in: Geodesy - The Challenge of the 3rd Millennium (2003, with editors Professor Dr. Erik W. Grafarend, Dr. Friedrich W. Krumm,Dr. Volker S. Schwarze, ISBN: 978-3-642-07733-3 (Print) 978-3-662-05296-9 (Online)), pages 113–128, 1999. (Cited on page 109.)

[Gemert 2008] Jan C. Gemert, Jan-Mark Geusebroek, Cor J. Veenman and Arnold W. Smeulders. *Kernel Codebooks for Scene Categorization*. In Proceedings of the 10th European Conference on Computer Vision: Part III, ECCV '08, pages 696–709, Berlin, Heidelberg, 2008. Springer-Verlag. (Cited on page 36.)

[Genuer 2008] Robin Genuer, Jean-Michel Poggi and Christine Tuleau. *Random Forests: some methodological insights*. Rapport de recherche RR-6729, INRIA, 2008. (Cited on page 41.)

[Gilbert 2009] Andrew Gilbert, John Illingworth and Richard Bowden. *Fast realistic multi-action recognition using mined dense spatio-temporal features.* In ICCV, 2009. (Cited on pages 35 and 201.)

[Gilbert 2011] A. Gilbert, J. Illingworth and R. Bowden. *Action Recognition using Mined Hierarchical Compound Features*. TPAMI, 2011. (Cited on page 203.)

[Gorelick 2007] Lena Gorelick, Moshe Blank, Eli Shechtman, Michal Irani and Ronen Basri. *Actions as Space-Time Shapes*. Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 12, pages 2247–2253, December 2007. (Cited on pages 201 and 202.)

[Guo 2010a] Kai Guo, Prakash Ishwar and Janusz Konrad. *Action recognition in video by sparse representation on covariance manifolds of silhouette tunnels*. Recognizing Patterns in Signals, Speech, Images and Videos, pages 294–305, 2010. (Cited on pages 101 and 102.)

[Guo 2010b] Kai Guo, Prakash Ishwar and Janusz Konrad. *Action recognition using sparse representation on covariance manifolds of optical flow*. In Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on, pages 188–195. IEEE, 2010. (Cited on pages 101 and 102.)

[Guo 2013] Kai Guo, Prakash Ishwar and Janusz Konrad. *Action recognition from video using feature covariance matrices*. IEEE Transactions on Image Processing, vol. 22, no. 6, pages 2479–2494, 2013. (Cited on pages 101 and 102.)

[Guo 2014] Guodong Guo and Alice Lai. *A survey on still image based human action recognition*. Pattern Recognition, vol. 47, no. 10, pages 3343 – 3361, 2014. (Cited on page 100.)

[Harris 1988] Chris Harris and Mike Stephens. *A combined corner and edge detector.* Alvey vision conference, vol. 15, page 50, 1988. (Cited on pages 23 and 50.)

[Hart 1968] P. Hart. *The condensed nearest neighbor rule (Corresp.)*. Information Theory, IEEE Transactions on, vol. 14, no. 3, pages 515–516, May 1968. (Cited on page 40.)

[Hofmann 1999] Thomas Hofmann, Jan Puzicha and Michael I Jordan. *Learning from dyadic data*. Advances in neural information processing systems, pages 466–472, 1999. (Cited on page 39.)

[Hsu 2003] Chih-Wei Hsu, Chih-Chung Chang and Chih-Jen Lin. *A practical guide to support vector classification*. Rapport technique, Department of Computer Science, National Taiwan University, 2003. (Cited on pages 59 and 62.)

[Iosifidis 2012] Alexandros Iosifidis, Anastasios Tefas and Ioannis Pitas. *View-invariant action recognition based on artificial neural networks*. Neural Networks and Learning Systems, IEEE Transactions on, vol. 23, no. 3, pages 412–424, 2012. (Cited on page 42.)

[Izadinia 2012] Hamid Izadinia and Mubarak Shah. *Recognizing complex events using large margin joint low-level event model*. In Computer Vision–ECCV 2012, pages 430–444. Springer Berlin Heidelberg, 2012. (Cited on pages 8 and 42.)

[Jain 1999] Anil K. Jain, M. Narasimha Murty and Patrick J. Flynn. *Data Clustering: A Review*. ACM Comput. Surv., vol. 31, no. 3, pages 264–323, 1999. (Cited on page 39.)

[Jain 2013] Mihir Jain, Hervé Jégou and Patrick Bouthemy. *Better exploiting motion for better action recognition*. In CVPR - International Conference on Computer Vision and Pattern Recognition, Portland, États-Unis, April 2013. (Cited on pages 37 and 206.)

[Jegou 2010] Herve Jegou, Matthijs Douze, Cordelia Schmid and Patrick Perez. *Aggregating local descriptors into a compact image representation*. In Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, pages 3304–3311. IEEE, 2010. (Cited on pages 36 and 37.)

[Jegou 2012] Herve Jegou, Florent Perronnin, Matthijs Douze, Jorge Sanchez, Patrick Perez and Cordelia Schmid. *Aggregating local image descriptors into compact codes*. Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 34, no. 9, pages 1704–1716, 2012. (Cited on page 38.)

[Jensen 1996] Finn V. Jensen. Introduction to bayesian networks. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1st édition, 1996. (Cited on page 40.)

[Jiang 2011] Z. Jiang, Z. Lin and L. S. Davis. *Recognizing Human Actions by Learning and Matching Shape-Motion Prototype Trees*. TPAMI, 2011. (Cited on pages 181, 193, 202 and 203.)

[Jiang 2012] Yu-Gang Jiang, Qi Dai, Xiangyang Xue, Wei Liu and Chong-Wah Ngo. *Trajectory-Based Modeling of Human Actions with Motion Reference Points*. In Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato and Cordelia Schmid, editeurs, Computer Vision – ECCV 2012, volume 7576 of *Lecture Notes in Computer Science*, pages 425–438. Springer Berlin Heidelberg, 2012. (Cited on page 206.)

[Johansson 1973] Gunnar Johansson. *Visual perception of biological motion and a model for its analysis*. Perception & Psychophysics, vol. 14, no. 2, pages 201–211, 1973. (Cited on page 18.)

[Jolliffe 2002] I.T. Jolliffe. Principal component analysis. Springer Series in Statistics. Springer, 2002. (Cited on page 138.)

[Kaaniche 2009] M.B. Kaaniche and F. Bremond. *Tracking HoG Descriptors for Gesture Recognition*. In Advanced Video and Signal Based Surveillance, 2009. AVSS '09. Sixth IEEE International Conference on, pages 140–145, Sept 2009. (Cited on page 26.)

[Kaaniche 2010] Mohamed-Becha Kaaniche and Francois Bremond. *Gesture Recognition by Learning Local Motion Signatures*. In CVPR, 2010. (Cited on page 203.)

[Kad 2003] Scale Saliency: a novel approach to salient feature and scale selection, July 2003. (Cited on page 24.)

[Kalal 2010] Z. Kalal, J. Matas and K. Mikolajczyk. *P-N Learning: Bootstrapping Binary Classifiers by Structural Constraints*. In CVPR, 2010. (Cited on page 159.)

[Kantorov 2014] Vadim Kantorov and Ivan Laptev. *Efficient feature extraction, encoding and classification for action recognition*. In CVPR 2014 - Computer Vision and Pattern Recognition, Columbus, United States, June 2014. (Cited on page 206.)

[Karpathy 2014] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar and Li Fei-Fei. *Large-scale Video Classification with Convolutional Neural Networks*. In CVPR, 2014. (Cited on page 42.)

[Khoshelham 2012] Kourosh Khoshelham and Sander Oude Elberink. *Accuracy and Resolution of Kinect Depth Data for Indoor Mapping Applications*. Sensors, vol. 12, no. 2, pages 1437–1454, 2012. (Cited on page 21.)

[Kim 2007] Tae-Kyun Kim, Shu-Fai Wong and Roberto Cipolla. *Tensor Canonical Correlation Analysis for Action Classification*. In CVPR, 2007. (Cited on page 202.)

[Klaser 2008] Alexander Klaser, Marcin Marszalek and Cordelia Schmid. *A Spatio-Temporal Descriptor Based on 3D-Gradients*. In BMVC, 2008. (Cited on pages 28, 198, 202 and 206.)

[Kliper-Gross 2012] Orit Kliper-Gross, Yaron Gurovich, Tal Hassner and Lior Wolf. *Motion Interchange Patterns for Action Recognition in Unconstrained Videos*. In ECCV, 2012. (Cited on page 206.)

[Koperski 2014] Michal Koperski, Piotr Bilinski and François Bremond. *3D Trajectories for Action Recognition*. In ICIP - The 21st IEEE International Conference on Image Processing, Paris, France, October 2014. IEEE. (Cited on page 205.)

[Kotsiantis 2007] S B Kotsiantis. *Supervised Machine Learning: A Review of Classification Techniques*. Informatica, vol. 31, no. 3, pages 249–268, 2007. (Cited on pages 40 and 42.)

[Kovashka 2010] Adriana Kovashka and Kristen Grauman. *Learning a Hierarchy of Discriminative Space-Time Neighborhood Features for Human Action Recognition*. In CVPR, 2010. (Cited on pages 34, 201 and 203.)

[Krapac 2011] Josip Krapac, Jakob Verbeek and Frédéric Jurie. *Modeling spatial layout with fisher vectors for image categorization*. In Computer Vision (ICCV), 2011 IEEE International Conference on, pages 1487–1494. IEEE, 2011. (Cited on page 38.)

[Kuehne 2011] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio and T. Serre. *HMDB: A Large Video Database for Human Motion Recognition*. In ICCV, 2011. (Cited on pages 14, 74 and 206.)

[Lafferty 2001] John D. Lafferty, Andrew McCallum and Fernando C. N. Pereira. *Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data*. In ICML, pages 282–289, 2001. (Cited on page 40.)

[Laptev 2003] Ivan Laptev and Tony Lindeberg. *Space-time Interest Points*. In International Conference on Computer Vision, pages 432–439, 2003. (Cited on pages 23, 49 and 50.)

[Laptev 2005] Ivan Laptev. *On space-time interest points*. International Journal of Computer Vision, vol. 64, no. 2-3, pages 107–123, 2005. (Cited on pages 9, 12, 24, 42, 100, 114, 138, 178 and 191.)

[Laptev 2006] Ivan Laptev and Tony Lindeberg. *Local descriptors for spatio-temporal recognition*. In Spatial Coherence for Visual Motion Analysis, pages 91–103. Springer Berlin Heidelberg, 2006. (Cited on page 28.)

[Laptev 2008] Ivan Laptev, Marcin Marszalek, Cordelia Schmid and Benjamin Rozenfeld. *Learning realistic human actions from movies*. In Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, pages 1–8. IEEE, 2008. (Cited on pages 11, 28, 30, 42, 49, 51, 57, 61, 63, 174 and 203.)

[Lazebnik 2006] Svetlana Lazebnik, Cordelia Schmid and Jean Ponce. *Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories*. In

Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, volume 2, pages 2169–2178. IEEE, 2006. (Cited on pages 11, 30, 36 and 174.)

[Li 2008] Xi Li, Weiming Hu, Zhongfei Zhang, Xiaoqin Zhang, Mingliang Zhu and Jian Cheng. *Visual tracking via incremental Log-Euclidean Riemannian subspace learning*. In Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, pages 1–8, June 2008. (Cited on page 111.)

[Li 2012] Peihua Li and Qilong Wang. *Local Log-Euclidean Covariance Matrix (L2ECM) for Image Representation and Its Applications*. In Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato and Cordelia Schmid, editeurs, Computer Vision - ECCV 2012, volume 7574 of *Lecture Notes in Computer Science*, pages 469–482. Springer Berlin Heidelberg, 2012. (Cited on page 111.)

[Litomisky 2012] Krystof Litomisky. *Consumer RGB-D Cameras and their Applications*. Rapport technique, University of California, 2012. (Cited on pages 20 and 21.)

[Liu 2008] Jingen Liu and Mubarak Shah. *Learning human actions via information maximization*. In Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, pages 1–8. IEEE, 2008. (Cited on pages 30, 33 and 42.)

[Liu 2009] Jingen Liu, Jiebo Luo and Mubarak Shah. *Recognizing Realistic Actions from Videos "in the Wild"*. In CVPR, 2009. (Cited on page 202.)

[Lowe 2004] David G. Lowe. *Distinctive Image Features from Scale-Invariant Keypoints*. International Journal of Computer Vision, vol. 60, no. 2, pages 91–110, November 2004. (Cited on page 28.)

[Lucas 1981] Bruce D. Lucas and Takeo Kanade. *An Iterative Image Registration Technique with an Application to Stereo Vision (DARPA)*. In Proceedings of the 1981 DARPA Image Understanding Workshop, pages 121–130, April 1981. (Cited on page 26.)

[MacQueen 1967] J. B. MacQueen. *Some Methods for Classification and Analysis of MultiVariate Observations*. In L. M. Le Cam and J. Neyman, editeurs, Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability, volume 1, pages 281–297. University of California Press, 1967. (Cited on pages 39 and 56.)

[Marszalek 2009] Marcin Marszalek, Ivan Laptev and Cordelia Schmid. *Actions in context*. In Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, pages 2929–2936. IEEE, 2009. (Cited on pages 57, 61 and 211.)

[Matikainen 2009] Pyry Matikainen, Martial Hebert and Rahul Sukthankar. *Trajectons: Action Recognition Through the Motion Analysis of Tracked Features*. In Workshop on Video-Oriented Object and Event Classification, ICCV 2009, September 2009. (Cited on page 26.)

[Matikainen 2010] Pyry Matikainen, Martial Hebert and Rahul Sukthankar. *Representing pairwise spatial and temporal relations for action recognition*. In Computer Vision–ECCV 2010, pages 508–521. Springer Berlin Heidelberg, 2010. (Cited on pages 32, 200 and 204.)

[Messing 2009] R. Messing, C. Pal and H. Kautz. *Activity recognition using the velocity histories of tracked keypoints*. In ICCV, 2009. (Cited on pages 14, 26, 27, 69 and 204.)

[Moeslund 2006] Thomas B. Moeslund, Adrian Hilton and Volker KrĂźger. *A survey of advances in vision-based human motion capture and analysis.* Computer Vision and Image Understanding, vol. 104, no. 2-3, pages 90–126, 2006. (Cited on page 6.)

[Müller 2006] Meinard Müller and Tido Röder. *Motion Templates for Automatic Classification and Retrieval of Motion Capture Data*. In Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA '06, pages 137–146, Aire-la-Ville, Switzerland, Switzerland, 2006. Eurographics Association. (Cited on page 205.)

[Murthy 1998] Sreerama K. Murthy. *Automatic Construction of Decision Trees from Data: A Multi-Disciplinary Survey*. Data Min. Knowl. Discov., vol. 2, no. 4, pages 345–389, 1998. (Cited on page 40.)

[Niebles 2006] Juan Carlos Niebles, Hongcheng Wang and Li Fei-Fei. *Unsupervised learning of human action categories using spatial-temporal words*. In BMVC, 2006. (Cited on pages 36 and 202.)

[Nowozin 2007] S. Nowozin, G. Bakir and K. Tsuda. *Discriminative Subsequence Mining for Action Classification*. In Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on, pages 1–8, Oct 2007. (Cited on page 42.)

[Oh 2014] Sangmin Oh, Scott McCloskey, Ilseo Kim, Arash Vahdat, KevinJ. Cannons, Hossein Hajimirsadeghi, Greg Mori, A.G.Amitha Perera, Megha Pandey and JasonJ. Corso. *Multimedia event detection with multimodal feature fusion and temporal concept localization*. Machine Vision and Applications, vol. 25, no. 1, pages 49–69, 2014. (Cited on page 42.)

[Oikonomopoulos 2005] A Oikonomopoulos, I Patras and M. Pantic. *Spatiotemporal salient points for visual recognition of human actions*. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, vol. 36, no. 3, pages 710–719, June 2005. (Cited on page 24.)

[Ojala 2002] T. Ojala, M. Pietikainen and T. Maenpaa. *Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns*. TPAMI, 2002. (Cited on page 158.)

[One 1913] *One Look Is Worth A Thousand Words*. Piqua Leader-Dispatch, page 2, August 1913. (Cited on page 1.)

[Oneata 2013] Dan Oneata, Jakob Verbeek and Cordelia Schmid. *Action and Event Recognition with Fisher Vectors on a Compact Feature Set*. In ICCV 2013 - IEEE Intenational Conference on Computer Vision, Sydney, Australia, December 2013. IEEE. (Cited on pages 37, 38, 56 and 60.)

[Oreifej 2013] Omar Oreifej and Zicheng Liu. *HON4D: Histogram of Oriented 4D Normals for Activity Recognition from Depth Sequences*. 2013 IEEE Conference on Computer Vision and Pattern Recognition, vol. 0, pages 716–723, 2013. (Cited on page 205.)

[Oshin 2011] O. Oshin, A Gilbert and R. Bowden. *Capturing the relative distribution of features for action recognition*. In Automatic Face Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on, pages 111–116, March 2011. (Cited on pages 35 and 201.)

[Over 2014] Paul Over, George Awad, Martial Michel, Jonathan Fiscus, Greg Sanders, Wessel Kraaij, Alan F. Smeaton and Georges QuĂŠnot. *TRECVID 2014 – An Overview of the Goals, Tasks, Data, Evaluation Mechanisms and Metrics*. In Proceedings of TRECVID 2014. NIST, USA, 2014. (Cited on page 211.)

[Pang 2008] Yanwei Pang, Yuan Yuan and Xuelong Li. *Gabor-Based Region Covariance Matrices for Face Recognition*. IEEE Trans. Circuits Syst. Video Techn., vol. 18, no. 7, pages 989–993, 2008. (Cited on page 100.)

[Perronnin 2007] Florent Perronnin and Christopher R. Dance. *Fisher Kernels on Visual Vocabularies for Image Categorization*. In CVPR, 2007. (Cited on pages 56 and 58.)

[Perronnin 2010a] Florent Perronnin, Yan Liu, Jorge Sánchez and Herve Poirier. *Large-Scale Image Retrieval with Compressed Fisher Vectors*. In CVPR, 2010. (Cited on pages 58 and 59.)

[Perronnin 2010b] Florent Perronnin, Jorge Sánchez and Thomas Mensink. *Improving the Fisher Kernel for Large-Scale Image Classification*. In ECCV, 2010. (Cited on pages 36, 37, 39, 58, 59 and 60.)

[Philbin 2008] James Philbin, Michael Isard, Josef Sivic and Andrew Zisserman. *Lost in quantization: Improving particular object retrieval in large scale image databases*. In In CVPR, 2008. (Cited on page 36.)

[Piro 2010] Paolo Piro, Richard Nock, Frank Nielsen and Michel Barlaud. *Boosting k-NN for categorization of natural scenes*. CoRR, vol. abs/1001.1221, 2010. (Cited on page 40.)

[Poppe 2010] Ronald Poppe. *A Survey on Vision-based Human Action Recognition*. Image Vision Comput., vol. 28, no. 6, pages 976–990, June 2010. (Cited on page 18.)

[Porikli 2006] Fatih Porikli, Oncel Tuzel and Peter Meer. *Covariance tracking using model update based means on Riemannian manifolds*. In Proc. IEEE Conf. on Computer Vision and Pattern Recognition, 2006. (Cited on pages 100 and 110.)

[Rabiner 1990] Lawrence R. Rabiner. *Readings in Speech Recognition*. chapitre A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, pages 267–296. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990. (Cited on page 40.)

[Raptis 2010] Michalis Raptis and Stefano Soatto. *Tracklet Descriptors for Action Modeling and Video Analysis*. In ECCV, 2010. (Cited on page 204.)

[Raptis 2011] Michalis Raptis, Darko Kirovski and Hugues Hoppe. *Real-time Classification of Dance Gestures from Skeleton Animation*. In Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA '11, pages 147–156, New York, NY, USA, 2011. ACM. (Cited on page 21.)

[Reynolds 1995] D.A Reynolds and R.C. Rose. *Robust text-independent speaker identification using Gaussian mixture speaker models*. Speech and Audio Processing, IEEE Transactions on, vol. 3, no. 1, pages 72–83, Jan 1995. (Cited on page 39.)

[Rifkin 2004] Ryan Rifkin and Aldebaro Klautau. *In Defense of One-Vs-All Classification*. J. Mach. Learn. Res., vol. 5, pages 101–141, December 2004. (Cited on page 63.)

[Rokach 2010] Lior Rokach. *Ensemble-based classifiers*. Artificial Intelligence Review, vol. 33, no. 1-2, pages 1–39, 2010. (Cited on page 42.)

[Rosten 2006] Edward Rosten and Tom Drummond. *Machine learning for high-speed corner detection*. In European Conference on Computer Vision, volume 1, pages 430–443, May 2006. (Cited on page 26.)

[Rumelhart 1986] D. E. Rumelhart, G. E. Hinton and R. J. Williams. *Learning internal representations by error propagation*. pages 318–362, 1986. (Cited on page 41.)

[Ryoo 2009] Michael S Ryoo and Jake K Aggarwal. *Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities*. In Computer Vision, 2009 IEEE 12th International Conference on, pages 1593–1600. IEEE, 2009. (Cited on pages 31, 200 and 202.)

[Sadanand 2012] S. Sadanand and J. J. Corso. *Action Bank: A High-Level Representation of Activity in Video*. In CVPR, 2012. (Cited on page 206.)

[Salton 1968] Gerard. Salton. Automatic information organization and retrieval. McGraw Hill Text, 1968. (Cited on page 36.)

[Satkin 2010] S. Satkin and M. Hebert. *Modeling the temporal extent of actions*. In ECCV, 2010. (Cited on page 204.)

[Saul 1997] Lawrence K. Saul and Fernando Pereira. *Aggregate and mixed-order Markov models for statistical language processing*. CoRR, vol. cmp-lg/9706007, 1997. (Cited on page 39.)

[Schapire 1999] Robert E. Schapire. *A Brief Introduction to Boosting*. In Proceedings of the 16th International Joint Conference on Artificial Intelligence - Volume 2, IJCAI'99, pages 1401–1406, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc. (Cited on page 41.)

[Schuldt 2004] Christian Schuldt, Ivan Laptev and Barbara Caputo. *Recognizing Human Actions: A Local SVM Approach*. In 17th International Conference on Pattern Recognition (ICPR), ICPR '04, pages 32–36, Washington, DC, USA, 2004. IEEE Computer Society. (Cited on pages 36 and 68.)

[Scovanner 2007] Paul Scovanner, Saad Ali and Mubarak Shah. *A 3-dimensional Sift Descriptor and Its Application to Action Recognition*. In Proceedings of the 15th International Conference on Multimedia, MULTIMEDIA '07, pages 357–360, New York, NY, USA, 2007. ACM. (Cited on page 28.)

[Shi 1994] Jianbo Shi and Carlo Tomasi. *Good Features to Track*. In 1994 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94), pages 593 – 600, 1994. (Cited on pages 26 and 53.)

[Shi 2013] Feng Shi, Emil Petriu and Robert Laganiere. *Sampling Strategies for Real-Time Action Recognition*. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2013. (Cited on pages 198 and 206.)

[Shotton 2012] Jamie Shotton, Ross Girshick, Andrew Fitzgibbon, Toby Sharp, Mat Cook, Mark Finocchio, Richard Moore, Pushmeet Kohli, Antonio Criminisi, Alex Kipman and Andrew Blake. *Efficient Human Pose Estimation from Single Depth Images*. Trans. PAMI, 2012. (Cited on pages 20 and 21.)

[Simonyan 2013] K. Simonyan, O. M. Parkhi, A. Vedaldi and A. Zisserman. *Fisher Vector Faces in the Wild*. In British Machine Vision Conference, 2013. (Cited on page 60.)

[Sivic 2003] J. Sivic and A. Zisserman. *Video Google: A Text Retrieval Approach to Object Matching in Videos*. In Proceedings of the International Conference on Computer Vision, volume 2, pages 1470–1477, October 2003. (Cited on page 36.)

[Sivic 2005] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman and W. T. Freeman. *Discovering Object Categories in Image Collections*. In Proceedings of the International Conference on Computer Vision, 2005. (Cited on page 36.)

[Smeaton 2006] Alan F. Smeaton, Paul Over and Wessel Kraaij. *Evaluation Campaigns and TRECVid*. In Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval, MIR '06, pages 321–330, New York, NY, USA, 2006. ACM. (Cited on page 162.)

[Solmaz 2012] B. Solmaz, Shayan A. Modiri and M. Shah. *Classifying Web Videos using a Global Video Descriptor*. Machine Vision and Applications, 2012. (Cited on page 206.)

[Soomro 2012] Khurram Soomro, Amir Roshan Zamir and Mubarak Shah. *UCF101: A dataset of 101 human actions classes from videos in the wild*. arXiv preprint arXiv:1212.0402, 2012. (Cited on page 211.)

[Spe 1911] *Speakers Give Sound Advice*. Syracuse Post Standard, page 18, March 1911. (Cited on page 1.)

[Sreekanth 2010] V. Sreekanth, A. Vedaldi, C. V. Jawahar and A. Zisserman. *Generalized RBF feature maps for efficient detection*. In Proceedings of the British Machine Vision Conference (BMVC), 2010. (Cited on pages 61 and 62.)

[Stöttinger 2010] Julian Stöttinger, Bogdan Tudor Goras, Thomas Pöntiz, Allan Hanbury, Nicu Sebe and Theo Gevers. *Systematic Evaluation of Spatio-temporal Features on Comparative Video Challenges*. In International Workshop on Video Event Categorization, Tagging and Retrieval, in conjunction with ACCV, 2010. (Cited on page 48.)

[Strang 2009] G. Strang. Introduction to linear algebra. Wellesley-Cambridge Press, 2009. (Cited on page 111.)

[Sun 2009] Ju Sun, Xiao Wu, Shuicheng Yan, Loong-Fah Cheong, Tat-Seng Chua and Jintao Li. *Hierarchical Spatio-Temporal Context Modeling for Action Recognition*. In CVPR, 2009. (Cited on pages 26 and 33.)

[Sun 2010] Ju Sun, Yadong Mu, Shuicheng Yan and Loong-Fah Cheong. *Activity recognition using dense long-duration trajectories*. In Multimedia and Expo (ICME), 2010 IEEE International Conference on, pages 322–327, July 2010. (Cited on page 27.)

[Sun 2013] Chen Sun and Ram Nevatia. *Large-scale web video event classification by use of Fisher Vectors*. IEEE Winter Conference on Applications of Computer Vision, vol. 0, pages 15–22, 2013. (Cited on page 38.)

[Sun 2014] Lin Sun, Kui Jia, Tsung-Han Chan, Yuqiang Fang, Gang Wang and Shuicheng Yan. *DL-SFA: Deeply-Learned Slow Feature Analysis for Action Recognition*. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2014. (Cited on page 203.)

[Sundaram 2010] N. Sundaram, T. Brox and K. Keutzer. *Dense point trajectories by GPU-accelerated large displacement optical flow*. In European Conference on Computer Vision (ECCV), Lecture Notes in Computer Science. Springer, Sept. 2010. (Cited on page 54.)

[Sutton 1988] Richard S. Sutton. *Learning to Predict by the Methods of Temporal Differences*. pages 9–44, 1988. (Cited on page 39.)

[Sutton 1998] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. IEEE Transactions on Neural Networks, vol. 9, no. 5, pages 1054–1054, 1998. (Cited on page 39.)

[Székely 2009] Gábor J. Székely and Maria L. Rizzo. *Brownian distance covariance*. The Annals of Applied Statistics, vol. 3, no. 4, pages 1236–1265, 2009. (Cited on pages 130, 132, 133 and 136.)

[Ta 2010] Anh Phuong Ta, Christian Wolf, Guillaume Lavoue, Atilla Baskurt and Jean-Michel Jolion. *Pairwise features for human action recognition*. In ICPR, 2010. (Cited on pages 31, 32, 200 and 202.)

[Thurau 2008] C. Thurau and V. Hlavac. *Pose primitive based human action recognition in videos or still images*. In Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, pages 1–8, June 2008. (Cited on page 42.)

[Tomasi 1991] C. Tomasi and T. Kanade. Detection and tracking of point features. Shape and motion from image streams. School of Computer Science, Carnegie Mellon Univ., 1991. (Cited on page 26.)

[Tuzel 2006] Oncel Tuzel, Fatih Porikli and Peter Meer. *Region Covariance: A Fast Descriptor for Detection And Classification*. In In Proc. 9th European Conf. on Computer Vision, pages 589–600, 2006. (Cited on pages 100 and 110.)

[Tuzel 2008] O. Tuzel, F. Porikli and P. Meer. *Pedestrian Detection via Classification on Riemannian Manifolds*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 30, no. 10, pages 1713–1727, October 2008. (Cited on pages 109, 110 and 112.)

[Vapnik 1979] V Vapnik. Estimation of dependences based on empirical data [in russian]. Nauka, Moscow, 1979. (English translation: Springer Verlag, New York, 1982). (Cited on page 60.)

[Vapnik 1995] Vladimir N. Vapnik. The nature of statistical learning theory. Springer-Verlag New York, Inc., New York, NY, USA, 1995. (Cited on pages 41 and 60.)

[Vilalta 2002] Ricardo Vilalta and Youssef Drissi. *A Perspective View and Survey of Meta-Learning*. Artificial Intelligence Review, vol. 18, no. 2, pages 77–95, 2002. (Cited on page 42.)

[Viola 2001] Paul Viola and Michael Jones. *Rapid object detection using a boosted cascade of simple features*. In Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, volume 1, pages I–511. IEEE, 2001. (Cited on pages 112 and 158.)

[Vishwakarma 2013] Sarvesh Vishwakarma and Anupam Agrawal. *A survey on activity recognition and behavior understanding in video surveillance*. The Visual Computer, vol. 29, no. 10, pages 983–1009, 2013. (Cited on page 17.)

[Vishwanathan 2010] S. V. N. Vishwanathan, Zhaonan Sun, Nawanol Ampornpunt and Manik Varma. *Multiple kernel learning and the SMO algorithm*. In Advances in neural information processing systems, pages 2361–2369, 2010. (Cited on page 180.)

[Wang 2007] Xiaogang Wang, Gianfranco Doretto, Thomas Sebastian, Jens Rittscher and Peter Tu. *Shape and appearance context modeling*. In Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on, pages 1–8. IEEE, 2007. (Cited on page 100.)

[Wang 2009] Heng Wang, Muhammad Muneeb Ullah, Alexander Klaser, Ivan Laptev, Cordelia Schmid*et al.* *Evaluation of local spatio-temporal features for action recognition*. In BMVC 2009-British Machine Vision Conference, 2009. (Cited on pages 25, 29 and 48.)

[Wang 2010] Jinjun Wang, Jianchao Yang, Kai Yu, Fengjun Lv, Thomas S. Huang and Yihong Gong. *Locality-constrained Linear Coding for image classification*. In CVPR, pages 3360–3367, 2010. (Cited on pages 27, 36 and 37.)

[Wang 2011a] Heng Wang, Alexander Klaser, Cordelia Schmid and Cheng-Lin Liu. *Action recognition by dense trajectories*. In Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, pages 3169–3176. IEEE, 2011. (Cited on pages 9, 12, 29, 42, 50, 53, 54, 55, 57, 61, 100, 179, 192 and 203.)

[Wang 2011b] Jiang Wang, Zhuoyuan Chen and Ying Wu. *Action recognition with multiscale spatio-temporal contexts*. In Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, pages 3185–3192. IEEE, 2011. (Cited on pages 10, 33, 34, 201, 203 and 204.)

[Wang 2012] Jiang Wang, Zicheng Liu, Ying Wu and Junsong Yuan. *Mining actionlet ensemble for action recognition with depth cameras*. In Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, pages 1290–1297. IEEE, 2012. (Cited on pages 14, 21, 70 and 205.)

[Wang 2013a] Heng Wang, Alexander Kläser, Cordelia Schmid and Cheng-Lin Liu. *Dense trajectories and motion boundary descriptors for action recognition*. International Journal of Computer Vision, vol. 103, no. 1, pages 60–79, 2013. (Cited on pages 27, 29 and 206.)

[Wang 2013b] Heng Wang and Cordelia Schmid. *Action recognition with improved trajectories*. In Computer Vision (ICCV), 2013 IEEE International Conference on, pages 3551–3558. IEEE, 2013. (Cited on pages 60, 205, 206 and 207.)

[Watkins 1992] Christopher JCH Watkins and Peter Dayan. *Q-learning*. Machine learning, vol. 8, no. 3, pages 279–292, 1992. (Cited on page 39.)

[Weinland 2011] Daniel Weinland, Remi Ronfard and Edmond Boyer. *A survey of vision-based methods for action representation, segmentation and recognition*. Computer

Vision and Image Understanding, vol. 115, no. 2, pages 224–241, 2011. (Cited on page 17.)

[Willems 2008] Geert Willems, Tinne Tuytelaars and Luc Van Gool. *An efficient dense and scale-invariant spatio-temporal interest point detector*. In Computer Vision–ECCV 2008, pages 650–663. Springer Berlin Heidelberg, 2008. (Cited on pages 24 and 29.)

[Wong 2007] Kwan-Yee Kenneth Wong and Roberto Cipolla. *Extracting spatiotemporal interest points using global information*. In Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on, pages 1–8. IEEE, 2007. (Cited on page 25.)

[Wu 2008] Xindong Wu, Vipin Kumar, J Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J McLachlan, Angus Ng, Bing Liu, S Yu Philip*et al. Top 10 algorithms in data mining*. Knowledge and Information Systems, vol. 14, no. 1, pages 1–37, 2008. (Cited on pages 42 and 60.)

[Wu 2011a] Shandong Wu, Omar Oreifej and Mubarak Shah. *Action recognition in videos acquired by a moving camera using motion decomposition of lagrangian particle trajectories*. In Computer Vision (ICCV), 2011 IEEE International Conference on, pages 1419–1426. IEEE, 2011. (Cited on pages 181, 193 and 202.)

[Wu 2011b] X. Wu, D. Xu, L. Duan and J. Luo. *Action Recognition using Context and Appearance Distribution Features*. In CVPR, 2011. (Cited on pages 181, 193, 201, 202 and 203.)

[Wu 2014] Baoxin Wu, Chunfeng Yuan and Weiming Hu. *Human Action Recognition Based on Context-Dependent Graph Kernels*. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2014. (Cited on pages 181, 193 and 202.)

[Yamato 1992] J. Yamato, J. Ohya and K. Ishii. *Recognizing human action in time-sequential images using hidden Markov model*. In Computer Vision and Pattern Recognition, 1992. Proceedings CVPR '92., 1992 IEEE Computer Society Conference on, pages 379–385, 1992. (Cited on pages 21, 22 and 42.)

[Yang 2012] Yang Yang and Mubarak Shah. *Complex Events Detection Using Data-Driven Concepts*. In Computer Vision - ECCV 2012 - 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part III, pages 722–735, 2012. (Cited on pages 8 and 42.)

[Yilma 2005] A Yilma and Mubarak Shah. *Recognizing human actions in videos acquired by uncalibrated moving cameras*. In Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on, volume 1, pages 150–157. IEEE, 2005. (Cited on pages 18 and 19.)

[Yuan 2009a] Chunfeng Yuan, Weiming Hu, Xi Li, Stephen J. Maybank and Guan Luo. *Human Action Recognition under Log-Euclidean Riemannian Metric*. In Hongbin

Zha, Rin ichiro Taniguchi and Stephen J. Maybank, editeurs, ACCV (1), volume 5994 of *Lecture Notes in Computer Science*, pages 343–353. Springer, 2009. (Cited on pages 101 and 102.)

[Yuan 2009b] J. Yuan, Z. Liu and Y. Wu. *Discriminative Subvolume Search for Efficient Action Detection*. In CVPR, 2009. (Cited on page 203.)

[Yuan 2012] Guo-Xun Yuan, Chia-Hua Ho and Chih-Jen Lin. *Recent Advances of Large-Scale Linear Classification*. Proceedings of the IEEE, vol. 100, no. 9, pages 2584–2603, 2012. (Cited on page 63.)

[Zhang 2000] Guoqiang Peter Zhang. *Neural networks for classification: a survey*. IEEE Transactions on Systems, Man, and Cybernetics, Part C, vol. 30, no. 4, pages 451–462, 2000. (Cited on page 41.)

[Zhang 2007] Jianguo Zhang, Marcin Marszalek, Svetlana Lazebnik and Cordelia Schmid. *Local features and kernels for classification of texture and object categories: A comprehensive study*. International Journal of Computer Vision, vol. 73, no. 2, page 2007, 2007. (Cited on pages 62, 63 and 64.)

[Zhang 2012] Yimeng Zhang, Xiaoming Liu, Ming-Ching Chang, Weina Ge and Tsuhan Chen. *Spatio-Temporal phrases for activity recognition*. In Computer Vision–ECCV 2012, pages 707–721. Springer Berlin Heidelberg, 2012. (Cited on pages 202 and 203.)

[Zhang 2014] Hao Zhang, Wenjun Zhou, Christopher Reardon and Lynne E. Parker. *Simplex-Based 3D Spatio-Temporal Feature Description for Action Recognition*. In Computer Vision and Pattern Recognition, 2014. CVPR 2014. IEEE Conference on, June 2014. (Cited on page 203.)

[Zhou 2010] Xi Zhou, Kai Yu, Tong Zhang and Thomas S Huang. *Image classification using super-vector coding of local image descriptors*. In Computer Vision–ECCV 2010, pages 141–154. Springer Berlin Heidelberg, 2010. (Cited on pages 36 and 37.)

[Zhou 2013] Qiang Zhou, Gang Wang, Kui Jia and Qi Zhao. *Learning to Share Latent Tasks for Action Recognition*. In Computer Vision (ICCV), 2013 IEEE International Conference on, pages 2264–2271, Dec 2013. (Cited on page 37.)